INSTITUT FÜR LINGUISTIK
Abteilung für Computerlinguistik

Universitätsstraße 1 D–40225 Düsseldorf

Heinrich Heine Universität Düsseldorf

Investigating Psycholinguistically-inspired Training Losses for Human Reading Time Prediction with Transformers

Lukas Michael Mielczarek

Master Thesis
in Artificial Intelligence and Data Science

Reviewers: Prof. Dr. Laura Kallmeyer
Prof. Dr. Benoit Crabbé

Abstract

This thesis investigates whether attention patterns in transformer language models provide psycholinguistically informative predictors of human reading times. While surprisal has been highly successful as an expectation-based predictor, memory-based theories suggest that processing difficulty also depends on retrieval and integration costs. To examine this connection, recent research has proposed attention entropy as an operationalisation of cue-based retrieval. This work introduced additional psycholinguistically-inspired attention-derived measures including weighted attention distance, activation, and difference. These are extracted from transformer language models and evaluated on eye-tracking data using linear mixed-effects models.

The first experiment shows that local unnormalised attention entropy and attention distance provide small but systematic improvements in predicting go-past times beyond lexical surprisal and standard baseline predictors. Similar patterns are observed for activation and difference. However, the measures are strongly correlated with each other and with sentence position, making it difficult to identify a single underlying cognitive mechanism.

The second experiment tests whether attention entropy and attention distance become more predictive when they are included directly in the transformer training objective. Models are trained with combined language-modelling and attention-cost losses. The results provide no evidence that this explicit optimisation improves reading-time prediction: pure language models yield equally strong or stronger attention-based predictors.

Overall, attention-derived structural costs complement surprisal as correlational predictors of reading difficulty, but their cognitive interpretation remains indirect and their mutual relations difficult to establish. The findings suggest that psycholinguistically relevant attention patterns emerge as by-products of predictive contextualisation and are not under reductional pressure.

Contents

List of Figures **v**

List of Tables **vi**

1 Introduction **1**

1.1 Motivation . . . 3

1.1.1 The role of structural predictors in the presence of surprisal . . . 3

1.1.2 Attention as a memory framework beyond interference . . . 5

1.1.3 Attention costs as minimisation constraints . . . 5

1.2 Research questions . . . 6

1.3 Contributions . . . 7

1.4 Related work . . . 7

1.4.1 Surprisal and neural language models . . . 7

1.4.2 Attention-based predictors in transformer models . . . 8

1.4.3 Other memory-based predictors . . . 8

1.4.4 Modifying language model training objectives . . . 9

1.5 Overview . . . 9

2 Prerequisites **10**

2.1 Notation and basic definitions . . . 10

2.1.1 Sets, tuples and functions . . . 10

2.1.2 Tensors, vectors and matrices . . . 11

2.1.3 Indexing and slicing . . . 13

2.1.4 Tensor composition . . . 13

2.1.5 Linguistic objects . . . 14

2.2 Probability theory . . . 15

2.3 Information theory . . . 17

2.3.1 Surprisal . . . 17

2.3.2 Entropy . . . 19

2.3.3 Mutual information . . . 19

2.3.4 Cross-entropy and Kullback-Leibler divergence . . . 21

3 Reading times **22**

3.1 Methodological overview 22
3.1.1 Variability between readers 23
3.1.2 Variability between tasks 24
3.2 Eye-tracking 24
3.2.1 Historical background 25
3.2.2 Eye movements during reading 25
3.2.3 Eye-tracking technology 26
3.2.4 Measures derived from eye movements 26
3.2.5 Methodological considerations 27
3.2.6 Established findings 28
3.2.7 Eye-tracking corpora 28
3.3 Key-press methods 33
3.3.1 Stationary-window presentation 33
3.3.2 Moving-window presentation 33
3.3.3 Measures derived from key-press data 34
3.3.4 Methodological considerations 34
3.3.5 Established findings 35
3.3.6 Key-press corpora 35
3.3.7 Comparing the paradigms 38

4 Human sentence processing **41**
4.1 Introductory overview 41
4.1.1 Linguistic structure 41
4.1.2 Representations and activation 42
4.1.3 Incrementality, connectedness and predictiveness 42
4.1.4 Linking hypotheses and resource limitations 43
4.1.5 Architectural issues 45
4.2 Expectation-based sentence processing theories 47
4.2.1 Surprisal theory 47
4.2.2 Entropy Reduction Hypothesis 60
4.3 Memory-based sentence processing theories 61
4.3.1 Dependency Locality Theory 62
4.3.2 Cue-based retrieval 67

5 Language models **76**

5.1 n-gram models . . . 77

5.1.1 Parameterisation . . . 78

5.1.2 Estimation . . . 78

5.1.3 Limitations . . . 79

5.2 Recurrent neural networks . . . 79

5.2.1 Abstract notion . . . 80

5.2.2 Estimation . . . 80

5.2.3 Limitations and extensions . . . 80

5.3 Transformer . . . 81

5.3.1 Abstract notion . . . 81

5.3.2 Parameterisation . . . 82

5.3.3 Estimation . . . 99

5.3.4 Limitations and extensions . . . 103

5.4 Transformers in sentence processing . . . 104

5.4.1 Next-token probability . . . 105

5.4.2 Attention structure . . . 109

6 Modelling reading times with linear mixed-effects models **124**

6.1 Hypothesis testing, p-values and error rates . . . 124

6.1.1 Statistical test, p-value . . . 125

6.1.2 Type-I error, Type-II error and power . . . 126

6.2 Linear regression . . . 127

6.2.1 Assumptions . . . 128

6.2.2 Estimation . . . 129

6.2.3 Why ordinary linear regression is insufficient . . . 130

6.3 From linear regression to linear mixed-effects models . . . 130

6.3.1 Fixed effects . . . 130

6.3.2 Random effects . . . 131

6.3.3 Definition . . . 131

6.3.4 Assumptions . . . 133

6.3.5 Shrinkage of extreme estimates and working with imbalanced data . . . 135

6.3.6 Estimation . . . 135

6.3.7 Likelihood-ratio tests . . . 136

6.3.8 Implementations . . . 137
6.4 Psycholinguistic application . . . 139
6.4.1 Crossed random effects in psycholinguistics . . . 139
6.4.2 Choosing the random-effects structure . . . 139
6.4.3 Spillover . . . 140
6.4.4 Multiple testing . . . 141

7 Experiment 1 **144**
7.1 Methodology . . . 144
7.1.1 Language model . . . 145
7.1.2 Proposed predictors . . . 152
7.1.3 Psycholinguistic evaluation . . . 159
7.2 Results . . . 166
7.2.1 Hyperparameter search . . . 166
7.2.2 Language model training . . . 168
7.2.3 Model identification . . . 173
7.2.4 Statistical testing . . . 178
7.2.5 Robustness to sentence position . . . 183
7.3 Discussion . . . 184

8 Experiment 2 **187**
8.1 Methodology . . . 187
8.1.1 Language model . . . 188
8.1.2 Psycholinguistic evaluation . . . 193
8.2 Results . . . 194
8.2.1 Weight search . . . 194
8.2.2 Model training . . . 196
8.2.3 Model identification . . . 197
8.2.4 Statistical testing . . . 198
8.3 Discussion . . . 201

9 Conclusion **204**

References **207**

List of Figures

1 Eye-movement schematic 27

2 Key-press self-paced reading time schematic 34

3 Incremental syntactic analysis of the garden-path sentence *While the man hunted the deer ran into the woods.* 46

4 Updated search space for hyperparameter optimisation run 2. 168

5 Learning rates sampled during hyperparameter search plotted against the minimal validation loss reported for the search trials 169

6 Residual dropout sampled during hyperparameter search plotted against the minimal validation loss reported for the respective search trials. 169

7 Feed-forward dropout sampled during hyperparameter search plotted against the minimal validation loss reported for the respective search trials. 169

8 Embedding dropout sampled during hyperparameter search plotted against the minimal validation loss reported for the respective search trials. 169

9 Minimal reported validation loss plotted separately for the trials that include a bias term and for those that omit it. 170

10 Development of loss during training. 172

11 Development of LMEM log likelihood resulting from fitting surprisal generated by model checkpoints to go-past times in Provo. 172

12 Residual plot for the baseline linear mixed-effects model in experiment 1 175

13 Q-Q plot for the baseline linear mixed-effects model in experiment 1 175

14 Q-Q plot for the model specified in Equation 171. 176

15 Histogram of local unnormalised attention entropy before removing the observations for the first word in each sentence. 177

16 Population-level predicted go-past times as a function of the attention measures, based on fixed effects from the mixed-effects models, with other predictors held constant at average values 180

17 Attention measures extracted for an example sentence 181

18 Weights for the language modelling objective sampled during hyperparameter search for $\mathrm{CM}^{\mathrm{entr}}$ plotted against the maximal log-likelihoods reported for the respective search trial 194

19 Weights for the language modelling objective sampled during hyperparameter search for $\mathrm{CM}^{\mathrm{entr}}$ plotted against the language modelling loss reported for the respective search trial 195

20 Weights for the language modelling objective sampled during hyperparameter search for $\mathrm{CM}^{\mathrm{entr}}$ plotted against the attention entropy per token reported for the respective search trial 195

21 Weights for the language modelling objective sampled during hyperparameter search for $\mathrm{CM}^{\mathrm{dist}}$ plotted against the maximal log-likelihoods reported for the respective search trial 196

22 Weights for the language modelling objective sampled during hyperparameter search for $\mathrm{CM}^{\mathrm{dist}}$ plotted against the language modelling loss reported for the respective search trial 197

23 Weights for the language modelling objective sampled during hyperparameter search for $\mathrm{CM}^{\mathrm{dist}}$ plotted against the attention entropy per token reported for the respective search trial 197

24 Bar-plot of log-likelihoods of LMEMs including surprisal generated by the LM, $\mathrm{CM}^{\mathrm{entr}}$ and $\mathrm{CM}^{\mathrm{dist}}$ models. 198

List of Tables

1 Eye-tracking datasets of "normal" reading. 29

2 Datasets of "normal" self-paced reading in English. 36

3 Example dataset containing repeated observations. 132

4 Fixed architectural hyperparameters for language model training in experiment 1. 146

5 Overview over the dataset used for LM training and validation. 147

6 Hyperparameters to identify for the transformer language model in experiment 1, corresponding search spaces and the initial distribution from which they are sampled. 149

7 Settings for the hyperparameter search procedure, sampler and pruner 152

8 Computational resources used for the hyperparameter search 167

9 Best configuration found during hyperparameter search in experiment 1. 168

10 Benchmarks on the final LM in experiment 1 171

11 Settings for training the final language model 171

12 Baseline LMEM effect estimates and standard errors. 177

13 Baseline LMEM random effect variances and correlations. 177

14 Identified LMEMs for testing the attention-based measures 178

15 Effects of attention-based predictors on go-past times in linear mixed-effects models 182

16 p-values and BH-corrected p-values for the effect of local unnormalised predictors controlled for word position 184

17 Pearson correlations between baseline predictors and attention metrics computed for the observations included in the LMEM analysis. 185

18 Language modelling perplexities on the WikiText training and validation splits for the three LMs, attention entropy CMs and attention distance CMs. 198

19 Simplified LMEMs for testing the attention costs in experiment 2. 199

20 Results of likelihood-ratio tests for the effect of attention entropy and attention distance extracted from LMs and CMs on Provo go-past times. 201

1 Introduction

Understanding how humans process language in real time is a central goal of psycholinguistics. It is commonly assumed that comprehenders incrementally construct and update an interpretation of a sentence during reading, integrating incoming words with prior context, although the precise nature of this incrementality remains debated (Altmann and Mirković, 2009; Ferreira et al., 2002).

A large body of research has aimed to characterise the mechanisms underlying this process and to explain variation in *human reading times* (RTs) as an observable proxy of processing difficulty.

Two major theoretical frameworks guide this line of work: expectation-based (Hale, 2001; Levy, 2008) and memory-based theories (Gibson, 2000). Expectation-based accounts posit that the predictability of a word given the context that precedes it is an indicator of its processing difficulty. Intuitively, unexpected words require larger updates to the current interpretation, leading to increased processing effort (Levy, 2008). This idea is most successfully operationalised using *surprisal*, defined as the negative logarithm of the conditional probability of a word (Hale, 2001; Levy, 2008).

In contrast, memory-based theories explain difficulties in processing with limitations of human working memory. In particular, these approaches attribute reading slow-downs to the retrieval of previously encountered linguistic elements and the integration of new words into an incremental interpretation of the sentence. The theories vary widely in their assumptions about the nature of cognitive representations and the mode of operation of the human working memory, leading to different and sometimes contradictory explanations for reading time patterns.

The *cue-based retrieval* theory of sentence processing (Van Dyke and Lewis, 2003; Van Dyke and McElree, 2006) assumes that a set of retrieval cues is provided by the current word. Items in memory that match the cues more closely are more likely to be retrieved. *Similarity-based interference* arises when multiple items partially match the cues, leading to competition during retrieval and increased processing difficulty (Gordon et al., 2002; Gordon et al., 2006).

In a similar spirit, *Dependency Locality Theory* (DLT, Gibson, 2000) assumes that humans explicitly and incrementally recover dependencies between linguistic items in memory and each new word, integrating it into a structure representing an interpretation of the sentence prefix observed so far. DLT's central assumption is that longer syntactic dependencies result in higher *integration costs*, i.e. in a higher demand on cognitive resources required for integration (Gibson, 2000).

Memory-based theories are difficult to operationalise quantitatively, as they typically require explicit assumptions about the representation of linguistic structure and memory. This often involves committing to a particular syntactic formalism or feature representation, making it difficult to disentangle theoretical claims about processing from assumptions about representational structure.

Recent research has aimed to improve our understanding of human sentence processing using neural network-based *language models* (LMs). These models are trainined on large amounts of text to predict the next word given its context. Despite being optimised for this relatively simple objective, they generate fluent text and achieve strong performance across a wide range of language processing tasks (Brown et al., 2020; Radford et al., 2019). Naturally, this motivates investigations into whether such models capture aspects of human language processing.

Surprisal theory has received substantial empirical support from neural LMs: strong correlations between surprisal and human reading times have been confirmed using the predictions of state-

of-the-art transformer-based LMs across a wide range of datasets, experimental paradigms and languages (Wilcox et al., 2023; Wilcox et al., 2020; Shain et al., 2022; Rathi, 2021). Additionally, neural language models provide a complementary approach to the representational issue of memory-based theories of sentence processing. Rather than specifying features a priori, they learn distributed representations directly from data as a by-product of optimising their weights with respect to the training objective (Ryu and Lewis, 2025). Thus, transformer model internals offer task-specific embeddings of the words in its input as well as additional structure associated with their interactions.

Against this backdrop, recent work has tried to unify expectation-based and memory-based approaches by linking intermediate units in transformers to variation in reading times. However, it is not straightforward to identify direct counterparts of memory-based costs in transformer language models. Due to their distributed representations and parallel processing architecture, transformers do not explicitly encode a symbolic syntactic analysis. Instead, information about prior context is encoded implicitly in continuous representations and accessed step-by-step via attention mechanisms.

A recent line of work targets transformer self-attention. The mechanism distributes weight over previous tokens when computing contextual representations, which has been interpreted as a potential analogue of memory retrieval by Ryu and Lewis (2021). Ryu and Lewis (2025) show that an attention entropy measure on transformer models explains variance in reading times complementary to surprisal. These studies are mostly focused on modelling only similarity-based interference effects, leaving open the question of whether alternative measures inspired by other contributors to processing difficulty proposed by psycholinguistic theory such as dependency locality, capture additional aspects of processing difficulty. Furthermore, it is usually evaluated only on one specific pre-trained transformer language model, `GPT-2 small` (Radford et al., 2019), with the generelisation to architectures of different size and to other tokenisation strategies being unclear.

The present work addresses this gap by proposing and systematically evaluating three additional attention-based predictors alongside attention entropy, including measures of attention distance, activation, and attention difference. These predictors are motivated by different theoretical considerations, with attention distance in particular drawing inspiration from Dependency Locality Theory. All predictors are evaluated in multiple variants, allowing for a more fine-grained analysis of how attention patterns relate to human reading behaviour. Additionally, the analysis is conducted on a larger transformer model than commonly used in prior work, serving to assess the robustness of earlier findings for attention entropy.

Moreover, if higher values of these attention-based measures indicate increased processing difficulty and thus longer reading times, they invite an interpretation as implicit cognitive constraints. This is inspired by the fact that a language model's training objective is equivalent to the minimisation of surprisal. Under the assumption that the human processing system is optimised to avoid elongated processing times, this raises the question of whether attention-based costs reflect merely by-products of the processing mechanism or if subjecting transformer LMs to minimisation of attention costs during training would increase their alignment with human RT patterns.

To investigate this, the present work explores whether explicitly incorporating attention-based cost terms into the training objective of transformer models leads to more predictive patterns of these costs over with respect to RTs. This second line of investigation allows for a more direct test of whether attention-derived measures can be interpreted as meaningful components of a cognitive

model of sentence processing.

1.1 Motivation

This section outlines the conceptual considerations motivating the present thesis. Particularly, it outlines why lexical surprisal alone might not provide a complete explanatory account of reading times and motivates the investigation of attention-based costs beyond entropy-based measures extracted from transformer language models. Furthermore, it provides the rationale for exploring the integration of such costs into the optimisation objective of transformer language models.

1.1.1 The role of structural predictors in the presence of surprisal

An increasing number of works relate transformer language models to human cognition. Their parameters are determined by the optimisation for next word prediction given the distributional properties of the training material and inductive biases of the model architecture. This naturally connects to *surprisal theory*, according to which processing difficulty is determined by the unexpectedness of linguistic input (Hale, 2001; Levy, 2008). Numerous works have found a strong linear relationship between surprisal and reading times (Wilcox et al., 2023; Brothers and Kuperberg, 2021; Xu et al., 2023), establishing it as a central baseline in psycholinguistic modelling.

A common interpretation suggests that surprisal is a *causal bottleneck*: under this view, all factors influencing processing difficulty manifest through their effect on the comprehender's internal probability distribution, reflecting an efficient allocation of cognitive resources over interpretations and possible continuations by the comprehender (Levy, 2008). This raises the fundamental question of the prospects of finding meaningful contributors to reading time beyond surprisal. In this context, it is important to note that the bottleneck claim does not imply that lexical surprisal estimates derived from current language models constitute a sufficient predictor of processing difficulty. Rather, it states that human processing difficulty is mediated by an explicit or implicit internal probability distribution which may differ from the one acquired by an artificial language model based on its training corpus.

Indeed, empirical evidence suggests systematic discrepancies between surprisal-based predictions and observed reading times. In particular, surprisal has been found to underestimate processing difficulty in cases involving syntactic ambiguity and reanalysis, such as garden-path constructions (Schijndel and Linzen, 2021; Arehalli et al., 2023). Similar mismatches arise in constructions involving the closing sites of long-distance dependencies and centre embeddings, where reading times appear to increase despite relatively predictable lexical items (Levy, 2008).

There are two orthogonal views attempting to explain these limitations (Wilcox, 2025; Arehalli et al., 2023):

Model inadequacy The first view argues that the structure of the probability distributions emerging in neural language models is inadequate and differs from the human model of language. Under this view, surprisal remains the correct theoretical quantity, but current models fail to approximate the relevant distribution accurately.

In principle one would expect cognitive constraints to reflect on the distributional properties of the

linguistic material in large corpora captured by language models, assuming that the statistics of language are in turn shaped by human processing ease. The argument reads as follows: constructions leading to larger cognitive costs are rarer than structures of lower costs conveying the same message because they consume more energy or take more time to process, impeding communication which is evolutionary disadvantageous. Evidence for this assumption is provided by Futrell et al. (2015) who discover dependency length minimisation in 37 languages. MacDonald (2013) argue that humans acquire the distribution of language they consume, to shape their comprehension processes

This connection could be violated if the cognitive language production and the comprehension apparatus are not shared, contrary to what is assumed by Pickering and Garrod (2013), so that the assumed processing burden would not inflict on the producer. Alternatively, if the minimisation of difficulties such as dependency lengths was driven by the cooperative principle (Grice, 1975), with the sender optimising their message for the receiver to be able to decode it most efficiently, as MacDonald (2013) suggest, distributional discrepancies could be explained by violations of this principle. For instance, the production of texts is often not a spontaneous process but a result of multiple phases of revisions guided by additional goals such as the compression of information into a restricted format or stylistic aspirations.

Furthermore, neural language models may lack inductive biases that favour the construction of explicit syntactic representations during processing, leading to probability distributions that differ from those employed by humans (Schijndel and Linzen, 2021). Arehalli et al. (2023) hypothesise that syntactic factors influence lexical surprisal more strongly in humans than in language models since we also need to understand and interpret what we read, for which syntax is essential (under the assumption of compositional semantics), while the goal of LMs is only to predict the succeeding word form. In addition, language models might maintain probability mass over multiple competing parses, whereas human comprehenders might commit more strongly to a single interpretation due to memory limitations or processing constraints (Timkey and Linzen, 2025). This could result in lower surprisal values in the model compared to human expectations.Finally, recent evidence suggests that large language models have a tendency to memorise their training material, leading to unrepresentative surprisal estimates (Oh and Schuler, 2023b).

Theory incompleteness The other view argues that surprisal alone does not capture all relevant aspects of processing difficulty and that additional mechanisms – particularly those related to memory and structural integration – contribute independently to reading times.

Empirical evidence suggests that while surprisal does predict higher reading times for constructions involving long-distance dependencies or similarity-based interference, it can be elevated at the wrong positions. Following Levy (2013), a long gap between a head and its dependent leads to a higher reading time at the integration point, even when the word is semantically obvious. See (1) from King and Just (1991) for an example. The slowdown is at *admitted*, for which reading times are underpredicted if only considering surprisal due to the left context making the appearance of a verb probable. In fact, surprisal would predict a slow-down as soon as the syntactic form is evident, here at the beginning of the phrase, due to constructions of this kind being rare in the training material (Levy, 2013).

(1) The reporter that the senator attacked admitted the error ...

A similar phenomenon for object relative clauses is shown by Levy (2008) to be insufficiently explained by surprisal. Thus, the measure may fail to capture local processing difficulty whereas memory-based and structural predictors, such as DLT-style integration cost in Example (1), can provide additional explanatory power.

1.1.2 Attention as a memory framework beyond interference

Motivated by the preceding argument, recent studies investigate LMs as models of human cognition beyond their output distributions. Transformer internals are tested for psycholinguistically interpretable patterns that emerge as a side-effect of next token prediction. For instance, Ryu and Lewis (2021) identify attention entropy as a proxy for similarity-based interference. Ryu and Lewis (2025) show that the measure predicts reading times above and beyond surprisal with statistical significance. So far, most work in this realm has focused on cue-based retrieval accounts of processing difficulty.

However, Example 1 would not be explainable by interference effects since *attacked* would already give rise to higher reading times, having two potential matching subjects in its context, instead of only *admitted*. This raises the question whether other theories such as DLT can be explored in the framework of transformer networks.

The present thesis attempts to do precisely that. This requires interpreting transformer attention not only as encoding interference between competing memory representations queried by the current word (Ryu and Lewis, 2021) but as implementing a continuous form of retrieval as assumed by Oh and Schuler (2022). Then, transformer attention provides a natural candidate for the extraction of alternative measures inspired by psycholinguistic theory. In particular – without getting ahead of a precise formalisation to follow in this thesis – establishing a dependecy between the current word and an item in memory in the framework of DLT could be expressed as a retrieval operation and the dependency distance derived from the surface distance between the participants of the retrieval, yielding a testable measure of processing difficulty. This motivates the proposal of multiple theory-driven attention-based costs in this work.

1.1.3 Attention costs as minimisation constraints

Most existing work treats attention-based measures as emergent properties of transformer models trained solely for next-token prediction. However, if memory-related costs reflect fundamental constraints on human sentence processing, it is natural to ask whether such constraints should influence the training procedure and by this the characteristics of processing and the distribution of other contributors to reading times such as surprisal.

Human language acquisition is often assumed to be shaped by pressures to optimise processing efficiency and minimise cognitive effort (Gibson et al., 2019; Fedzechkina et al., 2012). If processing difficulty is observable through reading times and reading times are influenced by factors such as memory retrieval or structural integration, one may expect these factors to be subject to optimisation as well. From this perspective, training language models solely to minimise cross-entropy loss[1] may be insufficient to capture all relevant aspects of human-like processing.

[1] equivalent to minimising next-word surprisal

One can see that this matters by taking cue-based retrieval as an example: It does not necessarily follow from unsupervised training for next-token prediction that the attention patterns of transformer models resemble human retrieval characteristics. Where retrieving several items might be beneficial for maintaining multiple alternative interpretations of a sentence in a next-token prediction model, the human brain might prefer fewer retrievals if such an operation induces processing costs, maybe even tolerating slightly higher surprisal depending on the magnitude of the effects.

This motivates the exploration of incorporating candidates for complementary predictors as part of the loss during language model training in this work. With respect to surprisal, the approach pursued here occupies an intermediate space between the two perspectives of model inadequacy and theory incompleteness. Introducing a bias into the transformer model is hypothesised to shape both attention-based costs and next-token distributions. Additionally, it extends the theory by presupposing that memory-based costs are necessary in addition to surprisal to adequately account for human reading time variation

1.2 Research questions

In the preceding section, I argue for investigating the effect of memory-based predictors of human reading times using the attention-patterns of transformer language models. Besides the established entropy measure, the interpretation of attention as a form of information retrieval invites the construction of additional predictors inspired by psycholinguistic theories such as Dependency Locality Theory. Furthermore, it raises the question of whether such psycholinguistically motivated costs should be treated as explicit constraints during neural network training.

Against this backdrop, the following research questions are addressed:

[Q1] How well do structural costs extracted from transformer models with a pure language modelling objective, in particular attention entropy and weighted attention distance, predict human reading times?

[Q2] Does integrating the minimisation of attention-based costs, in particular entropy and weighted distance, into a transformer language model's loss function improve the predictive quality of the model for reading times?

Based on these questions, the following hypotheses are formulated:

[H1a] Attention entropy derived from a transformer model trained with a standard language modelling objective accounts for variation in human reading times not captured by lexical surprisal.

[H1b] A DLT-inspired weighted attention distance accounts for variation in human reading times not captured by lexical surprisal.

[H2] A transformer language model trained with an augmented loss function that includes attention-based structural costs yields predictors that, in combination with surprisal, provide a better fit to human reading times than predictors derived from a model trained solely with a standard language modelling objective.

1.3 Contributions

To address the research questions formulated above, this work makes the following contributions.

First, inspired by attention entropy, it proposes a set of attention-based structural predictors of human reading times derived from transformer language models. These predictors are based on established theories of human sentence processing and operationalised as attention-based proxies for memory-related processing costs within transformer architectures. Their predictive value is then systematically evaluated with respect to their ability to explain variance in reading times beyond lexical surprisal.

Methodologically, this work introduces a framework for training *cost-augmented language models*, in which attention-based structural costs are incorporated into the optimisation objective of transformer models. This enables a direct test of whether such costs function as emergent properties of standard language modelling or as interpretable constraints during training. It evaluates these models using psycholinguistic data and linear mixed-effects modelling, establishing a principled comparison of predictive performance while accounting for subject- and item-level variability.

Conceptually, this work contributes to the ongoing discussion and partly contradictory line of research regarding the role and characteristics of memory-based costs in human sentence processing by relating attention-based predictors to established psycholinguistic theories, making them quantitatively testable in a performant framework, omitting explicit representational choices. Additionally, by considering transformer models of larger scale than those used in prior work on attention entropy, this study contributes to an assessment of robustness and generalisability of the relationship between attention entropy and human reading times.

Finally, as a secondary aim, this work provides a detailed theoretical and methodological overview of the relevant literature. This includes (i) a structured comparison of commonly used reading-time corpora, and (ii) a synthesis of recent transformer-based approaches to modelling human sentence processing, situating them within their psycholinguistic theoretical foundations.

Together, these contributions aim to clarify the relationship between predictive and memory-based accounts of sentence processing and to provide practical guidance for future research at the intersection of computational modelling and psycholinguistics.

1.4 Related work

1.4.1 Surprisal and neural language models

A large body of work has demonstrated that surprisal derived from probabilistic models is a strong predictor of human reading times. Early approaches relied on probabilistic parsers, where surprisal arises naturally as a by-product (Hale, 2001; Levy, 2008; Demberg and Keller, 2008). More recent work has shown that surprisal estimates obtained from neural language models, in particular recurrent and transformer-based architectures, achieve state-of-the-art performance in predicting reading times across a variety of datasets and languages (Smith and Levy, 2013; Goodkind and Bicknell, 2018; Wilcox et al., 2023).

Despite this success, several studies have identified systematic discrepancies between surprisal-based predictions and observed reading times. In particular, surprisal derived from neural language

models has been found to underestimate processing difficulty in cases involving syntactic ambiguity, reanalysis, and long-distance dependencies (Schijndel and Linzen, 2021; Arehalli et al., 2023). These findings have motivated the exploration of extensions to surprisal-based accounts.

One such extension is *lossy-context surprisal*, which incorporates memory limitations by conditioning predictions on a noisy or compressed representation of prior context (Futrell et al., 2020). Under this framework, memory-related effects are not treated as independent predictors but instead alter the effective probability distribution, thereby maintaining surprisal as the sole mediator of processing difficulty. Empirically, this idea has been supported by work showing that restricting the accessible context of transformer models (Kuribayashi et al., 2022) and biasing transformer attention based on recency (Clark et al., 2025) can improve reading time predictions.

I follow a similar idea in this work. Both lossy context-surprisal and distance-based cost calculated using attention scores induce a preference of local information. However, while lossy-context surprisal allows forgetting previous words and assumes that a comprehender fills memory-holes with likely content, in the transformer models employed in the work at hand, previous input is not obstructed by noise. Access to it is possible at any given point in time – but discouraged by the training algorithm.

1.4.2 Attention-based predictors in transformer models

Beyond surprisal, recent work has investigated whether internal representations of transformer language models contain additional information relevant to human sentence processing. In particular, attention patterns have been interpreted as reflecting mechanisms of retrieval from memory.

Ryu and Lewis (2021) propose attention entropy as a measure of similarity-based interference, showing that the distribution of attention weights over previous tokens can capture competition between retrieval candidates. Subsequent work demonstrates that entropy-based measures of attention diffuseness and changes in attention patterns predict reading times beyond surprisal (Oh and Schuler, 2022; Ryu and Lewis, 2022; Ryu and Lewis, 2025). These findings suggest that attention-based measures may encode memory-related processing costs that are not fully captured by lexical probabilities.

A recurring question in this line of work concerns the cognitive interpretability of attention. While transformer models employ many attention heads in parallel, some studies argue that a smaller number of retrieval operations are cognitively more plausible and may suffice to capture key psycholinguistic effects. For instance, Timkey and Linzen (2023) show that a model with a single attention head can reproduce interference patterns. This hint at a methodological weakness of prior approaches (excluding Timkey and Linzen (2023)), as their major focus lies on a single pretrained transformer language model (`GPT-2 small`), with the generalisability of their results to the overall family of transformers being unknown. Thus, this work fills a methodological gap by evaluating attention entropy extracted from a differently sized transformer model trained from scratch.

1.4.3 Other memory-based predictors

Recent work has begun to explore transformer-based predictors beyond attention entropy. In particular, Oh and Schuler (2022) propose distance-based measures derived from attention patterns

that quantify the change in attention entropy and the change in attention patterns between two time steps. While establishing a promising path to close the explanatory gap between surprisal and reading time measurements, these predictors lack an explictly theoretical foundation based on human sentence processing accounts. The present work aims to take a more foundational approach by proposing theory-inspired attention costs.

1.4.4 Modifying language model training objectives

Various studies have gone beyond observing language model internals and aim to increase their psycholinguistic plausibility by altering their training objective. First, there is a growing collection of work using psycholinguistic data as a supervision signal for fine-tuning language models (Hollenstein et al., 2021; Kiegeland et al., 2024; Toneva and Wehbe, 2019; Aw et al., 2024; Schrimpf et al., 2021). These differ from the work at hand in that its aim is to explain reading times instead of generating them. Mielczarek et al. (2025) pursued this very goal by integrating a parsing objective into transformer attention and relating its surprisal and attention-based structural measures to reading times. This is also different than the current thesis since I do not introduce additional data for model supervision here.

1.5 Overview

The remainder of this thesis is structured as follows. Section 2 introduces important notation and a selection of necessary definitions. Section 3 provides an overview of reading time measurements and relevant datasets. Theoretical accounts of sentence processing are described in Section 4, including surprisal theory, cue-based retrieval and Dependency Locality Theory. This is followed by a detailed description of transformer language models in Section 5. Section 6 introduces linear mixed-effects models, used in the subsequent experimental sections: Section 7 proposes and tests multiple attention-based predictors of reading times and Section 8 investigates their inclusion in transformer language mode training. The results of both experiments are discussed and contextualised at the end of their respective sections. Finally, Section 9 summarises the findings and outlines directions for future work.

2 Prerequisites

2.1 Notation and basic definitions

This section introduces the mathematical notation used throughout the thesis. I will omit lengthy and rigorous discussions of axiomatic expressions since these are not the focus of this experimental work. Rather, I will use functional and practical definitions that suffice in this context.

The set of natural numbers $\mathbb{N}$ will not include 0. Variables such as $x \in \mathbb{N}$ will be italicised. Definitional assignments will be written using the $\coloneqq$ operator. For instance,

$$\mathrm{a} \coloneqq b$$

defines the symbol a to be equal to the value of variable b. This will be mainly used for shorthand function definitions.

Definitions of tuples or matrices frequently require the use of sets of indices. Therefore, a dedicated notation is introduced:

Definition 2.1 (Index set)**.** For $n \in \mathbb{N}$, define

$$[n] \coloneqq \{\, 1, \ldots, n \,\}.$$

All indexing in this thesis starts with 1.

2.1.1 Sets, tuples and functions

Uppercase letters such as X denote sets while tuples (i.e. finite sequences) are referred to using bold lowercase letters, e.g. $\boldsymbol{\alpha}$. The length of a tuple is given by $|\boldsymbol{\alpha}|$. The Cartesian product of sets X and Y is written $X \times Y$ and denotes the set of all 2-element tuples where the first item is an element of X and the second item is in Y. The iterated cartesian product over sequences of sets is defined as follows:

Definition 2.2 (Iterated cartesian product)**.** Let $n \in \mathbb{N}$ and let $X_1, \ldots, X_n$ be sets. The iterated cartesian product operator is defined by

$$\bigtimes_{i=j}^{n} X_i \coloneqq \begin{cases} X_n & \text{if } j = n, \\ X_j \times \left(\bigtimes_{i=j+1}^{n} X_i \right) & \text{else.} \end{cases}$$

When tuples are explicitly defined, angle brackets are used. Furthermore, their elements may be indexed using round brackets in subscript position:

$$\boldsymbol{\alpha} = \langle\, \boldsymbol{\alpha}_{(1)}, \ldots, \boldsymbol{\alpha}_{(|\boldsymbol{\alpha}|)} \,\rangle.$$

If the function f assigns element y in its codomain Y to x in its domain X, I will write that f *maps x to y*. This may be defined explicitly using the notation in the lower left. Additionally, a

shorthand definition in the style of Goldberg (2017) is used in this work where the domain is noted explicitly below and the codomain is left implicit (see below on the right side).

$$\begin{array}{ll} \mathrm{f}\colon X \to Y & \mathrm{f}(x) \coloneqq y \\ x \mapsto y & x \in X \end{array}$$

The set of all functions from X to Y is denoted as Y^X.

Function composition is specified using the operator $\circ$. For two functions $f\colon X \to Y$ and $g\colon Y \to Z$, the composition $g \circ f\colon X \to Z$ is defined as $(g \circ f)(x) \coloneqq g(f(x))$.

Definition 2.3 (Iterated composition)**.** Let $n \in \mathbb{N}$ and let f_i, f_{i+1} for all $i \in [n-1]$ be functions such that the composition $f_{i+1} \circ f_i$. The iterated concatenation operator is defined by

$$\bigcirc_{i=j}^{n} f_i \coloneqq \begin{cases} f_n & \text{if } j = n, \\ \left(\bigcirc_{i=j+1}^{n} f_i\right) \circ f_j & \text{else.} \end{cases}$$

Parameterised function will be written as $f[\theta](\,x\,)$, where x is an argument to the function and θ is a parameter.

2.1.2 Tensors, vectors and matrices

Definition 2.4 (Tensor)**.** Let $[i_1], \ldots, [i_k]$ be index sets. A k-th order tensor over $\mathbb{R}$ is defined as a function

$$\mathbf{T} : [i_1] \times \cdots \times [i_k] \to \mathbb{R}.$$

Equivalently, one can write $\mathbb{R}^{[i_1]\times\cdots\times[i_k]}$. I will use a common shorthand notation where only the maximum element of each index set is specified: $\mathbb{R}^{i_1\times\cdots\times i_k}$ (Goldberg, 2017). The n-th argument of $\mathbf{T}$ will also be referred to as the n-th axis of $\mathbf{T}$.

Vectors and matrices are special cases of tensors:

$$\mathbf{v} : [n] \to \mathbb{R},$$

$$\mathbf{M} : [n] \times [m] \to \mathbb{R}.$$

Matrices are written as rectangular arrays with the first index corresponding to the rows and the second index corresponding to the columns. Vectors are written as matrices with one column. A tensor of order 0 is a value in $\mathbb{R}$. Sometimes tensors mapping onto to 0 or only to 1 will be written as $\mathbf{0}$ or $\mathbf{1}$ if their domains are clear from context.

Definition 2.5 (Tensor addition and scalar multiplication)**.** Let $\mathbf{A}, \mathbf{B}$ be tensors of identical shape $[i_1] \times \cdots \times [i_k] \to \mathbb{R}$. For $x \in \mathbb{R}$, I assume that addition and scalar multiplication are defined by

$$(\mathbf{A} + \mathbf{B})(i_1, \ldots, i_k) \coloneqq \mathbf{A}(i_1, \ldots, i_k) + \mathbf{B}(i_1, \ldots, i_k),$$

$$(x \cdot \mathbf{A})(i_1, \ldots, i_k) \coloneqq x \cdot \mathbf{A}(i_1, \ldots, i_k).$$

These operations are understood to satisfy the usual algebraic laws (commutativity and associativity of addition, distributivity of scalar multiplication, etc.). With the operations defined above, tensors of fixed shape form finite-dimensional real vector spaces, motivating the use of the term vector in this context.

The multiplicative equivalent of tensor addition is defined as follows.

Definition 2.6 (Element-wise multiplication)**.** Let $\mathbf{A}, \mathbf{B} : [i_1] \times \cdots \times [i_k] \to \mathbb{R}$ be tensors of identical shape. The *element-wise multiplication* of $\mathbf{A}$ and $\mathbf{B}$ is the tensor $\mathbf{A} \odot \mathbf{B} : [i_1] \times \cdots \times [i_k] \to \mathbb{R}$ defined by

$$(\mathbf{A} \odot \mathbf{B})(i_1, \ldots, i_k) \coloneqq \mathbf{A}(i_1, \ldots, i_k) \cdot \mathbf{B}(i_1, \ldots, i_k).$$

Matrix multiplication is distinct from element-wise multiplication and defined in the usual way,

Definition 2.7 (Matrix multiplication)**.**

$$(\mathbf{AB})_{(i,k)} \coloneqq \sum_{j=1}^{m} \mathbf{A}_{(i,j)} \mathbf{B}_{(j,k)},$$

with m being the number of columns of $\mathbf{A}$ and the number of rows of $\mathbf{B}$. This operation is extended to matrix-vector and vector-vector multiplications, assuming an implicit change from vector to 1-column matrix.

Definition 2.8 (Transpose)**.** Let $\mathbf{M} \colon [n] \times [m] \to \mathbb{R}$ be a matrix. The *transpose* of $\mathbf{M}$ is the matrix $\mathbf{M}^\mathsf{T} : [m] \times [n] \to \mathbb{R}$ defined by

$$\mathbf{M}^\top(i, j) \coloneqq \mathbf{M}(j, i) \qquad \text{for all } (i, j) \in [m] \times [n].$$

Additionally, concatenation will be frequently used.

Definition 2.9 (Concatenation)**.** For vectors $\mathbf{x} : [d_1] \to \mathbb{R}$ and $\mathbf{y} : [d_2] \to \mathbb{R}$, define

$$\mathbf{x} \circ \mathbf{y} : [d_1 + d_2] \to \mathbb{R}$$

by

$$(\mathbf{x} \circ \mathbf{y})(i) = \begin{cases} \mathbf{x}(i) & 1 \le i \le d_1, \\ \mathbf{y}(i - d_1) & d_1 < i \le d_1 + d_2. \end{cases}$$

This extends to tensors of all shapes where $\circ_i$ denotes the axis of concatenation. $\circ$ is used as a shorthand for $\circ_1$.

For matrices, vertical and horizontal concatenations are defined as concatenations along the first and second axis respectively. They can be written using explicit array notation, with the matrices pasted in variable form or providing their elements. Let $\mathbf{A} : [n] \times [d_1] \to \mathbb{R}$ and $\mathbf{B} : [n] \times [d_2] \to \mathbb{R}$. The following is an example of a horizontal concatenation:

$$\mathbf{A} \circ_2 \mathbf{B} = \left(\begin{array}{c|c} \mathbf{A} & \mathbf{B} \end{array} \right) = \left(\begin{array}{ccc|ccc} \mathbf{A}_{(1,1)} & \cdots & \mathbf{A}_{(1,d_1)} & \mathbf{B}_{(1,1)} & \cdots & \mathbf{B}_{(1,d_2)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{(n,1)} & \cdots & \mathbf{A}_{(n,d_1)} & \mathbf{B}_{(n,1)} & \cdots & \mathbf{B}_{(n,d_2)} \end{array} \right)$$

The symbol $\bigcirc$ is used to perform concatenation on a sequence of tuples or tensors. By extension, for a matrix $\mathbf{M}$ with m columns and n rows:

$$\bigcirc(\mathbf{M}) = \mathbf{M}_{(1)} \circ \ldots \circ \mathbf{M}_{(m)} \colon [mn] \to \mathbb{R}.$$

2.1.3 Indexing and slicing

It is necessary to specify some additional aspects about indexing. In particular, while concatenation composes larger tensors from smaller ones, slicing indices are used to decompose a tensor.

Definition 2.10 (Slicing)**.** For integers $a, b \in [n]$ with $a \leq b$, define a slice as

$$(a{:}b) \coloneqq \{\, a, a+1, \ldots, b \,\}.$$

Slices can be used in indexing. For instance,

$$\mathbf{M}_{(a:b,c:d)} \colon a{:}b \times c{:}d \to \mathbb{R}, \qquad \langle\, j, k \,\rangle \mapsto \mathbf{M}_{(j,k)}$$

denotes a restriction of $\mathbf{M}$ to a subdomain. This is extended to tensors of any order. Note that consecutive slices are embedded into a single set of parentheses as a convention.

A bare colon $(:)$ is used as a shorthand for the slice that is equal to the respective complete index set. Trailing : symbols in sequences of consecutive slices may be omitted. Additionally if slice $(a{:}b)$ is used for indexing a tensor at an axis with m dimensions, I will equivalently write $(:b)$ if $a = 1$ and $(a{:})$ if $b = m$.

Definition 2.11 (Partial Indexing)**.** Slices can be used together with indices. This corresponds to partial application of the tensor function. For instance, for a matrix $\mathbf{M}$ with m columns and n rows, $\mathbf{M}_{(:,a)}$ is defined as

$$\mathbf{M}_{(:,a)} \colon [n] \to \mathbb{R}, \qquad j \mapsto \mathbf{M}_{(j,a)}.$$

As can be seen, this yields a function of domain $[m]$, i.e. a vector representing the a-th column.

2.1.4 Tensor composition

Finally, pointwise applications of functions on vector elements are written using the composition operator $\circ$. This is motivated by the functional interpretation of tensors. Take for instance vector $\mathbf{v} \colon [n] \to \mathbb{R}$ and function $f \colon \mathbb{R} \to \mathbb{R}$. Then, element-wise application yields a vector $[n] \to \mathbb{R}$ characterised by

$$(f \circ \mathbf{v})(i) = f(\mathbf{v}(i))$$

Definition 2.12 (Partial tensor composition)**.** In the general case, for any $a, b \in \mathbb{N}_0$ with $b < a$, a tensor $\mathbf{T}$ of order a can be composed with a function that has as its domain the set of tensors of order b by specifying the axes of $\mathbf{T}$ that the composition should be applied on. For instance, given a third-order tensor $\mathbf{T} \colon [n] \times [m] \times [p] \to \mathbb{R}$ and some function mapping matrices to vectors $g \colon ([n] \times [p] \to \mathbb{R}) \to ([q] \to \mathbb{R})$, composition on axes 1 and 3 is defined as a function

$$\begin{aligned} g \circ_{\langle 1,3 \rangle} \mathbf{T} \colon [m] \times [q] &\to \mathbb{R} \\ \langle\, i, j \,\rangle &\mapsto g(\mathbf{T}_{(:,i,:)})_{(j)} \end{aligned}$$

The composed function is also written as $g(_{\langle 1,3 \rangle}\mathbf{T})$. Note that the axis introduced by the codomain of g is integrated as an axis of the resulting tensor.

It can be seen that the indices in the axes that are not specified next to the composition symbol identify the items that g is applied on. When only a single axis is specified, the brackets are omitted. For instance, a function $\mathrm{avg}(_2\,\mathbf{M})$ could yield a vector of the means of the rows in $\mathbf{M}$, as the avg function is applied row-wise. As a special case, the pointwise vector operation $f \circ \mathbf{v}$ given above can also be expressed as $f(_{\langle\,\rangle}\mathbf{v})$, where the short-hand $f(\mathbf{v}_{(i)}) = f(_{\langle\,\rangle}\mathbf{v})_{(i)}$ is adopted, since the domain of f is the set of 0-order tensors, i.e. $\mathbb{R}$, so that no axes are specified for composition.

2.1.5 Linguistic objects

Definition 2.13 (Vocabulary)**.** A *vocabulary* is a non-empty finite set of symbols.

A vocabulary whose symbols correspond to characters will be referred to as a *character vocabulary*.

Definition 2.14 (Symbol sequence)**.** A symbol sequence of length $n \in \mathbb{N}_0$ over a vocabulary V is an element of

$$\underbrace{V \times \cdots \times V}_{n \text{ times}} = V^n.$$

The set of all symbol sequences over V is denoted as V^*. I will deviate from regular tuple notation for symbol sequences by writing

$$w = w_1 \ldots w_n$$

for $w \in V^n$. Furthermore, prefixes of w will be written in the following fashion:

$$w_{\leq i} \coloneqq w_1 \ldots w_{i-1} w_i$$
$$w_{<i} \coloneqq w_1 \ldots w_{i-1}$$

A symbol sequence over a character vocabulary is commonly called a string.

Note that in formal language theory, vocabularies are usually referred to as *alphabets* and symbol sequences are called *words* (Kallmeyer, 2010, Chapter 1.5). In this thesis, the term *word* will be reserved for natural language words to prevent unnecessary ambiguity.

Definition 2.15 (Annotation)**.** Given a character sequence s over a character vocabulary C, an annotation scheme over s with label set L is a mapping that assigns labels from L to subsequences of s.

$$a \colon \{\, \langle\, i_1, \ldots, i_n \,\rangle \mid 1 \leq i_1 < \cdots < i_n \leq |s|, i_1, \ldots, i_n \in \mathbb{N} \,\} \to L,$$

Definition 2.16 (Corpus)**.** A *corpus* over a vocabulary V is a tuple $\langle\, w, \mathcal{L} \,\rangle$ where w is a character sequence over V and $\mathcal{L}$ is a set of annotation schemes over s.

Examples of annotations include word boundaries, sentence boundaries, part-of-speech tags, or syntactic dependencies.

In order to build computational language models, running text is often segmented into subunits using a process called *tokenisation*.

Definition 2.17 (Tokenisation)**.** Given a symbol sequence w over the vocabulary V, tokenisation is a mapping

$$\tau \colon V^* \to (V^*)^*$$

that segments a sequence into a sequence of substrings

$$\tau(s) = \langle\, t_1, \ldots, t_n \,\rangle,$$

where each $t_i \in V^*$ is called a *token*. Tokenisation is most commonly applied to character sequences and extends naturally to corpora.

The *vocabulary induced by* τ refers to the set of distinct tokens occurring in τw.

Tokenisation over a corpus may rely on explicit boundary annotations if they are available, or infer boundaries using heuristics such as whitespace or punctuation. In modern language models, tokenisation often produces subword units rather than full words (Jurafsky and Martin, 2026, Chapter 2.4).

Finally, following Jurafsky and Martin (2026, Chapter 2.1), two important concepts need to be distinguished when analysing corpora: *types* and *instances*.

Definition 2.18 (Types and instances)**.** Let w be a symbol sequence and $\tau(w)$ its tokenisation.

A *token instance* is an individual occurrence of a token in the sequence $\tau(w)$ whereas a *token type* is an element of the vocabulary induced by τ.

When dealing with word-based tokenisation, I will also use the terms *word instances* and *word types*. Traditionally, these concepts are referred to simply as *tokens* and *types*. However, the prevalence of tokenisation procedures in modern language model systems has led to a change in semantics for the term *token* Jurafsky and Martin (2026, Chapter 2.1).

2.2 Probability theory

Some preliminaries with respect to probability theory will be necessary. This section mainly aims to specify notation for some basic concept that the reader is assumed to have knowledge of. For a comprehensive introduction into the field, see for instance Grimmett and Stirzaker (1992).

Let X be a discrete random variable taking values in a finite set $\mathcal{X}$ with probability mass function $P_X(x) \coloneqq P(\, X = x \,)$ for $x \in \mathcal{X}$. Thus, $P_X \colon \mathcal{X} \to [0, 1]$ defines a *probability distribution* over $\mathcal{X}$ with $\sum_{x \in \mathcal{X}} P_X(x) = 1$. I will write $P(x)$ when the underlying random variable is clear from context.

Two random variables X and Y are independent if and only if

$$P(\, X = x, Y = y \,) = P(\, X = x \,) P(\, Y = y \,)$$

for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.

Conditional probability is defined as

$$P(\, X = x \,|\, Y = y \,) \coloneqq \frac{P(\, X = x, Y = y \,)}{P(\, Y = y \,)},$$

whenever $P(Y = y) > 0$, where $P(\, X = x, Y = y \,)$ denotes the joint probability of $X = x$ and $Y = y$.

Another important concept is the *chain rule of probability* (Jurafsky and Martin, 2026, Chapter 3.1). For discrete random variables $X_1, \ldots, X_n$ and values $x_1, \ldots, x_n$ with $P(X_1 = x_1, \ldots, X_{i-1} = x_{i-1}) > 0$ for all $i \in [n]$, the joint distribution can be written as

$$P(\, X_1 = x_1, \ldots, X_n = x_n \,) = \prod_{i=1}^{n} P(\, X_i = x_i \,|\, X_1 = x_1, \ldots, X_{i-1} = x_{i-1} \,).$$

The expectation of a random variable taking numerical values is defined as

$$\mathbb{E}[X] \coloneqq \sum_{x \in \mathcal{X}} P(x) x.$$

Conditional expectation is denoted as $\mathbb{E}(\, X \,|\, Y = y \,) = \sum_{x \in \mathcal{X}} P(\, X = x \,|\, Y = y \,)$. In cases where several random variables are involved, a subscript may indicate the variable over which the expectation should be taken, for instance $\mathbb{E}_X[f(X, Y)] = \sum_{x \in \mathcal{X}} P(x) f(x, Y)$ yields a new random variable. Additionally, if the random variable is distributed according to some distribution $\mathcal{D}$, I will sometimes use the shorthand notation $\mathbb{E}_{X \sim \mathcal{D}}[g(X)]$.

The standard definitions for variance and covariance are assumed (denoted by $\mathrm{Var}[X]$ and $\mathrm{Cov}[X, Y]$). Furthermore, knowledge of the normal ($\mathcal{N}$), unit ($\mathcal{U}$), Bernoulli ($\mathcal{B}$), Binomial (Binom) and χ^2 distributions will be assumed (Grimmett and Stirzaker, 1992, Chapter 4.4). In order to specify that a random variable X is distributed according to a distribution $\mathcal{D}$, I will write $X \sim \mathcal{D}$.

A categorical distribution (i.e. a discrete distribution describing the result of a random variable that can have one of n possible categories) can be expressed using a vector with length n by associating each outcome with a distinct index in $[n]$.

Definition 2.19 (Probability simplex)**.** A vector $\mathbf{v}$ with dimensionality $n \in \mathbb{N}$ such that $\mathbf{v}_{(i)} \geq 0$ for all $i \in [n]$ and $\sum_{i=1}^{n} \mathbf{v}_{(i)} = 1$ is called a *probability simplex* (Boyd and Vandenberghe, 2004, Chapter 7.2).

The set of all probability simplexes of dimensionality n is denoted as Δ^n.

Definition 2.20 (Uniform Dirichlet distribution)**.** For some $n \in \mathbb{N}$, the *uniform Dirichlet distribution* $\mathrm{Dir}(n)$ is defined as the uniform distribution over Δ^n.[2]

A function $\mu : \mathbb{R}^n \to \Delta^n$ for some dimensionality $n \in \mathbb{N}$ will be called a *normalisation function*. Given normalisation function μ and an element $0_\mu \in \mathbb{R}$ such that for any input vector $\mathbf{v} \in \mathbb{R}^n$ with $\mathbf{x}_{(j)} = 0_\mu$ at any position $j \in [d]$, it holds that

$$\text{(1)} \qquad \mu(\mathbf{x})_{(j)} = 0$$

$$\text{and (2)} \qquad \mu(\mathbf{x}) = \mu(\mathbf{x}_{(:j-1)} \circ \mathbf{x}_{(j+1:)}),$$

μ_0 is called a *null-element of* μ.

The most relevant normalisation function is the softmax function.

[2] This is a special case of the general Dirichlet distribution given by Bishop (2006, Chapter 2.2.1) for which the parameters $\boldsymbol{\alpha}$ are specified to be $\langle\, 1, \ldots, 1 \,\rangle$.

Definition 2.21 (Softmax)**.** For dimensionality $n \in \mathbb{R}$ and a temparature $t \in \mathbb{R}$ with $t > 0$, the $\mathrm{softmax}$ function is defined such that for any vector $\mathbf{v} \in \mathbb{R}^n$,

$$\mathrm{softmax}_t\left(\mathbf{v}\right)_{(j)} \coloneqq \frac{\mathrm{e}^{\mathbf{v}_{(j)}/t}}{\sum_{i=1}^{n} \mathrm{e}^{\mathbf{v}_{(i)}/t}}.$$

Usually, t is not explicitly specified and assumed to be 1. Temperature scaling changes the shape of the distribution encoded in the output simplex. For $t > 1$, the distribution becomes flatter, whereas for $t < 1$, it becomes more peaked (Liu et al., 2024).

Corollary 2.1. The null-element of the $\mathrm{softmax}$ function is $-\infty$.

Proof. Since it holds that $\lim_{x \to -\infty} \mathrm{e}^x = 0$, the numerator (and the whole term) approaches zero for negative infinity, fulfilling condition (1). Likewise, the term can be ignored in the addition in the denominator of other indices since it approaches zero, showing (2). □

2.3 Information theory

This thesis makes use of several concepts from information theory, a mathematical framework for quantifying uncertainty and information in probabilistic systems. Information theory was established by Shannon (1948), who showed how limits on data compression and communication can be derived from probabilistic properties of signals. For comprehensive introductions see Cover and Thomas (2006) and MacKay (2002).

2.3.1 Surprisal

Surprisal, also known as *self-information*, *Shannon information* or *information content*, quantifies the information gained when an event is observed given a belief about its probability. Intuitively, unlikely events convey more information than likely ones.

The term *surprisal* was not introduced by Shannon (1948) directly but coined by Samson, 1951; Samson, 1953. The first use of the term *self-information* can be attributed to Fano (1961) while the origin of the terms *Shannon information* and *information content* (e.g. MacKay, 2002) is unclear.

Definition 2.22 (Surprisal)**.** Let X be a discrete random variable and let x be an outcome with probability $P(X = x) > 0$. The surprisal of observing x is defined as

$$\mathrm{I}_X(x) \coloneqq -\log_b P(X = x),$$

for a base $b > 1$. If the underlying random variable is clear from context, I will simply write $\mathrm{I}(x)$.

When given an additional random variable Y and outcome y, this definition extends to joint surprisal $\mathrm{I}_{X,Y}(x, y) = -\log_b P(X = x, Y = y)$.

Surprisal is inversely related to probability: highly probable events have low surprisal, whereas unlikely events have high surprisal. In the extreme case where $P(x) = 1$, the event conveys no new information and therefore $\mathrm{I}(x) = 0$.

Information theory establishes a close connection between probability and optimal data compression. In a coding scheme, each possible event is assigned a codeword over some finite vocabulary of size b. A *prefix code* is a coding scheme in which no codeword is the prefix of another codeword, ensuring that encoded messages can be decoded unambiguously (Cover and Thomas, 2006, p. 106). Now, in an optimal (most efficient) prefix code the length of the codeword for an event x is $-\log_b P(x)$ symbols, up to rounding to an integer (Cover and Thomas, 2006, Chapter 5.4).

The base b determines the unit in which information is measured. If $b = 2$, surprisal is measured in *bits*, i.e. in the number of symbols given a vocabulary of size 2; if $b = e$, the unit is called *nats* (MacKay, 2002, Chapter 18.3). As a convention, the $\log$ function in this work is assumed to have base 2 unless stated otherwise.

Surprisal satisfies an important additivity property. For two independent events x and y, with probabilities $P(x)$ and $P(y)$ their joint surprisal is equal to the sum of their individual surprisals.

Corollary 2.2. Let x and y be independent events. Then

$$\mathrm{I}(x, y) = \mathrm{I}(x) + \mathrm{I}(y).$$

Proof. If x and y are independent, then $P(x, y) = P(x)P(y)$. Hence

$$\mathrm{I}(x, y) \overset{\text{(Def. 2.22)}}{=} -\log_b P(x, y) = -\log_b P(x) - \log_b P(y) \overset{\text{(Def. 2.22)}}{=} \mathrm{I}(x) + \mathrm{I}(y). \tag{1}$$

□

In many situations, the information conveyed by an event depends on information that has already been observed. This leads to the notion of *conditional surprisal*.

Definition 2.23 (Conditional surprisal)**.** Let X and Y be discrete random variables and let x and y be outcomes with $P(X = x \mid Y = y) > 0$. The conditional surprisal of x given y is defined as

$$\mathrm{I}(\, X = x \,|\, Y = y \,) \coloneqq -\log_b P(\, X = x \,|\, Y = y \,).$$

Conditional surprisal quantifies the information gained when the event x is observed under the assumption that y has already been observed.

Corollary 2.3 (Chain rule for surprisal)**.** Let $X_1, \ldots, X_n$ be discrete random variables and let $x_1, \ldots, x_n$ be outcomes such that $P(\, X_1 = x_1, \ldots, X_n = x_n \,) > 0$. Then the surprisal of the joint event decomposes additively as

$$\mathrm{I}(\, X_1 = x_1, \ldots, X_n = x_n \,) = \sum_{i=1}^{n} \mathrm{I}(\, X_i = x_i \,|\, X_1 = x_1, \ldots, X_{i-1} = x_{i-1} \,).$$

For $i = 1$, the conditioning event is empty, so that $\mathrm{I}(\, X_1 = x_1 \,|\, X_1 = x_1, \ldots, X_0 = x_0 \,) \coloneqq \mathrm{I}(\, X_1 = x_1 \,)$.

Proof. By the chain rule of probability (Section 2.2),

$$\begin{aligned}
\mathrm{I}(\, X_1 = x_1, \ldots, X_n = x_n \,) &= -\log_b P(\, X_1 = x_1, \ldots, X_n = x_n \,) \\
&= -\log_b \prod_{i=1}^{n} P(\, X_i = x_i \,|\, X_1 = x_1, \ldots, X_{i-1} = x_{i-1} \,) \\
&\overset{\text{(Cor. 2.2)}}{=} \sum_{i=1}^{n} -\log_b P(\, X_i = x_i \,|\, X_1 = x_1, \ldots, X_{i-1} = x_{i-1} \,) \\
&= \sum_{i=1}^{n} \mathrm{I}(\, X_i = x_i \,|\, X_1 = x_1, \ldots, X_{i-1} = x_{i-1} \,).
\end{aligned} \tag{2}$$

□

2.3.2 Entropy

While surprisal measures the information of a single outcome, *entropy* measures the expected information content of a random variable.

Definition 2.24 (Entropy)**.** Let X be a discrete random variable taking values in $\mathcal{X}$ and probability mass function P. The entropy of X is defined as

$$\mathrm{H}(X) \coloneqq \mathbb{E}[\mathrm{I}(X)] = -\sum_{x \in \mathcal{X}} P(x) \log_b P(x).$$

Note that $0 \cdot \log_b(0)$ is assumed to equal to 0 in this context since $\lim_{x \to 0} x \log(x) = 0$.

Entropy quantifies the uncertainty associated with a random variable (MacKay, 2002, Chapter 2.4). It is maximised when all outcomes are equally likely and equals zero when one outcome has probability 1, i.e. when the outcome is deterministic.

Finally, *conditional entropy* measures the remaining uncertainty of a random variable given that another random variable has been observed (MacKay, 2002, Chapter 9.1).

Definition 2.25 (Conditional entropy)**.** For discrete random variables X and Y, the conditional entropy of X given Y is

$$\mathrm{H}(\, X \,|\, Y \,) \coloneqq \mathbb{E}[\mathrm{I}(\, X \,|\, Y \,)] = -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P(x, y) \log P(\, x \,|\, y \,).$$

2.3.3 Mutual information

Pointwise mutual information (PMI) measures the degree of association between specific outcomes of two random variables (Fano, 1961, p. 28).

Definition 2.26 (Pointwise Mutual Information)**.** Let X and Y be discrete random variables. For outcomes x and y,´ the pointwise mutual information is defined as

$$\mathrm{PMI}(x, y) \coloneqq \log \frac{P(x, y)}{P(x) P(y)}.$$

PMI compares the probability of observing x and y together with the probability expected if the two events were independent. If x and y are independent, then $P(x, y) = P(x)P(y)$ and therefore $\mathrm{PMI}(x, y) = 0$. Positive PMI indicates that the events co-occur more frequently than expected under independence, while negative PMI indicates that they co-occur less frequently (Jurafsky and Martin, 2026, Appendix J).

Corollary 2.4. PMI can also be expressed in terms of surprisal:

$$\mathrm{PMI}(x, y) = \mathrm{I}(x) + \mathrm{I}(y) - \mathrm{I}(x, y).$$

Proof.

$$\begin{aligned}\mathrm{PMI}(x, y) &\overset{\text{(Def. 2.26)}}{=} \log \frac{P(x, y)}{P(x)P(y)} = \log P(x, y) - \log P(x)P(y) \\ &= \log P(x, y) - \log P(x) - \log P(y) \overset{\text{(Def. 2.22)}}{=} -\mathrm{I}(x, y) + \mathrm{I}(x) + \mathrm{I}(y). \quad (3)\end{aligned}$$

□

Thus PMI measures how much less surprising the joint occurrence of x and y is compared to what would be expected if the events were independent.

While PMI measures the association between individual outcomes, *mutual information* measures the overall statistical dependence between two random variables (Cover and Thomas, 2006, Chapter 1.1).

Definition 2.27 (Mutual Information)**.** Let X and Y be discrete random variables. The mutual information between X and Y is defined as

$$\mathrm{I}(\, X \,;\, Y \,) \coloneqq \sum_{x,y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)}.$$

Mutual information is always non-negative and equals zero if and only if X and Y are independent.

Corollary 2.5. Mutual information can be interpreted as the expected pointwise mutual information:

$$\mathrm{I}(\, X \,;\, Y \,) = \mathbb{E}_{X,Y}[\mathrm{PMI}(X, Y)]\,.$$

Proof.

$$\mathbb{E}_{X,Y}[\mathrm{PMI}(X, Y)] \overset{\text{(Def. 2.26)}}{=} \sum_{x,y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)} \overset{\text{(Def. 2.27)}}{=} \mathrm{I}(\, X \,;\, Y \,) \quad (4)$$

□

Corollary 2.6. Equivalently, mutual information can be expressed in terms of entropy:

$$\mathrm{I}(\, X \,;\, Y \,) = \mathrm{H}(X) - \mathrm{H}(\, X \,|\, Y \,).$$

Proof.

$$\begin{aligned}\mathrm{H}(X) - \mathrm{H}(\,X \,|\, Y\,) &\overset{\text{(Def. 2.24,2.25)}}{=} \mathbb{E}[\mathrm{I}(X)] - \mathbb{E}[\mathrm{I}(\,X \,|\, Y\,)] = \mathbb{E}[\mathrm{I}(X) - \mathrm{I}(\,X \,|\, Y\,)] \\ &= \mathbb{E}\left[-\log \frac{P(X)}{P(\,X \,|\, Y\,)}\right] = \mathbb{E}\left[\log \frac{P(\,X \,|\, Y\,)}{P(X)}\right] = \mathbb{E}\left[\log \frac{P(X,Y)}{P(X)P(Y)}\right] \\ &= \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \overset{\text{(Def. 2.27)}}{=} \mathrm{I}(\,X\,;\,Y\,) \quad (5)\end{aligned}$$

□

2.3.4 Cross-entropy and Kullback-Leibler divergence

Cross-entropy measures the expected surprisal of a distribution Q when the data is generated according to another distribution P.

Definition 2.28 (Cross-Entropy)**.** Let P and Q be probability distributions over the same sample space $\mathcal{X}$. Following Dezert and Dambreville (2023), cross-entropy between P and Q is defined as

$$\mathrm{H}(P,Q) \coloneqq \mathbb{E}_{X\sim P}[\mathrm{I}_Q(X)] = -\sum_{x\in\mathcal{X}} P(x) \log Q(x),$$

where $\mathbb{E}_P[\cdot]$ gives the expected value with respect to distribution P.

The final important quantity discussed here is the *Kullback-Leibler divergence* (KL divergence) (Kullback and Leibler, 1951). It measures the difference between two probability distributions.

Definition 2.29 (Kullback–Leibler divergence)**.** Let P and Q be probability distributions over the same sample space $\mathcal{X}$. The KL-divergence between Q and P is defined as

$$\mathrm{D}_{\mathrm{KL}}(\,P \,\|\, Q\,) \coloneqq \mathbb{E}_{X\sim P}\left[\log \frac{P(X)}{Q(X)}\right] = \sum_{x\in\mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}.$$

The measure can be interpreted as the expected additional surprisal when approximating the true distribution P using distribution Q. It is always non-negative and equals zero if and only if $P = Q$. It carries a notion of distance between two distributions, however, it is not symmetric.

Corollary 2.7. KL-divergence is closely related to entropy and cross-entropy. In particular, it can be composed as

$$\mathrm{D}_{\mathrm{KL}}(\,P \,\|\, Q\,) = \mathrm{H}(P,Q) - \mathrm{H}(P).$$

Proof.

$$\begin{aligned}\mathrm{H}(P,Q) - \mathrm{H}(P) &\overset{\text{(Def. 2.28,2.24)}}{=} \mathbb{E}_{X\sim P}[\mathrm{I}_Q(X)] - \mathbb{E}_{X\sim P}[\mathrm{I}_P(X)] = \mathbb{E}_{X\sim P}\left[-\log \frac{Q(X)}{P(X)}\right] = \\ &\mathbb{E}_{X\sim P}\left[\log \frac{P(X)}{Q(X)}\right] \overset{\text{(Def. 2.29)}}{=} \mathrm{D}_{\mathrm{KL}}(\,P \,\|\, Q\,) \quad (6)\end{aligned}$$

□

3 Reading times

Reading-time measures provide a central empirical basis for the study of human sentence processing, offering a signal of comprehension difficulty during reading. In the following, I first give a general methodological overview of reading-time research, outlining key experimental distinctions and assumptions. I then discuss the two most widely used paradigms in detail: eye-tracking, which records natural eye movements during reading, and key-press methods, in which reading times are derived from controlled self-paced presentation.

3.1 Methodological overview

The empirical investigation of human sentence processing often relies on neural and behavioural measures that are linked to the difficulty of comprehension. A family of such measurements tracks the process of reading: subjects are presented text and different kinds of responses are recorded and assigned to predefined regions of the input material while they read. According to Haberlandt (1994), reading implies the extraction of meaning from a text and the construction of representations in the mind. For more information on the nature of these representations, see Section 4. Consequently, reading times have played a central role in testing theories and computational models of language processing.

Various experimental techniques have been developed for taking neural and behavioural measures during reading. Commonly used methods include eye-movement recording, key-press measures, *Event-related potentials* (ERPs) using *electroencephalography* (EEG) and *functional Magnetic Resonance Imaging* (fMRI) (Carreiras et al., 2004). These paradigms differ substantially in the type of information they provide and in the degree to which they allow natural reading behaviour. Some techniques primarily measure neural activity associated with language processing, whereas others record behavioural responses that accompany reading.

Reading paradigms differ along several methodological dimensions. First, reading time is either controlled externally or determined by the participants themselves. In timed paradigms such as *fixed-rate rapid serial visual presentation* (RSVP), words are presented to the readers sequentially at a fixed rate in the centre of a screen (Ditman et al., 2007). This approach is often used in ERP experiments because it ensures that differences in neural responses cannot be attributed to variations in exposure time (Ditman et al., 2007). However, such presentation methods deviate substantially from natural reading conditions. In particular, RSVP prevents readers from controlling their reading speed and from performing processes such as rereading that are observed in unconstrained settings (Just et al., 1982).

In contrast, self-paced paradigms allow participants to read at their own speed. The guiding assumption behind such measurements is that the time readers spend processing a word or region reflects the cognitive effort required to integrate that information into the developing interpretation of the text (Haberlandt, 1994). Self-paced paradigms therefore provide an indirect measure of comprehension processes during reading. Two widely used methods in this category are eye-tracking, which records eye movements during reading and key-press paradigms such as moving-window or stationary-window self-paced reading (Haberlandt, 1994).

Second, measurements may either be naturalistic or rely on motor responses. Naturalistic paradigms record behaviour that occurs during reading, such as eye movements or neural ac-

tivity, whereas other frameworks derive their measurements from explicit actions performed by the participant (e.g. key presses) (Haberlandt, 1994). In the latter case, the response time includes motor execution in addition to cognitive processing.

A further distinction concerns whether a measurement captures the process of comprehension itself or only its final outcome. So-called *online measures* record behavioural or neural responses while participants are reading, thereby providing information about the temporal dynamics of language processing. In contrast, *offline measures* such as comprehension questions or acceptability judgments are collected after reading has finished and therefore only reflect the final interpretation constructed by the reader (Sekerina et al., 2008). One type of offline experiment deserves additional attention: cloze experiments yielding predictability ratings such as those performed by Luke and Christianson (2018). You can find a description of this procedure in Section 4.2.

Language processing is most commonly investigated using online tasks due to the assumption of a direct connection between slow-downs in reading and comprehension processes (Haberlandt, 1994). Historically, many reading experiments were conducted in highly controlled laboratory settings designed to test specific linguistic hypotheses. In such studies, participants typically read isolated sentences or short texts containing carefully manipulated linguistic structures (Wolfer, 2015). While this approach allows precise control over experimental variables, it captures only a limited portion of the variability present in natural reading.

With the development of computational models of language processing, a need emerged for larger and more naturalistic datasets that allow model evaluation on more diverse linguistic material. This has led to the collection of several large eye-tracking and self-paced reading corpora containing measurements obtained on longer texts or entire narratives (Cop et al., 2017). They provide valuable resources for studying reading behaviour and for evaluating quantitative models of human sentence processing. Due to their wide availability and empirical success, I will focus on eye-tracking and key-press methods in this work.

Methodologically, researchers typically derive linguistic predictors (e.g. frequency, surprisal, or syntactic structure) from the stimulus text and related them to the observed behaviour in order to test explanations for it. In this sense, analyses based on naturalistic reading corpora correspond to observational/correlational studies (see Shadish et al., 2002), since the independent variables are not manipulated experimentally by the researchers but arise from the naturally occurring variation in the text shown to all participants.

In the following sections, first, I will discuss important considerations with respect to reading time variability between readers and between reading tasks. Then I will present eye-tracking and key-press methods in more detail and give a comprehensive overview of large general-purpose English-language reading time corpora

3.1.1 Variability between readers

Reading behaviour exhibits considerable variability across individuals. Large differences are observed in overall reading speed, eye movement patterns and strategies employed during reading. These differences are influenced by factors such as age and cognitive capacity. For instance, children and old adults show a higher frequency of rereading (Rayner, 2015; Kemper and Liu, 2007). Reading skill also plays an important role: skilled readers typically read faster and possess a wider perceptual span, while unskilled readers read slower and tend to process a narrower region of text

at any point in time (Rayner et al., 2010). As a consequence, investigating reading speed, the likelihood of skipping words and other characteristics that vary depending on the reader requires special statistical tools (see Section 6) and sensible considerations with respect to the sample size of reading time experiments to be able to capture general population-level effects.

The corpora presented in Sections 3.2 and 3.3 contain measurements collected from regularly skilled adult readers. Furthermore, the number of participants and statistics with respect to the allocation of the experimental material to these participants will be given.

Finally, a common element of reading experiments is the inclusion of comprehension questions that serve as a control condition to ensure that participants read attentively. These questions are usually designed to be relatively easy, allowing researchers to detect inattentive participants and explude the data they procuded from subsequent analyses (Radach and Kennedy, 2013). Additionally, experiments commonly include practice trials prior to the actual data collection phase so that participants can familiarise themselves with the experimental procedure and task requirements (Cop et al., 2017).

3.1.2 Variability between tasks

Even for the same participant, reading behaviour varies depending on the task they are given in the experimental setting (Radach and Kennedy, 2013). More demanding assignments or tasks requiring detailed comprehension may lead to slower reading and more careful processing. Most general-use datasets aim to approximate what is commonly referred to as *normal* or *natural* reading (Radach and Kennedy, 2013). According to Haberlandt (1994), reading involves extracting meaning from text and constructing mental representations of its content. Experiments that aim to study these processes therefore attempt to minimise the influence of additional cognitive demands unrelated to comprehension (Carreiras et al., 2004) such as decision processes. In practice, participants are typically instructed to read the presented texts either without a specific goal or with the goal of understanding their content (see Sections 3.2 and 3.3).

The notion of natural reading is not precisely defined and remains the subject of debate (Haberlandt, 1994). For instance, experimental settings inevitably differ from everyday reading situations in several respects, including the artificial environment of the laboratory and the type of textual material presented. Nonetheless, the corpora presented in Sections 3.2 and 3.3 are designed to adhere to this informal standard. Information on the tasks given to the participants as well as a brief overview of the experimental protocols as provided by the datasets' documentations will be given.

3.2 Eye-tracking

Eye-tracking (ET) refers to the recording of the position and movement of the eyes. Modern eye-tracking systems measure the position of one or both eyes and record the location and duration of gaze fixations over time (Haberlandt, 1994). Because eye movements during reading are closely linked to ongoing cognitive processing, ET has become one of the central methods in psycholinguistic research on sentence comprehension.

3.2.1 Historical background

Research on eye movements has a long history described in great detail by Wade (2010). Researchers have realised early on that our eyes do not move smoothly across text when reading but that they proceed in a ragged manner, focusing for some time on one position and then swiftly moving to the next place.

Reading research developed at the end of the 19th century: Émile Javal, founder of the ophthalmology laboratory at Sorbonne, Paris, used the term *saccade* (Javal, 1878), meaning "jerk" or "twitch" in French, for the characteristic movement of the eye. This was later adopted by the general field and became an established scientific term (Wade, 2010). Following Wade (2010), two other pioneers of reading research concerned with eye movements are Ewald Hering and M. Lamare. They observed the sound of muscular movements during reading of text, making it possible to record the number of saccades per line (Hering, 1879; Lamare, 1892).[3]

Subsequent research involved attempts to accurately record the direction of eye movements. According to Wade (2010), Javal failed in recording them on a smoked drum by attaching a pointer to the eye and using mirrors. Huey (1898) and Delabarre (1898) first achieved this using an eye-cup attached to an aluminium rod transcribing the movements onto a rotating drum (Hutton, 2019). This was previously tried unsuccessfully by Raehlmann (1878) for pathological eye-movements and by Ahrens (1891) for reading. It required the use of anasthetics for inserting the eye-cups. Other, similarly uncomfortable devices followed soon (Wade, 2010)

A major methodological breakthrough occurred with the development of non-invasive photographic recording methods. Dodge and Cline (1901), Dodge (1903), and Dodge (1904)'s device could trace eye movements without the need for an eye-cup or a direct connection between recording surface and eye. However, specialised contact lenses remained in wide use throughout the 20th century (Hutton, 2019).

Today, researchers most commonly rely on highly precise commercial video-based eye-trackers (Hutton, 2019). They will be discussed in more detail in Section 3.2.3.

3.2.2 Eye movements during reading

Eye movements during reading consist primarily of two types of events: *fixations* and *saccades*. A fixation refers to a short period during which the eye remains relatively stable and visual information is acquired (Rayner, 2015). In contrast, saccades are rapid movements that shift the point of gaze from one location to another. During saccades, visual perception is strongly suppressed, meaning that linguistic information is primarily obtained during fixations (Rayner, 2015).

Fixations typically last around 200 ms to 250 ms, whereas saccades are much shorter, taking approximately 20 ms to 40 ms (Haberlandt, 1994). In English reading, saccades usually move the eyes roughly seven to ten letters to the right (Hyönä and Kaakinen, 2019). Readers frequently skip short or predictable words and may fixate a word more than once, possibly at different locations in the word (Rayner, 2015). Additionally, the eyes sometimes move backwards in the text to previously read material. Such backward movements are called *regressions* and occur in roughly 10 % to 15 % of fixations (Rayner, 2015).

[3] Lamare (1892) was published 13 years after the experiments were initially conducted according to Wade (2010).

These eye-movement patterns arise from biological properties of the human visual system. The region of highest sharpness is called the *fovea*, a small area at the centre of the retina (Pouget, 2019). Around it lies the *parafoveal* region[4] with reduced acuity, followed by the *peripheral* region with substantially lower resolution and increased blurriness (Rayner, 2015). Because only a small part of the visual field can be processed with high precision at a time, readers must move their eyes frequently in order to bring successive parts of the text they want to read into the foveal region (Rayner, 2015).

According to Rayner (2015), the *perceptual span* is dynamic, based on attention and dependent on the informational density of the writing system and asymetric. In alphabetic writing systems, it ranges from approximately 3-4 letters to the left of the fixation to roughly 14-15 letters to the right (or reversed for writing systems read from right to left), depending on reading skill Rayner (2015). Furthermore, the field concluded that no information is gained from the lines of text above or below the currently fixated one (Rayner, 2015).

3.2.3 Eye-tracking technology

Modern eye-tracking systems typically employ video-based technology. A high-speed camera records the eye looking at a screen while an infrared light source creates a corneal reflection that can be detected by image-processing algorithms (Hutton, 2019). Using calibration parameters obtained at the beginning of the experiment, the system computes the gaze position on the screen for each recorded frame (Hutton, 2019). Depending on the device, sampling rates range from several hundred to over one thousand measurements per second.

Because head movements affect the mapping between the eye and the screen coordinates, traditional eye-tracking setups often use chin and forehead rests or bite bars to stabilise the participant's head (Wade, 2010). More recent systems compensate for head movements computationally by tracking both pupil position and corneal reflections or by employing wearable recording devices that capture head movements (Hutton, 2019).

3.2.4 Measures derived from eye movements

Raw eye-tracking recordings consist of time series of gaze coordinates sampled at high temporal resolution. These samples are subsequently segmented into periods of fixation and saccade using event detection algorithms (Nyström and Holmqvist, 2010). The resulting fixations are then mapped to regions of interest such as words, from which reading-time measures are derived. When investigating language processing, saccade durations are usually ignored due to their shortness and the fact that no visual information is obtained during movement of the eye (Rayner, 2009).

Several standard measures are commonly used in reading research:

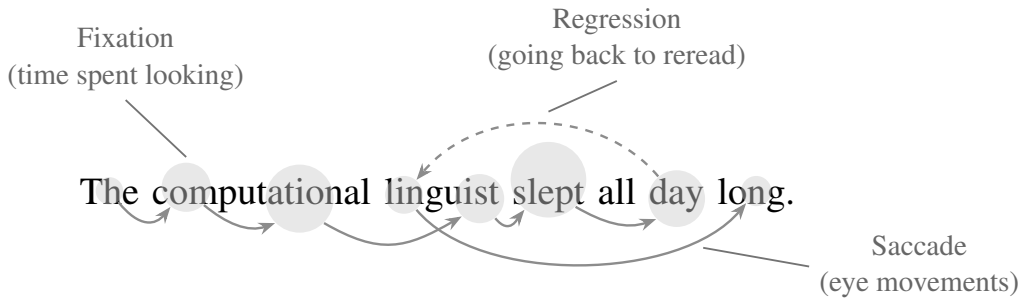
- *First fixation duration* (FFD) refers to the duration of the first fixation on a word (Rayner, 2015).
- *Gaze duration* (GD) is defined as the sum of all fixations on a word (possibly on different regions of it) from the first fixation until the gaze leaves to any other word for the first time

[4] from fovea centralis, the central pit in the innermost tissue layer of the eye, filled with photoreceptor cells responsible for sharp vision (Klein et al., 2019)

(Rayner, 2015). Thus, the measure consists of the duration of the first fixation and the time of all immediately following refixations.

- *Go-past time* (GPT), also called *regression path duration*, includes all fixation durations from the first fixation on a word until the gaze moves to the next word located to the right, including regressions to earlier parts of the sentence (Radach et al., 2008).

You can find a visualisation of these measures for an example sentence in Figure 1. They are thought to reflect different stages of language processing. According to Radach and Kennedy (2013), first fixation duration is assumed to be determined by early prelexical processing such as orthographic recognition while gaze duration is thought to include later stages such as lexical access. Measures that include regressions such as GPT capture later integration processes of meaning during sentence comprehension. Thus, they are often the go-to measure for high-level comprehension research (Radach and Kennedy, 2013).



FFD	90	260	100	250	0	140	90
GD	90	520	100	250	0	140	90
GPT	90	520	100	250	0	210	90

Figure 1: Schematic eye-movement record with fixation overlays and saccade/regression paths. Note that saccades are rather direct in reality. Curved lines are used here for reasons of readability. Illustrative first fixation duration, gaze duration and go-past time measures are shown below the word they are associated with.

3.2.5 Methodological considerations

Besides the choice of eye-tracking device and how to account for head-movements, there are other factors important to keep in mind when comparing and replicating eye tracking experiments. This includes the size and font of the text, the distance from the screen and the software used to control the experiments (Siegelman and Kuperman, 2025).

Additionally, eye-tracking recordings typically require several preprocessing steps before analysis. Fixations must be mapped to the corresponding words in the stimulus text and errors caused by calibration drift or recording noise need to be corrected, especially if working with multiple lines of text (Carr et al., 2022). Traditionally, such corrections were often performed manually (Siegelman et al., 2022), leading to a high researcher degree of freedom. More recently, automated alignment algorithms implemented in tools such as the `popEye` package (Schroeder, 2019) have become widely used, enabling more standardised processing pipelines across different datasets.

Corpora documentations specify these aspect to varying degrees with the field having gained increasing awareness on the importance of standardised and replicable approaches in recent years (see for instance the documentation for the ZuCo 1.0 corpus, Hollenstein et al. (2018)).

3.2.6 Established findings

Eye-movement measures have revealed a number of robust findings about reading behaviour and language processing. For instance, shorter and more frequent words tend to be read faster than longer or less frequent words Just and Carpenter (1980). Words that are highly predictable given their context are also processed more quickly (Zola, 1984).

Another well-known phenomenon is the *wrap-up effect*: fixation durations tend to increase at clause boundaries and especially at sentence-final positions. This pattern is commonly interpreted as reflecting postponed syntactic integration processes in which readers combine information from earlier parts of the sentence (Just and Carpenter, 1980; Rayner et al., 2000). Furthermore, researchers frequently observe *spillover* or *lag effects*: processing difficulty associated with a word can influence not only the fixation durations on that word itself but also those of subsequent words (Radach and Kennedy, 2013; Rayner, 2015; Kliegl et al., 2006).

On the other hand, research have also found that readers extract information from upcoming words in the parafovea most strongly from the next word ($n+1$) (Schotter et al., 2011). This is called *preview effect*. Studies show that orthographic and phonological information about upcoming words can be obtained before fixation and lexical properties such as word frequency may also influence this early processing (Rayner, 1998; Schotter et al., 2011; Radach and Kennedy, 2013). Whether readers extract semantic information from parafoveal preview remains debated, with mixed empirical findings (Hohenstein et al., 2010; Schotter et al., 2011). Following Radach et al. (2013), under favourable conditions, preview effects can occasionally extend beyond the next word to $n + 2$, although such effects are typically a lot weaker.

These findings have helped establish and refine various theories of sentence processing that aim to give a mechanistic account of reading times driven by language comprehension. Section 4 will discuss several of the most relevant approaches.

3.2.7 Eye-tracking corpora

In recent years, a large number of corpora have been created that record the eye movements of participants reading extended texts. In contrast to traditional psycholinguistic experiments that employ carefully controlled stimulus sentences, these datasets aim to capture reading behaviour in more natural settings. This allows researchers to test whether model-derived predictors generalise beyond specific experimental constructions. As a result, they have become an important resource for evaluating computational models of human sentence processing (Cop et al., 2017).

In the following, I describe a selection of widely used English-language eye-tracking datasets for reading in the order of their publication. For each corpus, I report key characteristics of the data collection procedure, including the type of stimulus material, the number of participants, presentation format and available additional annotations. If specified in the documentation, further information such as comprehension checks, recording devices and experimental protocols are also included. Note that while most datasets are recording using one of two eye-tracking devices, cal-

	Dundee	UCL (ET)	GECO†	Provo	ZuCo 1‡	ZuCo 2§	MECO 1	MECO 2
Natural text	natural	modified*	natural	natural	natural	natural	constructed	constructed
Characteristics		frequent words single sentences			≥ 1 relation		Wikipedia-based	
Participants	10	43	14	84	12	18	46	50
Observations		72 502	702 506	212 436	72 729	112 941	80 564	69 994
ROIs		1726	51 210	2529	7794	6478	2006	1635
Subjects per ROI		42	13.7	84	9.33	17.4	40.2	40.2
ROIs per subject		1686	50 179	2529	6061	6274	1751	1315
Avg. sentence length		9.42	11.4	19.9	27.0	19.6	20.9	19.2
Avg. ROI length		3.90	4.39	4.70	4.99	4.89	5.14	5.18
Shuffled	no	yes	no	yes	no	no	no	no
Additional data	cloze	SPR		cloze	EEG	EEG		
Publicly available	no	yes	yes	yes	yes	yes	yes	yes

Table 1: Eye-tracking datasets of "normal" reading in English; all provide GPT, FFD and GD (+various additional metrics). The number of participants excludes trials with failed comprehension tests. Datasets are marked as shuffled if sentences or small paragraphs were shown in random order or without context to the participants. For unavailable corpora, numbers are adopted from documentation or the curators' paper – see the respective sections for the more information. ROI length is measured by number of characters; sentence length is determined by splitting the datasets using `spacy` (Honnibal et al., 2020). *replacement of proper names; †English monolingual; ‡task 2; §task 1; cloze: predictability norms collected through a cloze task; SPR: moving-window self-paced reading times; ROI: region of interest (usually a word plus punctuation).

ibration settings and preprocessing steps can still differ and complicate studies that make use of multiple corpora. Table 1 provides a concise comparison of the datasets discussed in this section. In addition to information reported in the original publications, I include several unified statistics that were extracted manually from the corpora where access to the data was possible. To my knowledge, this constitutes the first comprehensive comparison of general-purpose eye-tracking corpora for reading. Such an overview may help researchers select appropriate datasets for evaluating computational models of language processing. Links to freely available corpora are provided in footnotes.

3.2.7.1 Dundee corpus

The Dundee corpus (Kennedy, 2003; Kennedy et al., 2003) is one of the earliest eye-tracking datasets collected for natural reading. It contains eye-movement recordings obtained while participants read newspaper editorials from *The Independent* and from the French newspaper *Le Monde*. The English portion of the corpus consists of 20 articles containing 2368 sentences and 51 502 word instances (Barrett et al., 2015). Ten native speakers of English participated in the experiment.

The text was presented in segments of five lines at a time in their original order (Kennedy et al., 2013). Unfortunately, no information is readily available on whether each participant read all of the segments. Eye movements were recorded from the right eye using a *Dr Bouis Oculometer Eye Tracker*, while participants' heads were stabilised with bite bars (Kennedy et al., 2013). According to Barrett et al. (2015), the dataset provides detailed information on fixation durations, fixation

order and landing positions at the word level.

In addition to eye-movement data, the Dundee corpus includes predictability estimates for each word obtained through a cloze task (see Section 4.2.1.1). It has been widely used in research on sentence processing. For example, Mitchell et al. (2010) investigate joint syntactic and semantic measures of processing difficulty, Fossum and Levy (2012) compare sequential and hierarchical syntactic models and Kennedy et al. (2013) analyse spillover effects in reading.

Unfortunately, the Dundee corpus is not freely available for download. Researchers interested in using it are advised to contact Alan Kennedy.

3.2.7.2 UCL corpus

The *University College London* (UCL) *corpus* was introduced by Frank et al. (2013). It contains 361 sentences derived from three novels written by novice authors. The sentences were selected to consist only of terms from a list of 7754 frequent words and can be interpreted without broader discourse context. Proper names were replaced to prevent participants from reconstructing the original narrative. The rationale is to provide measurements that reflect factors of processing on the sentence-level, thus controlling for extra-sentential influences (Frank et al., 2013).

The dataset includes both eye-tracking and self-paced reading measurements. Eye-tracking data were collected for a subset of 205 sentences that could be displayed on a single line. A total of 43 English-speaking participants took part in the experiment, the majority of whom were native speakers. Participants rested their heads on a chin rest while their eye movements were recorded with a head-mounted *SR Research EyeLink II* eye tracker (Frank et al., 2013). Sentences were presented in random order and divided into six blocks, with breaks between blocks. The length of the brakes was determined by the participants.

Because the stimuli were constructed to minimise extra-sentential influences, the corpus has been widely used for evaluating computational models of sentence processing and cognitive load in the working memory (e.g. Frank, 2017; Merkx and Frank, 2021; Aurnhammer and Frank, 2019). However, the experimental setting of presenting unrelated sentences in short succession can be considered rather unnatural. The dataset is freely available for download.[5]

3.2.7.3 GECO

The *Ghent Eye-Tracking Corpus* (GECO) was introduced by Cop et al. (2017). It contains eye-movement recordings from both monolingual English readers and bilingual English-Dutch readers. GECO is the first eye-tracking corpus in which participants read an entire novel, making it particularly valuable for the study of reading behaviour in extended discourse. It is also one of the first large corpora to investigate bilingual reading systematically, reporting the participants' language skills Cop et al. (2017).

In the following, I focus on the monolingual English portion of the corpus, since the research questions pursued in this work do not concern bilingualism. It contains recordings from 14 English-only speakers reading *The Mysterious Affair at Styles* by Christie (1920). In total, the material comprises 5031 sentences. Participants were instructed to minimise head movements during the

[5] https://link.springer.com/article/10.3758/s13428-012-0313-y#SecESM1

recording and eye movements were recorded from the right eye using a desktop-mounted *SR Research EyeLink 1000* eye tracker. The text was presented in paragraphs, with at most 145 words shown on the screen at a time. Participants read the novel across four sessions, each covering three to four chapters and lasting approximately 1.5 hours. After each chapter, comprehension questions were administered. Participants were informed in advance that such questions would follow, in order to encourage attentive reading.

GECO has mainly been used in research comparing first- and second-language reading (e.g. Cop et al., 2015a; Cop et al., 2015b). However, because it provides eye-movement data for extended naturalistic reading, it is also well suited for evaluating more general theories and computational models of language processing. It is freely available for download.[6]

3.2.7.4 Provo corpus

The *Provo corpus* was collected by Luke and Christianson (2018). A total of 84 native English speakers took part in the experiment, with each of them reading 55 short paragraphs drawn from a variety of sources, including news articles, science magazines and fiction. The paragraphs were presented in random order and contained on average approximately 2.5 sentences each (Luke and Christianson, 2018).

Eye movements were recorded using an *SR Research EyeLink 1000 Plus* eye tracker. Participants' head movements were reduced using a chin and forehead rest and gaze position was recorded from the right eye. Unlike most other corpora discussed in this section, the Provo corpus does not include comprehension questions to identify inattentive trials (Luke and Christianson, 2018).

Similar to the Dundee corpus, the Provo dataset provides predictability estimates for each word obtained through a cloze task. Furthermore, it includes estimates of morpho-syntactic and semantic predictability, used by Luke and Christianson (2016) to investigate the graded nature of prediction during reading. Due to its randomised paragraph design, Provo represents a middle-ground between the strict control of extra-sentential context by showing only single sentences as in the UCL corpus and letting participants read whole connected texts as in GECO. As such, the corpus has been used to evaluate general models of human language processing (e.g. Luke and Christianson, 2016; Cevoli et al., 2022; Rego et al., 2024). It is freely available for download.[7]

3.2.7.5 ZuCo

The *Zurich Cognitive Language Processing Corpus* (ZuCo) is a multimodal dataset that pioneered combining eye-tracking measurements with *electroencephalography* (EEG) recordings in order to study the relationship between eye movements and brain activity during reading (Hollenstein et al., 2018; Hollenstein et al., 2020). The first version of the corpus (ZuCo 1.0) was released in 2018 and contains simultaneous eye-movement and EEG recordings from 12 native English speakers (Hollenstein et al., 2018). Participants performed three reading tasks: sentiment rating, normal reading and answering yes/no questions about semantic relations. In the following, I will focus on the normal reading task.

The stimulus material was derived from Wikipedia articles annotated with semantic relations by Culotta et al. (2006). The articles were selected to contain a relatively high number of relations,

[6] https://expsy.ugent.be/downloads/geco
[7] https://osf.io/sjefs

which resulted in a large proportion of texts concerning well-known individuals. From this material, Hollenstein et al. (2018) sampled 300 sentences containing at least one relation. This setting differs from datasets such as the UCL corpus where the effect of proper names is controlled for by replacing them with generic and frequent names. Researchers who aim to evaluate a model of language processing on ZuCo should therefore assess whether it has broad coverage of real-world entities to mitigate conflating factors.

All participants read the full set of sentences in the same order. Only one sentence was displayed on the screen at a time, with a maximum of 13 words per line and up to five lines (Hollenstein et al., 2018). Comprehension was assessed using multiple-choice questions presented after 68 of the sentences. Eye position and pupil size were recorded using an *SR Research EyeLink 1000 Plus* eye tracker.

The second version of the corpus (ZuCo 2.0) extends the dataset with recordings from 18 native English speakers and improves the reliability of the experimental process (Hollenstein et al., 2020). The data includes both normal reading and relation rating tasks and was collected on the same day with the goal of providing a more controlled comparison between natural and task-specific reading conditions. In contrast, the corresponding tasks in ZuCo 1.0 were conducted on different days so that influence of eternal factors on reading performance could not be ruled out (Hollenstein et al., 2020). Additionally, ZuCo 2.0 conducts the recordings in blocks of approximately 50 sentences, with the two tasks alternating (Hollenstein et al., 2020). Other aspects of the experimental setup correspond largely to those used in ZuCo 1.0.

ZuCo 2.0 contains 349 sentences drawn from the same Wikipedia dataset. The sentences were selected to be comparable in length and reading difficulty to those used in ZuCo 1.0. 100 sentences occur in both ZuCo 1.0 task 2 and ZuCo 2.0 task 1 (Hollenstein et al., 2020). Unfortunately, not all of these duplicates are one-to-one correspondences and the authors did not provide any labelling to filter them. Both versions of ZuCo are available for download.[8,9]

3.2.7.6 MECO

The *Multilingual Eye-movement Corpus* (MECO) is an international collaborative project that collects eye-tracking data for reading in more than 30 languages (Siegelman et al., 2022; Siegelman et al., 2025). The initiative was motivated by the historical bias of eye-tracking research for English and a small number of European languages. In addition to native-language reading, MECO also provides data for English second-language reading (Kuperman et al., 2023; Kuperman et al., 2025), which I do not discuss here. Among the 13 languages included in the first wave of MECO (Siegelman et al., 2022) is an English L1 (first language) experiment conducted in Canada. The second edition of the project (Siegelman et al., 2025) provides an additional set of data – again comprising 13 language – including English L1 data collected in the United Kingdom. The two English subsets shall be the focus of the following description.

Although the data was collected in multiple laboratories worldwide, the experimental procedure was standardised across sites. Participants read 12 custom-written encyclopedic texts inspired by Wikipedia articles. After each text, four yes/no comprehension questions were presented (Siegelman et al., 2022; Siegelman et al., 2025). Head movements were restricted using a chin rest and

[8] https://osf.io/q3zws
[9] https://osf.io/2urht

head rest. Eye movements were recorded from the right eye using an *SR Research EyeLink 1000* or *EyeLink 1000 Plus* eye tracker. The order of the texts was fixed across participants.

For preprocessing and correction of fixation locations, the researchers used automated algorithms implemented in the `popEye` package by Schroeder (2019), which helps reduce variability introduced by manual alignment procedures (Siegelman et al., 2022). Finally, trials and participants with poor-quality recordings were excluded from the dataset.

The first wave of MECO contains recordings for 99 sentences made up of 2109 words read by 46 native English speakers. The second wave includes 104 sentences with a total of 2107 words read by 50 native English speakers. MECO is openly available.[10]

3.3 Key-press methods

Another widely used paradigm for collecting reading-time measurements are *key-press methods*. In these experiments, participants reveal successive parts of a text by pressing a key. The time between two button presses is recorded and interpreted as the reading time associated with the textual unit that was visible during that interval (Haberlandt, 1994). The text is typically divided into words or short phrases, allowing researchers to obtain reading time measurements at a fine level of granularity. Here, word-level reading times shall be of interest. Although eye-tracking experiments are usually also self-paced (see Section 3.1), the term *self-paced reading* (SPR) is conventionally used in literature to denote key-press paradigms, most commonly in their moving-window implementation (e.g. Keller et al., 2009).

3.3.1 Stationary-window presentation

The original key-press paradigm proposed by Aaronson and Scarborough (1976) employed a *stationary-window method* in which words are presented one after another at a fixed location in the centre of the. Each word replaces the previous one once the participant presses a key. The time interval between two key presses is then interpreted as the reading time for the displayed word.

Although this paradigm allows for accurate timing measurements, it has been criticised for its limited ecological validity (Just et al., 1982). Because only a single word is visible at any given time, readers cannot make use of information about the upcoming text. As discussed in Section 3.2.6, in natural reading, readers can extract partial information about neighbouring words through parafoveal vision, including cues such as word length and orthographic structure. Removing this information may therefore affect the reading process – with unknown effect.

3.3.2 Moving-window presentation

The *moving-window method* introduced by Just et al. (1982) addresses this limitation. In this paradigm, the entire sentence is displayed on the screen, but all words are initially masked (typically by sequences of dashes) except for the currently revealed word. Pressing a key masks the current word and reveals the next one. Spaces between words remain visible, allowing readers to

[10] https://osf.io/3527a

perceive the length of upcoming words (Haberlandt, 1994). You can find an illustration of this paradigm in Figure 2.

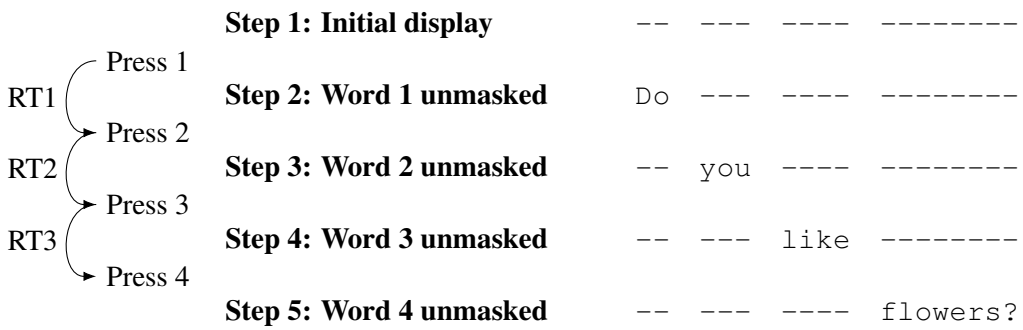


Figure 2: Schematic key-press self-paced reading example. Each line represents a step presented to the participant. A key-press replaces one line with the line below. The time between two successive key-presses is associated with the word that was unmasked during the respective time span.

This design preserves spatial characteristics of natural text while still ensuring that participants process the text in a strictly left-to-right sequence, without regressions to past material. As a result, the moving-window paradigm has become the most widely key-press method in reading time research. The corpora presented in Section 3.3.6 follow this paradigm.

3.3.3 Measures derived from key-press data

Key-press experiments produce a sequence of reaction times associated with successive textual units. If the text is segmented into words, the reading time for word i can be defined as

$$\mathrm{RT}_i = t_{i+1} - t_i, \tag{7}$$

where t_i denotes the time stamp of the key press that reveals word i. These measurements are typically analysed using statistical models that relate reading times to linguistic predictors such as word frequency, predictability, or syntactic structure (Futrell et al., 2018).

Unlike eye-tracking, key-press methods provide only a single reading time per textual unit and do not capture regressions or multiple fixations. This reduces potential for finer grained analyses which would potentially help to identify subcomponents of reading processes but provides an easily interpretable high-level measure that is hypothesised to reflect relatively late stages of processing connected to incremental integration and comprehension (Just et al., 1982).

3.3.4 Methodological considerations

Because key-press paradigms enforce a strictly serial presentation of the text, they differ from natural reading in several respects. Readers cannot skip words or refocus on previously read material. Moreover, parafoveal information is limited to word lengths. These constraints may influence reading strategies and processing behaviour (Rayner, 2015).

On the other hand, key-press methods are widely used due to their simple technical implementation and comparability. Experiments can be conducted using standard computer equipment and do not require specialised hardware such as eye trackers. Software packages such as `Linger` (Rohde, 2001) provide widely used tools for implementing self-paced reading experiments.

3.3.5 Established findings

Self-paced reading experiments have reproduced many of the empirical findings of eye-tracking studies. For example, words that are longer or less frequent tend to elicit longer reading times and words that are more predictable from the preceding context are processed more quickly (Kapteijns and Hintz, 2021; Shain, 2019). Key-press methods also exhibit spillover effects (Smith and Levy, 2013) and wrap-up effects (Just et al., 1982) (see Section 3.2.6 for an explanation). For more fine-grained predictions of sentence processing theories with respect to high-level comprehension processes, data from eye-tracking studies and key-press experiments start to differ. Section 3.3.7 provides a detailed discussion of their comparative advantages and disadvantages for sentence processing research.

3.3.6 Key-press corpora

Several general-use corpora have been created using key-press-based self-paced reading experiments. As in eye-tracking corpora (see Section 3.2.7), these datasets involve reading time measurements for successive words – single sentences or whole texts – rather than isolated areas embedded in specifically crafted and controlled experimental contexts. This makes them valuable resources for testing theories of reading behaviour and evaluating computational models of human sentence processing.

In the following, I describe a selection of widely used English-language self-paced reading corpora in to their order of publication. For each corpus, I report key characteristics of the data collection procedure, including the type of stimulus material, the number of participants, the segmentation of the text and the availability of additional annotations such as predictability estimates. Where possible, further details about the experimental setup and presentation procedure are also included. All corpora presented here were collected using the *moving-window paradigm*. Due to its greater validity (see Section 3.3.2.

Table 2 provides a concise comparison of the datasets described below. In addition to information reported in the original publications, I include several unified statistics that were extracted manually from the corpora where access to the data was possible. Links to freely available corpora are provided in footnotes.

3.3.6.1 MIT corpus

The *Massachusetts Institute of Technology* (MIT) *corpus* introduced by Roark et al. (2009) provides per-word self-paced reading times collected on four short texts with a total of 3534 word instances. The measurements were obtained from 23 native speakers of English. The texts were specifically constructed to contain a high number of syntactically complex constructions, including sentential embeddings, relative clauses and non-local dependencies, while still sounding natural (Roark et al., 2009).

	MIT (SPR)	Brown	UCL (SPR)	Natural Stories
Natural text	constructed	natural	modified*	modified
Characteristics	rare syntax		freq. words, single sentences	rare syntax
Participants	23	32	117	180
Observations			327 912	809 467
ROIs	3534		4596	9779
Subjects per ROI	23		71.3	82.8
ROIs per subject	3534	3912	2803	4497
Avg. sentence length			13.7	21.5
Avg. ROI length			3.95	4.48
Shuffled	no	no	yes	no
additional measures	fMRI while hearing		ET†	
Publicly Available	no	no	yes	yes

Table 2: Datasets of "normal" self-paced reading in English. The number of participants excludes trials with failed comprehension tests. Datasets are marked as shuffled if sentences or small paragraphs were shown in random order or without context to the participants. For unavailable corpora, numbers are adopted from documentation or the curators' paper – see the respective sections for the more information. ROI length is measured by number of characters; sentence length is determined by splitting the datasets using `spacy` (Honnibal et al., 2020). *replacement of proper names; †for a subset of the data that fit on a single line of the display.

The data was collected using the `Linger` software. The text was presented to the participants in fixed order and one line at a time, with an average of 11.5 words per line. Throughout each text, Roark et al. (2009) interspersed ten comprehension questions to ensure attentive reading and held short breaks between them the texts.

The dataset was originally collected to evaluate the predictive power of surprisal and other expectation-based measures of reading difficulty (Roark et al., 2009). According to Roark et al. (2009), a parallel fMRI experiment was also conducted with results being "in preparation". However, it appears that they have never been published. The SPR corpus was later used by Wu et al. (2010) to evaluate complexity metrics derived from an incremental right-corner parser and by Luong et al. (2015) to assess models of memory storage and computation in sentence processing. Unfortunately, the data is not publicly available and appears to be difficult to obtain. According to Brian Roark, the dataset was originally collected by Asaf Bachrach, a co-author of Roark et al. (2009). It is currently unclear whether it is still accessible.

3.3.6.2 Brown

Smith and Levy (2013) collected self-paced reading times for passages drawn from the Brown corpus of American English (Kučera and Francis, 1964), delivering measurements for more natural material than the MIT corpus. The experiment involved 35 native speakers of English. Each participant read several short text passages containing between 292 and 902 words. The total amount of text read varied across participants. However, the description provided by Smith and Levy (2013) does not clearly specify whether the passages presented to different participants were fully disjoint or partly overlapping. Likewise, it is unclear whether the order of the passages was

randomised for each participant.

Comprehension was assessed using questions, which led to the exclusion of three participants who did not meet the required accuracy threshold. Unfortunately, the publication does not specify when these questions were presented during the experiment. The data is mainly used to evaluate lexical predictability effects on reading times (Smith and Levy, 2013; Shain et al., 2022; Pimentel et al., 2023). Despite a remark by Shain et al. (2022), the dataset currently does not appear to be publicly accessible.

3.3.6.3 UCL corpus

The *University College London* (UCL) *corpus* (Frank et al., 2013) was already introduced in Section 3.2.7. In addition to the eye-tracking recordings described there, self-paced reading measurements were collected from 117 native English speakers.

Each participant read a random sample from a selection of 361 sentences, with a trial taking 40 minutes. Thus, the number of sentences read differs by participant depending on how many sentences they managed to read within the available time span. Practice trials were included and after completing a sentence, in approximately half of the cases a comprehension question was shown to ensure attentive reading (Frank et al., 2013).

Because the stimuli were designed to minimise extra-sentential and extra-linguistic influences, the dataset has been widely used for evaluating computational models of sentence processing (e.g. Frank, 2017; Merkx and Frank, 2021; Aurnhammer and Frank, 2019). The dataset is publicly available for download.[11]

3.3.6.4 Natural Stories

Many computational models of human sentence processing make predictions about syntactic constructions that occur only rarely in naturally occurring corpora, making it hard to compare them and to assess their validity. To address this limitation, Futrell et al. (2018) compiled the Natural Stories corpus, which was specifically designed to contain a variety of rare syntactic constructions while still sounding natural.

The corpus consists of ten texts of approximately 1000 words each, derived from publicly available sources such as Wikipedia articles and legends. These texts were edited to incorporate syntactic phenomena including subject- and object-extracted relative clauses, structural ambiguities and sentential subjects. The resulting dataset contains 485 sentences with a total of 10 245 word instances (Futrell et al., 2018).

Self-paced reading measurements were collected from 181 native English speakers using the crowdsourcing platform *Amazon Mechanical Turk* – a procedure that sets it apart from the other corpora discussed here. Participants differed in the number of texts they read. Each text was followed by six comprehension questions to ensure attentive reading and Trials with fewer than five correct answers or with extremely short or long reading times were excluded from the dataset (Futrell et al., 2018). Information about the syntactic constructions in each sentence is available via a spreadsheet that is bundled with the dataset.

[11] https://link.springer.com/article/10.3758/s13428-012-0313-y#SecESM1

Due to its focus on syntactically rare phenomena, the Natural Stories corpus is particularly suitable for the evaluation of theories targeting working memory and syntactic analysis. However, the authors released an updated version of the dataset in May 2025[12] after discovering that the original version contained a systematic misalignment error in which each token was assigned the reading time of the following token. Consequently, results based on earlier versions of the corpus such as Wilcox et al. (2020) should be interpreted with caution.

3.3.7 Comparing the paradigms

Eye-tracking and moving-window self-paced reading are both widely used methods for obtaining word-level reading-time measurements. In both paradigms, the duration associated with a word is interpreted as an indicator of the processing effort required to integrate that word into the developing sentence representation. Despite this similarity, the two methods differ in several important respects. Eye-tracking provides highly granular measurements of the reading process, including multiple fixations within a word, regressions to earlier parts of the sentence, and the precise spatial trajectory of eye movements (Section 3.2.4). In contrast, moving-window paradigms yield a single, easy to extract, reading-time measurement per textual unit, doing away with the necessity of choosing a suitable aggregation metric as with eye-tracking. However, they enforce a strictly left-to-right progression through the text. As a result, readers cannot revisit previously presented words. While this is an "unnatural" scenario for reading, it bears similarities with listening tasks where the progression of input is also strictly incremental and non-regressive (Haberlandt, 1994). Yet, if the aim is to observe reading in a listening-like setting, researcher-timed paradigms such as RSVP would be more suitable since the presentation of speech in natural listening is not paced by the listeners themselves.

Empirical comparisons of the two paradigms reveal substantial differences in the resulting measurements. Just et al. (1982) report that reading times obtained with a moving-window paradigm correlate only moderately with gaze durations from eye-tracking experiments (Pearson's $r \approx 0.57$) and that most known effects such as word length and word frequency are lower for SPR than for ET. Furthermore, reading times in the key-press task were approximately 80 % longer than gaze durations. This difference can partly be explained by the fact that readers frequently skip short function words in eye-tracking experiments, resulting in zero fixation durations for those words (Just et al., 1982). However, this factor alone does not fully account for the discrepancy.

Several explanations have been proposed. One possibility is that the motor action required to press a key introduces additional delay. Ferreira and Henderson (1990) argue that reading and the control of eye-movements have a stronger mental connection than reading and pressing a button. Another explanation, suggested by Danks (1986), states that participants delay the processing of new content to a certain extent to be able to increase the speed of button presses to match the speed of eye movements. However Haberlandt (1994) argue that the existence of a correlation between word frequency and window-based reading times as found by Just et al. (1982) contradicts this hypothesis and indicates immediacy of processing.

On the other hand, it is thought that the absence of regressions in moving-window paradigms increases the working-memory demands of the task, since readers cannot return to previously presented material (Witzel et al., 2012). In eye-tracking experiments, regressions often occur

[12] https://github.com/languageMIT/naturalstories

when readers detect syntactically or semantically demanding constructions (Kennedy and Murray, 1984) or inconsistencies (Just et al., 1982). Preventing such rereading may alter the strategies readers used to process the text (Danks, 1986) or even prevent comprehension (Just et al., 1982).

At the same time, the "naturalness" of the eye-tracking paradigm does not mean that measures like GPT cleanly correspond to resources used in cognitive processing. According to Witzel et al. (2012), readers have a lot of freedom to chose a reading strategy which in turn determines whether fixation durations and locations indicate characteristics of online integration of new words into the cognitive representation of the sentence read so far. The stronger experimental control of self-paced reading paradigms may therefore have advantages. Because the presentation of the text is tightly constrained, the range of possible reading strategies is reduced compared to natural reading situations. This may help standardise participants' behaviour across trials and reduce variability in the resulting measurements (Witzel et al., 2012). However, even under these constraints, readers may adopt different strategies, for instance by integrating words immediately after revealing them or by reading faster and temporarily storing them in memory for later integration which would lead to stronger wrap-up effects (Witzel et al., 2012).

Compared to eye-tracking, key-press methods seem appealing since they come with a single, easy to extract measure. The large selection of eye-tracking measures creates a challenge for researchers aiming to simply quantify the duration of processing. On the same time, it also encourages the development of fine-graded theories that explain these complex and natural variables with characteristics of the language processing device (Haberlandt, 1994). The single duration type provided by key-press experiments does not offer this amount of interpretability. However, it is important to keep in mind that eye-tracking research suggests that not all observable patterns are determined by cognitive control of the language processing device. Saccade generation is thought to follow to a certain degree from low-level physiological processes in the oculomotor system (Rayner, 2015; Radach and Kennedy, 2013), as evidenced by the success of saccade generation models such as SWIFT (Engbert et al., 2005).

From a practical perspective, the two paradigms also differ in their technical requirements. Eye-tracking experiments require specialised and often expensive equipment, whereas self-paced reading experiments can be conducted using standard computer hardware and relatively simple software (Just et al., 1982; Haberlandt, 1994), even making it possible to crowdsource the experiment (Futrell et al., 2018). This makes key-press methods significantly easier to deploy.

Finally, it is important to stress that reading-time measurements do not provide direct access to the cognitive mechanisms underlying language processing. Instead, they offer indirect behavioural evidence that must be interpreted through theoretical models of sentence comprehension (Haberlandt, 1994). Such models should be understood as abstractions that attempt to capture regularities observed in behavioural data rather than as literal descriptions of the underlying cognitive architecture. Consequently, the predictive success of a model indicates the degree to which it captures generalisable patterns in observed behaviour, but it does not by itself justify causal claims about the mechanisms of the human language processing system (Haberlandt, 1994). See Box (1976) for a more thorough discussion of models in the scientific method.

Progress in the field depends crucially on the observational methods available for learning about empirical regularities in order to assert the plausibility of said models and to refine them. Computational models of language processing are therefore often evaluated on both eye-tracking and self-paced reading datasets. More recently, multimodal datasets that combine behavioural and neural

measurements have begun to emerge. For example, the ZuCo corpus integrates eye-tracking and *electroencephalography* (EEG) recordings during reading (Hollenstein et al., 2018; Hollenstein et al., 2020). Such datasets may provide additional constraints on language processing models and thus may be essential for further theoretical progress.

4 Human sentence processing

The term *human sentence processing* refers to the cognitive procedures involved in comprehending sentences as they are read or heard (Reichle, 2021, Chapter 4). It occupies an intermediate level of analysis between two related domains. At a lower level, *word identification* concerns the perceptual and lexical processes that map visual or auditory input onto word representations (Reichle, 2021, Chapter 3). At a higher level, *discourse representation* addresses how readers integrate the meanings of multiple sentences into coherent mental models of extended text or conversation (Reichle, 2021, Chapter 5). The broader field of *human language processing* encompasses all three levels of analysis. In practice, however, the boundaries between them are often blurred because sentence comprehension relies both on lexical processing and on discourse-level interpretation.

The following introduction will focus specifically on sentence processing. For more insight into word identification and discourse representation, please refer to Reichle (2021, Chapter 3 and Chapter 5).

4.1 Introductory overview

Research on sentence processing is a central topic in *psycholinguistics*, a field situated at the intersection of *cognitive psychology* and *linguistics*. Cognitive psychology studies the mechanisms underlying human mental processes, such as perception, memory and attention (Reichle, 2021, Chapter 4). Linguistics, by contrast, traditionally focuses on the structural properties of language itself. Many linguistic theories describe language as an abstract formal system of rules that determine the well-formedness and interpretation of sentences (Reichle, 2021, Chapter 4). This distinction is often expressed in terms of *competence* and *performance* (Chomsky, 1965): competence refers to the abstract knowledge of linguistic structure, whereas performance concerns the use of this knowledge in real-time communication under cognitive and perceptual constraints. Sentence processing research investigates precisely these performance constraints and attempts to explain how human comprehenders construct linguistic structure incrementally during comprehension (Reichle, 2021, Chapter 4).

4.1.1 Linguistic structure

Understanding sentence processing requires a brief discussion of the structures that linguistic theories posit to underlie sentences. A central domain is *syntax*, which studies how words combine into larger units. Most syntactic frameworks assume that sentences are organised hierarchically into *constituents*, such as noun phrases or verb phrases, which together form a tree-graph (Chomsky, 1957). Alternative formalisms represent syntactic structure in terms of *dependencies*, where directed relations connect individual words rather than abstract phrase nodes (Tesnière, 1959; Tesnière, 2015). Both approaches capture the fact that sentences contain structural relations that extend beyond linear word order.

An important consequence of hierarchical structure is that sentences can exhibit *structural ambiguity*, where the same sequence of words can correspond to multiple syntactic analyses. Such ambiguities often lead to measurable differences in processing difficulty, with one hypothesis being that readers must select among alternative interpretations (Reichle, 2021, Chapter 4). Psycholin-

guistic theories therefore aim to explain how readers and listeners construct and revise structural representations while processing a sentence.

4.1.2 Representations and activation

Many cognitive accounts of language comprehension assume that the mind constructs internal *representations* of linguistic input at multiple levels, including lexical items, syntactic structures and discourse-level meaning (Haberlandt, 1994). These representations are assumed to be temporarily maintained in memory during processing. Evidence suggests that representations associated with higher-level units, such as propositions or discourse entities, may remain accessible longer than those corresponding to individual words (Haberlandt, 1994)

A widely adopted assumption in cognitive psychology is that mental representations vary in their degree of *activation* (Lewis and Vasishth, 2005; Lewis et al., 2006). Recently processed representations tend to be highly activated and therefore more easily retrieved, whereas representations that have not been recently used gradually lose activation and become harder to access (Reichle, 2021, Chapter 4).

4.1.3 Incrementality, connectedness and predictiveness

A property of human sentence comprehension is that it largely proceeds *incrementally* (Demberg, 2010). Rather than waiting until an entire sentence has been perceived, comprehenders interpret linguistic input, as soon as it becomes available, i.e. word-by-word. Experimental studies using eye tracking and cross-modal paradigms provide broad evidence for this claim (e.g. Tanenhaus et al., 1995; Burmester et al., 2014). Consequently, sentence processing theories formalise syntactic and semantic representations that are constructed over sentence prefixes and updated as additional input arrives.

Incremental processing is closely related to the notion of *connectedness* (Demberg, 2010). During comprehension, newly encountered words tend to be integrated into the partial structure built so far, forming relations with previously processed elements and allowing the interpretation of each new word in the context of the developing structure (Sturt and Lombardo, 2005). Note that some theoretical approaches and formal parsers assume strictly connected incremental parses (e.g. Demberg, 2010) while others argue that connections may remain underspecified when the available information is insufficient to determine the correct analysis, leaving room for later revision or delayed integration (e.g. Swets et al., 2008). This might also be the case when discursive goals do not mandate a precise parse, as predicted by *good enough processing* (Ferreira et al., 2002).

Because sentences unfold over time, comprehenders must interpret each new word under uncertainty about the remainder of the utterance. One strategy for dealing with this uncertainty is *predictive processing*. Numerous studies suggest that readers and listeners form probabilistic expectations about upcoming linguistic material based on the preceding context (Kutas et al., 2011; Levy, 2008). Furthermore, prediction of syntactic components is a logical consequence if assuming strictly connected and incremental syntactic processing (Demberg, 2010). Pickering and Garrod (2007) argue that anticipating likely continuations can make comprehension easier and faster.

Taken together, incrementality, connectedness and predictiveness imply that sentence comprehen-

sion involves continuous inference over partially observed linguistic structures. As a sentence unfolds, comprehenders maintain provisional analyses, integrate new information with existing representations, and update their expectations about possible continuations.

Different theoretical accounts emphasise different aspects of this process. Some theories highlight probabilistic expectations about upcoming words as a central mechanism of comprehension, while others emphasise limitations of the working memory that stores the incremental prefix interpretation and the cost of integrating new material into it. The former are often grouped together as *prediction-based theories* (Section 4.2), whereas the latter are commonly referred to as *memory-based theories* (Section 4.3).

4.1.4 Linking hypotheses and resource limitations

Psycholinguistic theories are typically evaluated by relating predictions about processing difficulty to behavioural measurements. A central assumption is that increased cognitive load during sentence processing results in longer reading times, forming the basis for a set of influential *linking hypotheses* connecting mental processes to duration measures obtained in reading experiments (Just and Carpenter, 1980; Just et al., 1982; Haberlandt, 1994). Under closer inspection, this assumption actually consists of three separate assumptions (Haberlandt, 1994):

1. The *processing time hypothesis* states that the time required to process a word reflects the amount of cognitive work performed at that point (Haberlandt, 1994).
2. The *eye-mind assumption* proposes that the duration of a reader's fixation on a word closely corresponds to the time needed to process it (Reichle, 2021, Chapter 6).
3. The *immediacy hypothesis* suggests that readers attempt to interpret each word as soon as it is encountered, integrating it into the developing sentence representation without delay (Reichle, 2021, Chapter 6). This is the notion of incrementality dicussed in the preceding section.

The fact that word frequency and length have an effect on the reading time of a word in a sentence and that spillover effects are usually weaker than the immediate effect are taken as support for the general validity of these assumptions (Just et al., 1982). Together, they motivate the widespread use of reading time measures such as self-paced reading times and eye-tracking fixation durations as indicators of processing difficulty. Section 3 provides a more detailed discussion of these types of data.

In addition to the assumptions outlined above, many theories explain processing difficulty in terms of limited cognitive resources. According to this perspective, some syntactic structures require more memory or computation than others, and these demands lead to observable slowdowns during comprehension (Just and Carpenter, 1992; Gibson, 2000). In particular, resource-based accounts typically assume that

1. different syntactic configurations require different amounts of cognitive resources,
2. the human sentence processor has limited resources, and
3. increased resource demands lead to increased processing difficulty.

4.1.4.1 Reading time notation

Different notations exists for formalising linking hypotheses between measures derived from models of human sentence processing and reading times. This section shall establish the conventions used in this work. The central object is a theoretical reading time function defined over arbitrary symbol sequences. Empirical observations in a corpus are interpreted as noisy realisations of this underlying quantity.

Definition 4.1 (Reading time function)**.** Let V be a vocabulary. The *reading time function* is a mapping

$$\mathrm{RT}\colon V^* \times V^* \to \{\, x \in \mathbb{R} \mid x \geq 0 \,\},$$

where $\mathrm{RT}(r, c)$ assigns a non-negative reading time to a *region of interest* $r \in V^*$ with a left context $c \in V^*$.

In the case of word-level processing, I consider $u = w_i$ and $c = w_{<i}$ for some $w \in V^*$.

The theoretical reading time function is assumed to decompose over sequences. This motivates the following structural property.

Definition 4.2 (Additivity)**.** A function $f\colon V^* \times V^* \to \mathbb{R}$ for a vocabulary V is *additive* if for all $r_1, r_2, c \in V^*$,

$$f(r_1 r_2, c) = f(r_1, c) + f(r_2, c r_1).$$

This definition formalises the intuition that the processing cost of a sequence can be decomposed into the costs of its parts, where the context for the second part is extended by the first.

Note that any function $g\colon V \times V^* \to \mathbb{R}$ on single symbols induces an additive function $\tilde{g}$ via

$$\tilde{g}(r, c) \coloneqq \sum_{i=1}^{|r|} g\big(r_i,\, c r_1 \dots r_{i-1}\big). \tag{8}$$

Thus, formalising word-level processing costs is without loss of generality under the additivity assumption. However, note that if V^* is induced by a tokenisation τ_1, a different tokenisation τ_2 requires the definition of a new function g' over the vocabulary it induces. Depending on the underlying model, these might not conform to $g(r, c) = g'(r', c')$ where $\tau_1^{-1}(cr) = \tau_2^{-1}(c'r')$.

Empirical reading times in a corpus are treated as noisy observations of RT. For a token w_i in context $w_{<i}$, one observes

$$RT_i^p = \mathrm{RT}\left(w_i, w_{<i}\right) + b_i^p + \varepsilon_i, \tag{9}$$

where p refers to a participant, ε_i captures measurement noise and unexplained variance and b_i^p is a participant-wise deviation (see Section 6 for the statistical model).

The linking hypotheses described in the following will assume that model-derived quantities are linearly related to reading times up to affine transformation. The symbol $\propto$ will be used, inspired by Levy (2008) to describe this relationship.

Definition 4.3 (Positive affine relation)**.** For parameters $\beta_1, \beta_2 > 0$, an equivalence relation is defined by

$$\propto_{\beta_1, \beta_2} \coloneqq \{\, \langle\, a, b \,\rangle \in \mathbb{R} \times \mathbb{R} \mid a = \beta_1 b + \beta_2 \,\}.$$

Given a candidate RT predictor $f\colon V^* \times V^* \to \mathbb{R}$, a *linking hypothesis* is formulated as

Hypothesis 4.1. There exists a positive affine relation $\propto_{\beta_1,\beta_2}$ such that

$$\mathrm{RT}\left(w_i, w_{<i}\right) \propto_{\beta_1,\beta_2} f\left(w_i, w_{<i}\right) \qquad \text{for all } w_1 \ldots w_i \in V^*.$$

That is, the predictor f is assumed to explain reading times up to a linear rescaling. The existential statement *There exists…* along with the subscript of $\propto$ will be omitted for brevity.

4.1.5 Architectural issues

The following sections discuss two central architectural questions concerning the cognitive mechanisms underlying sentence comprehension: modularity vs. interactivity and serial vs. parallel processing. These dimensions can be used to characterise theories of sentence processing.

4.1.5.1 Modular and interactive models of sentence processing

One of the central debates in psycholinguistics concerns the degree to which sentence comprehension is driven by syntactic structure alone or by multiple sources of information operating simultaneously. Two broad classes of theories have been proposed: *modular* and *interactive* models of sentence processing.

Early accounts assumed a largely modular architecture in which syntactic analysis proceeds independently of semantic or contextual information (Fodor, 1983). In these models, an initial structural analysis of the sentence is constructed using only syntactic cues, after which other sources of information may influence interpretation if necessary (Reichle, 2021, Chapter 4). For this reason, models of this kind are also called *two-stage models* (Reichle, 2021, Chapter 4). In contrast, newer *interactive models* assume that multiple sources of information, including syntax, semantics, lexical statistics and contextual knowledge, jointly influence parsing decisions from the beginning of processing (Reichle, 2021, Chapter 4).

Both approaches agree that sentence processing is incremental but differ in their assumptions about how information sources interact during comprehension.

4.1.5.2 Serial and parallel processing

Another architectural question concerns whether sentence processing proceeds *serially* or *in parallel*. The notion of parallelism, however, can refer to different levels of processing and must therefore be interpreted carefully.

At the level of syntactic interpretation, the classical debate concerns whether the comprehender constructs a single syntactic analysis of the sentence at a time or maintains multiple competing analyses simultaneously. In *serial parsing* models, the processor constructs only one syntactic structure for a given sentence prefix (Reichle, 2021, Chapter 4). This structure is incrementally extended as new words are encountered, and if later input proves incompatible with the current analysis, the parser must revise the structure, resulting in reanalysis and increased processing difficulty. The garden-path model of Frazier (1978) is a prominent example of such a serial architecture.

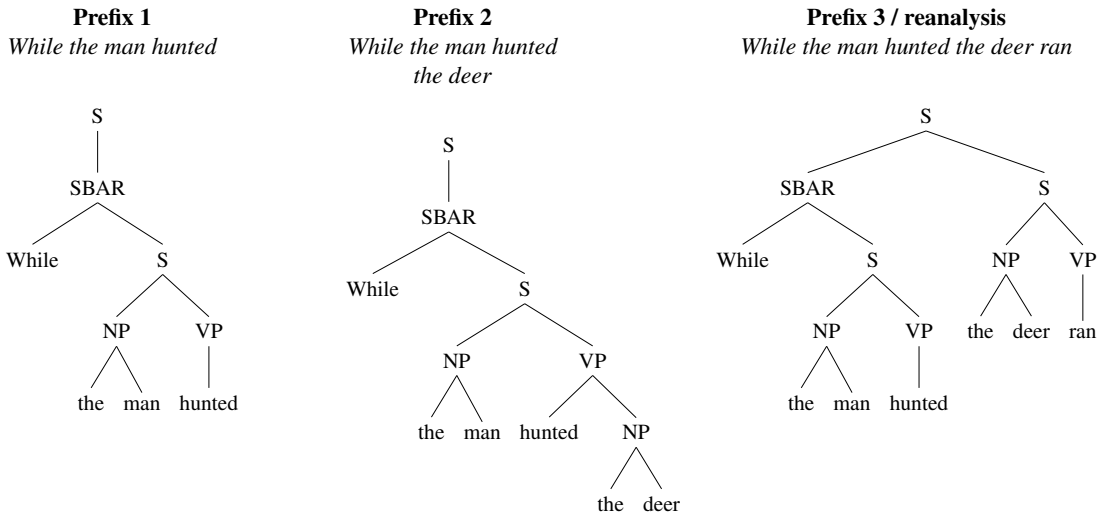


Figure 3: Incremental syntactic analysis of the garden-path sentence *While the man hunted the deer ran into the woods* (Bailey and Ferreira, 2003) in a phrase-structure framework. After the prefix *While the man hunted*, the comprehender has built a subordinate-clause analysis. After *the deer*, a preferred initial parse attaches *the deer* as the object of *hunted*. Encountering *ran* forces a reanalysis in which *the deer* is interpreted as the subject of the main clause.

In contrast, *parallel* models assume that the comprehender may entertain several alternative interpretations of a sentence prefix simultaneously (Reichle, 2021, Chapter 4). Different candidate analyses are supported by various probabilistic constraints, such as lexical preferences, semantic plausibility and contextual information (Levy, 2008). Under this view, ambiguity results in increased competition between analyses and its resolution can happen gradually as the sentence unfolds while serial models always comit to one posible analysis (Hopf et al., 2003).

The *garden-path model* of Frazier (1978) is a prominent example of a serial and modular parsing architecture. According to this framework, the human parser first constructs a single syntactic structure based only on syntactic information. If subsequent input proves incompatible with that analysis, the structure must be revised, which leads to increased processing difficulty. Garden-path sentences illustrate this phenomenon. Consider Example (2) taken from (Bailey and Ferreira, 2003).

(2) While the man hunted the deer ran into the woods.

Readers initially interpret *the deer* as the object of *hunted*. When encountering the verb *ran*, however, this interpretation becomes incompatible with the remainder of the sentence, forcing the comprehender to revise the syntactic structure. Such reanalysis processes have been associated with observable slowdowns in reading time experiments (Sturt et al., 1999; Grodner et al., 2003). Figure 3 illustrates how the single constituent analysis of a garden-path sentence evolves as the sentence unfolds according to serial models.

On the opposite side of the spectrum lie *constraint-based accounts*, architectures which are interactive and parallel in nature (Reichle, 2021, Chapter 4). Classical examples include MacDonald et al. (1994) and McRae et al. (1998).

4.2 Expectation-based sentence processing theories

Expectation-based theories of sentence processing assume that readers continuously form probabilistic predictions about upcoming linguistic input (Ryu and Lewis, 2025). As each new word is encountered, these predictions are updated in light of the observations. Processing difficulty arises when the incoming word leads to a substantial update of these expectations (Levy, 2008).

Two closely related measures have been proposed to formalise this idea. The first is *surprisal*, which links processing difficulty to the conditional probability of the observed word given its preceding context (Hale, 2001; Levy, 2008). The second is the *entropy reduction hypothesis*, which attributes processing cost to decreases in uncertainty over possible syntactic analyses as the sentence unfolds (Hale, 2003; Hale, 2006). Both measures derive from information-theoretic principles and are typically computed using probabilistic models of language.

In the following two sections, I will discuss surprisal theory in more detail before briefly outlining the entropy reduction hypothesis.

4.2.1 Surprisal theory

Surprisal theory proposes that the processing difficulty of a word is determined by how unexpected it is is given its preceding linguistic context (Hale, 2001; Levy, 2008). Formally, this idea is grounded in information theory, where the information content of an event is defined as the negative logarithm of its probability (Shannon, 1948, see Section 2.3).

4.2.1.1 Theoretical motivation

Early psycholinguistic research investigated the effect of the predictability of words in a context on reading times with so-called *cloze tests*. Often, they focused on a single critical region in a relatively constraining context like a sentence ending. Participants were shown an incomplete sentence and were asked to provide the most likely continuation (Taylor, 1953; Luke and Christianson, 2018). Aggregating and normalising these responses yields *cloze probabilities*, i.e. empirical estimates of how likely readers judge a continuation to be in context (Luke and Christianson, 2018). Recent *serial cloze* procedures provide more natural continuation judgments word by word across complete sentences (Luke and Christianson, 2018).

Using such cloze probabilities, researchers established that well predictable words tend to be processed more easily than unlikely words considering both reading-time and electrophysiological measures (Ehrlich and Rayner, 1981; Rayner et al., 2004; Kutas and Hillyard, 1984). At a first glance, this invites the idea that processing difficulty decreases as the contextual probability of the current word increases:

Hypothesis 4.2. Given random variables W_i over the events that some word appears at position i in a word sequence, the following relationship holds:

$$\mathrm{RT}(w_i, w_{<i}) \propto -P(\, W_i = w_i \,|\, W_1 = w_1, \ldots, W_{i-1} = w_{i-1} \,) \qquad \forall\, w_1 \ldots w_i \in V^*.$$

However, a linear mapping from difficulty to raw probability is theoretically unsatisfactory. While cloze tasks usually determine probability on a single-word basis, it would be natural for this relationship to extend to multi-word expressions and whole phrases with their probability also being

linearly related to reading times. However, probability itself is not additive across concatenated regions, so a cost function based on a linear transformation of probability, does not extend naturally from single words to larger strings. Formally,

$$\begin{aligned} & \mathrm{RT}(w_i, w_{<i}) \propto -P(\, w_i \,|\, w_{<i}\,) \\ \not\Rightarrow \quad & \underbrace{\mathrm{RT}(w_i w_{i+1}, w_{<i})}_{\mathrm{RT}(\, w_i \,|\, w_{<i}\,)+\mathrm{RT}(\, w_{i+1} \,|\, w_{\leq i}\,)} \propto \underbrace{-P(\, w_i w_{i+1} \,|\, w_{<i}\,)}_{-P(\, w_i \,|\, w_{<i}\,)\cdot P(\, w_{i+1} \,|\, w_{\leq i}\,)}. \end{aligned} \tag{10}$$

This holds since generally $-P(\, w_i \,|\, w_{<i}\,) \cdot P(\, w_{i+1} \,|\, w_{\leq i}\,) \neq -P(\, w_i \,|\, w_{<i}\,) - P(\, w_{i+1} \,|\, w_{\leq i}\,)$.

Additionally, following Levy (2008), measurements with respect to differential probabilities (e.g. Rayner et al., 1996) provide evidence against a linear relationship between conditional word probability and reading times, as they suggest that absolute predictability differences between two possible continuations manifest more strongly for lower probabilities than for high probabilities. Based on these observations, Levy (2008) hypothesise that the difference in difficulty between two items is a function of the ratio of their probabilities and not of their difference.

Hypothesis 4.3. There exists some function f, so that for all contexts $w_{<i} \in V^*$ and all pairs of alternative continuations $w_i^a \ldots w_{i+k}^a \in V^*$ and $w_i^b \ldots w_{i+k}^b \in V^*$ the following relationship holds:

$$\mathrm{RT}(w_i^a \ldots w_{i+k}^a, w_{<i}) - \mathrm{RT}(w_i^b \ldots w_{i+k}^b, w_{<i}) \propto f\left(\frac{P(\, w_i^a \ldots w_{i+1}^a \,|\, w_1 \ldots w_{i-1}\,)}{P(\, w_i^b \ldots w_{i+1}^b \,|\, w_1 \ldots w_{i-1}\,)}\right)$$

A related approach is taken by Hale (2001) who interpret sentence processing as the incremental cognitive elimination of hypothetical continuations. At any word i, the measure $1 - P(w_{\leq i})$ gives the probability of the possible word sequences over the vocabulary that were not confirmed up to i. Hale (2001) call this the *work* which the processor has performed for that sentence prefix.

Hypothesis 4.4.

$$\mathrm{RT}(w_{\leq i}) \propto \mathrm{work}(w_{\leq i}) \coloneqq 1 - P(w_{\leq i}) \qquad \forall\, w_1 \ldots w_i \in V^* \tag{11}$$

Assuming that processing is incremental,i.e. that new information is always used when it is encountered (Section 4.1.3) before continuing on to the next input symbols, the incremental work performed at position i can be defined as

$$\begin{aligned} \mathrm{work}(w_i, w_{<i}) \coloneqq \mathrm{work}(w_{\leq i}) - \mathrm{work}(w_{<i}) &\overset{\text{(Eq. 11)}}{=} (1 - P(w_{\leq i})) - (1 - P(w_{<i})) \\ &= P(w_{<i}) - P(w_{\leq i}). \end{aligned} \tag{12}$$

Using the chain rule of probability (Section 2.2) one can replace

$$P(w_{\leq i}) = P(w_{<i}) P(\, w_i \,|\, w_{<i}\,), \tag{13}$$

in Equation 12, which yields

$$\mathrm{work}(w_i, w_{<i}) = P(w_{<i})\,(1 - P(\, w_i \,|\, w_{<i}\,))\,. \tag{14}$$

Thus, the incremental work corresponds to the overall probability mass of continuations compatible with the prefix $w_{<i}$ that become impossible when the word w_i is observed. Because this

quantity depends on the absolute probability of the prefix itself, it decreases exponentially with sentence length. However, a decrease of reading times of this order does not appear to be supported by experimental data (Section 3.2.6).

All in all, raw probability does not extend naturally from single words to larger regions, it contradicts observations with respect to differential predictability and it implies an empirically unsupported exponential decrease of reading times throughout the sentence, suggesting that it is not a linear predictor of processing difficulty.

4.2.1.2 Linking hypothesis

The issues outlined above and information theoretical considerations motivated Hale (2001) to propose transforming word predictability to a log-scale, yielding a linking hypothesis between negative log probability and reading times:

Hypothesis 4.5 (Surprisal linking hypothesis)**.**

$$\mathrm{RT}(w_i, w_{<i}) \propto -\log P(\, w_i \,|\, w_{<i}\,) \qquad\qquad \forall\, w_1 \ldots w_i \in V^*.$$

Note that the specification of a logarithm base is not relevant for this relationship since it amounts to a fixed difference in coefficients: $\log_a x = \frac{1}{\log_b a} \log_b x$ for any two bases a, b. Furthermore, Levy (2008) actually define the probability measure on an extended domain $P(\, w_i \,|\, w_{<i}, C\,)$ with C representing the extra-sentential context. However, C is commonly omitted in subsequent works.

The influence of word frequency on reading times, which is well-established in the field (Section 3.2.6), is generally taken to manifest logarithmically (e.g. Just et al., 1982). Thus, word frequency can be seen as a special case of the relationship proposed in Hypothesis 4.5 with a context size of zero (Levy, 2008). The right-hand-side in Hypothesis 4.5 is equal to conditional surprisal over running text (cf. Definition 2.23). As discussed in Section 2.3, surprisal measures the information conveyed by an observation under a given probability distribution. Applied to running text, it provides a word-by-word measure of contextual unexpectedness. If surprisal is low, word i is predictable given the context. If it is high, predictability is low.

A major advantage of surprisal over raw probability is that it composes additively over successive words. By the chain rule for surprisal established in Corollary 2.3, for every $k \in \mathbb{N}$,

$$\mathrm{RT}(w_i \ldots w_{i+k}, w_{<i}) = \sum_{j=i}^{i+k} \mathrm{RT}(w_j, w_{<j}) \overset{\text{(H. 4.5)}}{\propto} \sum_{j=i}^{i+k} \mathrm{I}(\, w_j \,|\, w_{<j}\,) \overset{\text{(Cor. 2.3)}}{=} \mathrm{I}(\, w_i \ldots w_{i+k} \,|\, w_{<i}\,) \tag{15}$$

This logarithmic additivity makes it a natural candidate for relating word-by-word probabilistic expectations to region-based reading times From Equation 15 it also follows that surprisal is invariant with respect to any tokenisation. This allows to compute word-level surprisal as used by the linking hypothesis above from subword probabilities in tokenised language models.

Furthermore, logarithmic probabilities explain Rayner and Well (1996)'s data with respect to differential cloze probabilities discussed in the preceding section (see Hypothesis 4.3), as the difference between the surprisal values between two alternative continuations reduce to the logarithm

of the ratio of their probabilities:

$$\begin{aligned}\mathrm{RT}(w_i^a \ldots w_{i+k}^a, w_{<i}) - \mathrm{RT}(w_i^b \ldots w_{i+k}^b, w_{<i}) \\ \overset{\text{(H. 4.5)}}{\propto} \mathrm{I}(\, w_i^a \ldots w_{i+k}^a \mid w_{<i}\,) - \mathrm{I}(\, w_i^b \ldots w_{i+k}^b \mid w_{<i}\,) \\ = -\log \frac{P(\, w_i^a \ldots w_{i+k}^a \mid w_{<i}\,)}{P(\, w_i^b \ldots w_{i+k}^b \mid w_{<i}\,)}. \end{aligned} \tag{16}$$

4.2.1.3 Interpretations

Although surprisal can be computed directly from conditional next-word probabilities, different authors motivate it through richer interpretations of incremental sentence processing.

Disconfirmed continuations Let's revisit Hale (2001)'s approach of interpreting sentence processing as the incremental elimination of possible utterances outlined in Section 4.2.1.1. The work that a sentence processor needs to perform for a given sentence prefix was defined as the probability of the word sequences that could be ruled out from appearing, $\mathrm{work}(w_{\leq i}) = 1 - P(w_{\leq i})$. Later, I disregarded this hypothesis, concluding that raw probability was not suitable as a reading time predictor.

However, when log-transforming the prefix probabilities, one gets a closely related interpretation. Define the cumulative work for processing a prefix as

$$\mathrm{work}_{\leq i} \coloneqq -\log P(w_{\leq i}) \tag{17}$$

Now, revisiting Equation 12 with this new definition of work, the incremental cost at position i becomes per-word surprisal:

$$\begin{aligned}\mathrm{work}_i &= \mathrm{work}_{\leq i} - \mathrm{work}_{<i} \\ &= -\log P(w_{\leq i}) + \log P(w_{<i}) \\ &= -\log \frac{P(w_{\leq i})}{P(w_{<i})} \\ &= -\log P(\, w_i \mid w_{<i}\,) \end{aligned} \tag{18}$$

A prefix probability can be simply operationalised as the sum of the probabilities of all derivations that can generate the prefix (Hale, 2001). Hale (2001) show that these prefix probabilities naturally emerge from constituent parsers generating leftmost derivations such as Stolcke (1995)'s probabilistic Earley parser. In the following, motivated by Levy (2008), an abstract notion of a syntactic analyses will be adopted without assuming a specific formalism. The only condition will be that surface strings are part of the analyses.

Let $\mathscr{T}$ be a set of complete hidden structures, such as full syntactic analyses licensed by a probabilistic grammar and let $P(T)$ be the probability assigned to analysis $T \in \mathscr{T}$. For a prefix $w_1 \ldots w_i$, define the set of compatible derivations

$$\mathscr{T}_i \coloneqq \{\, T \in \mathscr{T} \mid T \text{ yields a string whose prefix is } w_1 \ldots w_i \,\}. \tag{19}$$

The *prefix probability* of $w_1 \ldots w_i$ is then obtained by marginalising over all compatible derivations, with

$$P(w_1 \ldots w_i) = \sum_{T \in \mathscr{T}} P(\, w_1 \ldots w_i \,|\, T\,)\, P(T) = \sum_{T \in \mathscr{T}_i} P(T), \tag{20}$$

since $P(\, w_1 \ldots w_i \,|\, T\,)$ is 0 for every T where $w_1 \ldots w_i$ is not a prefix and 1 if it is a prefix. Finally, the conditional probability of a word given its context can be computed using the formula in Equation 18.

In this view, the processing of a new word w_i eliminates all syntactic analyses that are incompatible with w_i. The quantity $P(w_1 \ldots w_{i-1})$ represents the total probability mass assigned to all structures compatible with the prefix seen so far. After observing w_i, only those derivations that generate $w_{\leq i}$ remain possible, with a total probability mass of $P(w_1 \ldots w_i)$. The more probability mass had previously been assigned to such disconfirmed structures, i.e. the larger the difference between $P(w_1 \ldots w_{i-1})$ and $P(w_{\leq i})$, the greater the work for a processor (Hale, 2001).

This relation is relative: if half of the probability mass maintained at step $i-1$ is pruned, the effect is the same irrespective of the absolute probability of the prefix $w_1 \ldots w_{i-1}$. This is compatible with the assumption that the comprehender assigns a fixed amount of computational resources to possible syntactic analyses at any point in time which has to be reassigned when analyses turn out to be incompatible with new input.

Bayesian belief update Levy (2008) gives a related but more general Bayesian interpretation. Suppose that at position i, the comprehender maintains a probability distribution over possible linguistic analyses compatible with the current prefix. Let

$$P_i(T) \coloneqq P(\, T \,|\, w_1 \ldots w_i\,) \tag{21}$$

be the prior distribution over analyses before observing w_{i+1}. When the next word w_{i+1} is observed, this distribution is updated to the posterior P_{i+1}. Levy (2008) propose that the processing cost of this update can be characterised by the penalty incurred from encoding the posterior distribution over syntactic analysis with the prior measured using Kullback-Leibler divergence (Definition 2.29):

$$\mathrm{D}_{\mathrm{KL}}(\, P_{i+1} \,\|\, P_i\,) = \sum_{T \in \mathscr{T}} P_{i+1}(T)\, \log \frac{P_{i+1}(T)}{P_i(T)} = \sum_{T \in \mathscr{T}_{i+1}} P_{i+1}(T)\, \log \frac{P_{i+1}(T)}{P_i(T)}. \tag{22}$$

Corollary 4.1. KL-divergence between P_{i+1} and P_i is equal to lexical surprisal $-\log P(\, w_{i+1} \,|\, w_{\leq i}\,)$ for all prefixes $w_1 \ldots w_i \in V^*$ (Levy, 2008).

Proof. First, note that for any integer i and any $T \in \mathscr{T}_i$:

$$P_i(T) \overset{\text{(Eq. 21)}}{=} P(\, T \,|\, w_1 \ldots w_i\,) \overset{(*)}{=} \frac{P(T, w_1 \ldots w_i)}{P(w_1 \ldots w_i)} = \frac{P(T)}{P(w_1 \ldots w_i)}, \tag{23}$$

where equality $(*)$ holds since any compatible tree $T \in \mathscr{T}_i$ yields $w_1 \ldots w_i$. Now, substitution into the ratio in Equation 22 yields

$$\frac{P_{i+1}(T)}{P_i(T)} = \frac{\left(\frac{P(T)}{P(w_1 \ldots w_{i+1})}\right)}{\left(\frac{P(T)}{P(w_1 \ldots w_i)}\right)} = \frac{P(w_1 \ldots w_i)}{P(w_1 \ldots w_{i+1})} = \frac{1}{P(\, w_{i+1} \,|\, w_{\leq i}\,)} \tag{24}$$

Therefore, the ratio can be factored out, leaving a sum that equals 1.

$$\begin{aligned} \mathrm{D}_{\mathrm{KL}}(\, P_{i+1} \,\|\, P_i \,) &= \sum_{T \in \mathscr{T}_{i+1}} P_{i+1}(T) \log \frac{P_{i+1}(T)}{P_i(T)} \\ &= \log \frac{1}{P(\, w_{i+1} \,|\, w_{\leq i} \,)} \sum_{T \in \mathscr{T}_{i+1}} P_{i+1}(T) \\ &= -\log P(\, w_{i+1} \,|\, w_{\leq i} \,) \end{aligned} \tag{25}$$

□

This shows that surprisal can be interpreted as the information-theoretic cost incurred, or equivalently, as the information gained when updating probabilistic beliefs about sentence structure.

An interesting consequence of Equation 24 is that the relative ranking of the surviving analyses does not change when a new word is observed, as the probability of syntactic analysis T given the extended prefix $w_1 \ldots w_{i+1}$ is equal to the probability of T at the previous step multiplied by the factor $P_k(T)/P(\, w_{j+1} \,|\, w_{\leq j} \,)$. Note that this factor is constant across all surviving parses, thus all remaining parses get scaled up uniformly to again sum to 1, maintaining their ranking position. This contrast the *reranking* (as referred to by Levy (2008)) of syntactic analyses assumed by constraint-satisfaction models such as MacDonald et al. (1994). As shall be seen in Section 4.2.1.5, this fact is guaranteed by the distributions being defined over analyses for complete sentences.

Prefix codes and pre-activation Surprisal can also be interpreted as the amount of information conveyed by a linguistic observation without requiring explicit symbolic parsing. In information theory, it quantifies the code length required to represent the next word w_i in an optimal prefix code adapted for probability distribution $P(\, \cdot \,|\, w_{<i} \,)$ (see Section 2.3). In such codes, high-probability events receive short codes whereas rare events require longer codes.

If identifying an incoming word involves mapping it onto an internal representation that must be processed sequentially and remain uniquely decodable, and if processing cost and time scales with the size or complexity of this representation, then an optimal coding scheme minimises expected processing cost by assigning shorter/less complex representations to more probable continuations. Under these assumptions, contextual word surprisal would be linearly related to processing difficulty.

A related intuition can be formulated in terms of pre-activation. Following Kuperberg and Jaeger (2016), contextual information may partially activate neural representations associated with likely upcoming words, thereby reducing the additional processing required once the word is observed (Kuperberg and Jaeger, 2016). Under resource limitations, such anticipatory activation must be distributed across possible continuations. This suggests that prediction may be understood as a form of resource allocation that facilitates efficient processing of incoming input.

In both interpretations, prediction would be a consequence of optimisation in the face of pressure to reduce the time or energy for processing word observations.

4.2.1.4 Surprisal as a causal bottleneck

The preceding section showed that the emergence of surprisal is not tied to any specific syntactic formalism (or the employment of discrete syntax at all). For this reason, Levy (2008) describe surprisal as a *causal bottleneck*: distinct representational theories may differ in the structures they posit but as long as these structures induce the same probability distribution over continuations, they yield the same surprisal prediction for processing difficulty. Thus, surprisal can be operationalised via *n-gram models*, *probabilistic context-free grammars* (PCFGs) or neural language models without any implementation being inherently more plausible according to the linking hypothesis (Levy, 2008). See Section 5 for more detail on how to gain surprisal using artificial neural networks.

Slaats and Martin (2025) criticise the focus on surprisal as a computational-level linking hypothesis in recent work on sentence processing, arguing that it distracts from the nature of cognitive representations and is not explanatory with respect to comprehension processes. Since the conditional probability $P(\, w_i \,|\, w_{<i} \,)$ is estimated from observed language data, it implicitly reflects factors that influences the distribution of words in texts. These influences may include syntactic structure, lexical frequency, semantic plausibility, discourse context and pragmatic constraints (Slaats and Martin, 2025). Following Slaats and Martin (2025), these underlying variables can be viewed as *latent factors* in that they affect both the probability distribution over upcoming words and human processing behaviour, while not being explicitly represented in the surprisal value itself. Formally, let Z_i denote a set of latent linguistic variables available at word i such as syntactic structure, semantic context or lexical frequency. According to the view of surprisal as a causal bottleneck (Levy, 2008) the corresponding causal graph would look as follows:

$$Z_i \rightarrow \tilde{P}(\, w_i \,|\, w_{<i} \,) \rightarrow \mathrm{RT}(w_i, w_{<i}). \tag{26}$$

$\tilde{P}$ shall denote the cognitive distribution assumed by surprisal theory while P is a researcher's estimation, for instance gained using a transformer language model.

Now, Slaats and Martin (2025) argue that instead of causality, Z_i may jointly influence both the probability of a word and the observed processing difficulty (here $\mathrm{RT}(w_i, w_{<i})$), as written in Equation 27, and that consequently correlations between surprisal and behavioural or neural responses do not uniquely identify the underlying causal mechanism.

$$Z_i \rightarrow \begin{cases} \tilde{P}(w_i \mid w_{<i}) \\ \mathrm{RT}(w_i, w_{<i}) \end{cases} \tag{27}$$

Following this idea, evaluating potential predictors of processing difficulty using models that contain surprisal makes it difficult to determine the identity of the effects that are actually responsible for elevated reading times. This motivates the view that surprisal functions as a statistical summary of the input rather than as a mechanistic explanation of language comprehension (Slaats and Martin, 2025) and contrasts the syntax-based interpretation taken by Hale (2001) and Levy (2008). An alternative view is to attribute surprisal with a proper causative role for a subcomponent of processing. In particular due to its effect being observable in early eye-tracking measures, Ryu and Lewis (2025) suggest surprisal to reflect an effect of probabilistic expectations in lexical identification, motivating testing other predictors as explanans for higher-level comprehension against a surprisal-baseline.

4.2.1.5 On the computational feasibility of predictions

As Levy (2008) themselves note, given the interpretation of surprisal as a measure extracted from a highly parallel syntactic parser making predictions over a large space of possibly input yet to come, the question suggests itself of how feasible and plausible such a form of processing is for a human comprehender. In the following, I will outline two considerations that pertain to this issue.

Partial structures The formalisation of the Bayesian interpretation of surprisal in Section 4.2.1.3 assumed probability distributions over all complete structures, including all surface strings, compatible with the prefix seen so far and required summing over the set of these analyses. However, assuming a small vocabulary size of 2000 and a fixed sentence length of 10, the set of all possible word sequences already contains $2000^{10} = 1.024 \times 10^{33}$ elements, clearly showing that this feat is unfeasible for any computational process, let alone for a human comprehender.

Levy (2008) note that surprisal could be closely approximated if the parser maintained only a limited set of highly probable analyses rather than a full parallel distribution. Then, substantial deviations from complete surprisal would only occur when processing requires shifting attention to analyses that were not previously allocated cognitive resources. However, assuming a distribution over complete structures, longer sentence lengths let probabilities shrink exponentially, letting a fixed beam size of considered analyses represent a fraction of the overall probability mass.Therefore, one can conclude that if humans make concrete predictions regarding the upcoming surface content and/or hidden analysis yielding the surface content, that these predictions span only a limited window into the future.[13] Therefore, a parsing-based formalisation of surprisal theory must be based on partial structures such as eager incremental parses, underspecified syntactic states, or other prefix-based representations. Indeed, this is the mechanism adopted by the parser used by Hale (2001).

As Levy (2008) point out, under a limited number of explored analyses, the equivalence between KL-divergence for consecutive distributions over analyses and lexical surprisal does not hold. In fact, this is also not guaranteed if assuming prefix structures. Levy (2008) mention the necessity of distributions over partial parses but do not provide a formalisation. In the following, I will explore the consequences of explicitly assuming incomplete parses, yielding two key results for the relationship between KL-divergence of parse probabilities and surprisal. Note that it will assume that the abstracted grammar formalism does not include deletion of deducted structure as an operation. This is standard in symbolic parsing paradigms.

First, I will extend the notion of sentence prefixes to *prefix structures* (e.g. prefix trees for constituent parsing). A prefix structure s can correspond to many complete parses. Assume that the parser has processed the context up to position i for some sentence w and maintains a probability distribution over a selection of prefix structures given this context. Let S_i be a random variable ranging over the set $\mathscr{S}_i$ of prefix structures maintained by the parser after processing the prefix $w_{\leq i}$. I require that all elements in $\mathscr{S}_{i+1}$ are derivable from elements in $\mathscr{S}_i$ and that no two distinct items in $\mathscr{S}_i$ derive the same item in $\mathscr{S}_{i+1}$. Define

$$P_i(s) \coloneqq P(S_i = s \mid w_{\leq i}) \qquad \text{for } s \in \mathscr{S}_i^{\mathscr{T}}. \tag{28}$$

After observing the next word w_{i+1}, one can define a posterior distribution over the same prefix

[13] Note that this does not rule out more abstract predictions about upcoming ideas or syntactic components.

structures,

$$P^*_{i+1}(s) \coloneqq P(S_i = s \mid w_{\leq i+1}) = \sum_{\{\, s' \in \mathscr{S}_{i+1} \mid s \overset{*}{\Rightarrow} s' \,\}} P(\, S_{i+1} = s' \mid w_{\leq i+1}\,), \tag{29}$$

where $s \overset{*}{\Rightarrow} s'$ means that s' is derivable from s via parsing operations. Hence, the Kullback-Leibler divergence between posterior and prior over prefix structures is

$$\mathrm{D}_{\mathrm{KL}}(\, P^*_{i+1} \,\|\, P_i\,) = \sum_{s \in \mathscr{S}_i} P^*_{i+1}(s) \log \frac{P^*_{i+1}(s)}{P_i(s)}. \tag{30}$$

Analogous to Equation 24, the ratio equates to

$$\frac{P^*_{i+1}(s)}{P_i(s)} = \frac{\left(\frac{P(S_i = s, w_{\leq i+1})}{P(w_{\leq i+1})}\right)}{\left(\frac{P(S_i = s, w_{\leq i})}{P(w_{\leq i})}\right)} = \frac{P(\, w_{i+1} \mid S_i = s, w_{\leq i}\,)}{P(\, w_{i+1} \mid w_{\leq i}\,)} \tag{31}$$

Substituting Equation 31 into Equation 30 yields

$$\mathrm{D}_{\mathrm{KL}}(\, P^*_{i+1} \,\|\, P_i\,) = \sum_{s \in \mathscr{S}_i} P^*_{i+1}(s) \log \frac{P(\, w_{i+1} \mid S_i = s, w_{\leq i}\,)}{P(\, w_{i+1} \mid w_{\leq i}\,)}. \tag{32}$$

Since the denominator does not depend on s, this can be rewritten as

$$\begin{aligned} \mathrm{D}_{\mathrm{KL}}(\, P^*_{i+1} \,\|\, P_i\,) &= -\log P(\, w_{i+1} \mid w_{\leq i}\,) + \sum_{s \in \mathscr{S}_i^{\mathscr{T}}} P^*_{i+1}(s) \log P(\, w_{i+1} \mid S_i = s, w_{\leq i}\,). \\ &= \mathrm{I}(\, w_{i+1} \mid w_{\leq i}\,) - \mathbb{E}_{s \sim P^*_{i+1}}[\mathrm{I}(\, w_{i+1} \mid S_i = s, w_{\leq i}\,)] \end{aligned} \tag{33}$$

Therefore, it can be concluded that

$$\mathrm{D}_{\mathrm{KL}}(\, P^*_{i+1} \,\|\, P_i\,) \leq -\log P(\, w_{i+1} \mid w_{\leq i}\,), \tag{34}$$

with equality iff $P(\, w_{i+1} \mid S_i = s, w_{\leq i}\,) = 1$ for every $s \in \mathscr{S}_i$ such that $P^*_{i+1}(s) > 0$.

Thus, KL divergence over prefix structures coincides with next-word surprisal only if each posterior-supported prefix structure is already deterministic with respect to the next surface word. This means that it must either contain the word itself or a unique preterminal for it

In a parser, $\mathscr{S}_i^{\mathscr{T}}$ would likely be a subset of prefix structures such that no prefix structure can be derived from the other via valid operations of the parsing paradigm. This constitutes an equivalence relation over complete structures where all complete parses that can be reached from an $s \in \mathscr{S}_i$ comprise an equivalence class. In this context, the intuition of Equation 34 is as follows: the coarser the equivalence classes considered at step i, i.e. the less detailed the prefix structure predictions with respect to the following observation, the smaller the probability mass pruned away by the posterior distribution due to wrong predictions.

The above formalisation also makes it suitable to analyse violations of the principle of incrementality, which would lead to even smaller word-by-word KL-divergence than incremental but unpredictive processing. Non-integration is thought to be adopted by humans where the communicative

settings indicate that a shallow analysis suffices (*good enough processing* Ferreira et al., 2001; Ferreira et al., 2002) or where readers buffer lexical input to analyse it later at the phrase ending, leading to wrap-up effects (Sections 3.2.6 and 3.3.5). However, in cases of difficult syntax and garden path constructions surprisal typically underpredicts reading times (Schijndel and Linzen, 2021). Thus, it seems counterintuitive to assume lower effects based on KL-divergence. This could either indicate that partial structures play no role, that shifts in probabilities over syntactic analyses pose no suitable linking hypothesis or that reading times are influenced by orthogonal factors. The last option is plausible insofar as Equation 31 does not attribute any cost to the refinement of a prefix structure $s \in \mathscr{S}_i$ to a set of derivations in $\mathscr{S}_{i+1}$. Syntactic integration could therefore represent the other cost component besides belief update.

Another result is that reranking of analyses emerges for prefix structure sets that are nondeterministic with respect to the next surface word. More concretely, rearranging Equation 31 shows that the scaling of the probabilities of items surviving the update is not uniform but depends on the probability of the succeeding word given the prefix structure:

$$P^*_{i+1}(s) = P_i(s)\frac{P(\, w_{i+1} \mid S_i = s, w_{\leq i}\,)}{P(\, w_{i+1} \mid w_{\leq i}\,)}. \tag{35}$$

Thus, reranking of syntactic interpretations corresponds to the pruning of those complete structures derivable from prefix structure s that are incompatible with the new word. The smaller the fraction of derivations of a prefix structure s that is compatible with w_{i+1}, the stronger it is ranked down by the update.

Lexical prediction Surprisal theory posits a linking hypothesis between processing difficulty and the conditional probability of the observed word given its context. It does not require that the human processor explicitly compute prior probabilities for upcoming words or syntactic analyses. Surprisal can also be formalised as the compatibility of new word input with the comprehender's working representation of the preceding sentence, capturing ease of integration (Demberg, 2010, Section 6.1.2.1). This could be operationalised via a compatibility function $\psi(w_i, h_{i-1})$ yielding values in $\mathbb{R}$, where w_i is the new word and h_{i-1} is the cognitive state after having read words $w_{<i}$.

However, since surprisal theory is formalised in terms of a distributions over all words in a vocabulary, it is neccessary to derive scores that sum to 1. Thus, to gain surprisal from a compatability score between w_i and h_{i-1}, one has to weigh it against all other compatibility scores. Following the standard formulation used in neural language models, this can for instance be achieved using the $\mathrm{softmax}$ function.[14]

Let $\boldsymbol{\alpha}$ denote the sequence of all word input representations in vocabulary V. Then,

$$\begin{aligned} \mathrm{softmax}(\mathbf{c}) &= \frac{\exp(\mathbf{c})}{\sum_{j=1}^{|V|}\exp(\mathbf{c}_{(j)})}, \\ \text{where} \qquad \mathbf{c} &= \psi({}_2\boldsymbol{\alpha}, h_{i-1}), \end{aligned} \tag{36}$$

yields a probability simplex where each index is associated with an item in vocabulary V. When computing surprisal using this distribution, it reduces down to an observation-specific component

[14] If parameterising word probabilities through a probabilistic context-free grammar, production probabilities specified in that grammar do provide scores for single words. In this case, the problem reduces down to collecting them across all available parse options. The question lies in whether the grammar adequately incorporates sufficient contextual information of various kinds (including lexical, semantic) in its specification.

and a normalisation constant based on the compatibilities with all items in the vocabulary,[15]

$$
\begin{aligned}
-\log P(\, w_i \mid w_{<i}\,) &= -\log \left(\frac{\exp(\psi(w_i, h_{i-1}))}{\sum_{w \in V} \exp(\psi(w, h_{i-1}))} \right) \\
&= -\psi(w_i, h_{i-1}) + \log \sum_{w \in V} \exp(\psi(w, h_{i-1})). \quad (37)
\end{aligned}
$$

Let's define a function for the second term as $\mathrm{z}(h) \coloneqq \log \sum_{w \in V} \exp(\psi(w, h))$. It does not depend on the observed word, only on state h_{i-1}. z reflects the competition between continuation options: it becomes large when many continuations are plausible. This means that surprisal depends on: (1) how well the observed word fits into the interpretation and (2) how many items would fit well into the interpretation overall.

It is computationally demanding to maintain compatibility scores for all possible words in the vocabulary (but arguably feasible if interpreting neural network language models in this fashion, see Section 5.3.2.11). Nation (2006) report that a vocabulary with a size of around 8000–9000 words is needed for comprehension of English novels or newspaper articles, thus leading to at least 8000 comparisons at each processing step. This potential inefficiency could be addressed using heuristic prediction. The processor might track compatibility only with respect to a small set of promising words dominating the sum $z(h_{i-1})$ Depending on the number of elements, such a measure could approximate surprisal closely.

The above decomposition also demonstrates the potential merits of prediction: it is not necessary to wait with computing $\phi(w_i, h_{i-1})$ until after the lexical identification of w_i. If adequate resources are available, this can be performed in parallel at step $i-1$ to speed up overall processing. Furthermore, since determining the compatibility between h_{i-1} and some word $w \in V$ could require the activation of its cognitive representation, this could speed up its subsequent lexical identification, if it is actually observed (Kuperberg and Jaeger, 2016; Ryskin and Nieuwland, 2023). Lastly, assuming a reduced set of promising continuations (however these could be determined) has two consequences: (1) If the next word is confirmed to be an item from the promising set and the selection of items is too small or not appropriate for capturing all relevant alternatives, the approximated measure is lower than regular surprisal. (2) If an item that is not in this set is observed, additional processing would be necessary to determine its compatibility.

4.2.1.6 Empirical support

A substantial body of empirical work shows that surprisal derived from probabilistic language models predicts human reading times in both self-paced reading and eye-tracking experiments (Levy, 2008; Demberg and Keller, 2008; Roark et al., 2009; Smith and Levy, 2013; Wilcox et al., 2023). Across these studies, words that are less probable in their preceding context are associated with longer reading times, consistent with the prediction of surprisal theory that processing difficulty increases with the information conveyed by a word.

Early demonstrations of this relationship used surprisal estimates derived from incremental probabilistic parsers, for instance Hale (2001) and Levy (2008) who evaluated the predictions qualitatively and using RTs for dedicated experimental regions. Both Roark et al. (2009) and Demberg

[15] Note that regular normalisation by dividing through the sum of compatibility scores as opposed to taking exponential functions through $\mathrm{softmax}$ would also result in a separation between observation-specific and vocabulary-specific components.

(2010) conduct regression analyses for surprisal on natural reading time corpora, finding significant results for a variant reduced to syntactic unexpectedness.

A particularly influential result concerns the functional relationship between surprisal and reading time. Using large self-paced reading corpora, using an n-gram model, Smith and Levy (2013) showed that reading times increase approximately linearly with surprisal as stated in Hypothesis 4.5. More recent work has confirmed this finding using neural language models, including recurrent neural networks and transformer-based architectures (Wilcox et al., 2020; Shain et al., 2022; Wilcox et al., 2023; Rathi, 2021). Surprisal estimates derived from these models consistently predict self-paced reading times and eye-tracking measures.

Another important observation in this line of work is that the predictive power of surprisal tends to increase with language-model quality. However, recent evidence shows that this relationship is non-monotonic, with large and performant transformer language models yielding surprisal that predicts reading times worse than their smaller counterparts (Oh and Schuler, 2023b; Liu et al., 2024). This issue – either architectural or conceptual – will be discussed in more detail in Section 5.4.1.

4.2.1.7 Relation to other theories

The surprisal framework has consequences for long-standing debates about the architecture of the human sentence processing system introduced in Section 4.1.5. In particular, it has implications for the debate on modular vs interactive processing and for the question of whether syntactic analysis proceeds serially or in parallel.

First, surprisal-based models align naturally with the interactive perspective. The conditional probability $P(w_i \mid w_{<i})$ implicitly integrates all available cues that influence the likelihood of the next word – not only syntactic information – and processing difficulty is predicted to reflect the combined effect of these constraints. Secondly, surprisal naturally corresponds to parallel models of sentence processing in which multiple hypotheses about evolving sentence structure are maintained simultaneously – both in the standard form and when approximating it with a limited set of high-probability items as Levy (2008) suggests. This follows from the definition of surprisal which requires marginalisation over all possible next words or parses of the sentence prefix (see Sections 4.2.1.3 and 4.2.1.5).

Section 4.3.1.5 compares the surprisal theory with the Dependency Locality Theory while Section 4.3.2.4 discusses key similarities and differences in the predictions of surprisal and similarity-based interference.

4.2.1.8 Limitations and extensions

Despite its empirical success, surprisal does not explain all aspects of online sentence processing. Orthogonal to the issue of large transformer models' surprisal decreasing in psychometric predictive power (Section 4.2.1.6) are observations which show systematic discrepancies between surprisal-based predictions and observed reading times. In particular, surprisal-based accounts can underpredict difficulty in constructions involving temporary ambiguity, reanalysis, or strong memory demands (Schijndel and Linzen, 2021; Wilcox et al., 2021; Huang et al., 2024). Following Wilcox (2025) and Arehalli et al. (2023), this leaves open at least two possibilities.

First, human sentence processing may rely on probability distributions that are structured differently from those induced by current language models. For instance, they might assign significantly lower probability to garden-path constructions. In this case, the issue lies not with surprisal as such, but with the model from which it is estimated. As transformers are highly performative universal function approximators (Yun et al., 2019), maybe they lack biases present in the human language processing device. This contrasts Levy (2008)'s assumption that the models representations underlying next-token predictions are irrelevant. This would be the case if we could actually be able to directly approximate the hypothesised cognitive lexical distribution. However, any neural model approximation trained on large corpora exhibits a non-zero loss with a distribution dependent on the model's biases. These deviations influence the predictiveness of of the resulting surprisal measure.

Second, processing difficulty may not be explained sufficiently by surprisal in its current formalisation and/or by surprisal alone. Additional costs may arise from memory retrieval, structural revision, or the allocation of bounded processing resources.

A prominent approach of the latter kind is an extension of classical surprisal theory called *lossy-context surprisal* (Futrell et al., 2020). In this framework, the comprehender no longer has perfect access to the entire left context. Instead, predictions are conditioned on a noisy memory representation $M(w_{<i})$ of the preceding input. Processing difficulty is then defined as the expected surprisal of the next word under that degraded context representation (Futrell et al., 2020):

Hypothesis 4.6 (Lossy-context surprisal linking hypothesis)**.** There exists a probabilistic memory representation function M on V^*, such that

$$\mathbb{E}_{c \sim M(w_{<i})}[\mathrm{I}(\, w_i \mid c\,)] \qquad \forall\, w_1 \ldots w_i \in V^*.$$

Lossy-context surprisal derives locality and forgetting effects from prediction under this imperfect memory and thus forms an important bridge between expectation-based and memory-based theories. Note that its formulation is recursive since $M(w_{<i})$ is computed from $M(w_{<i-1})$ and w_i (Futrell et al., 2020).

Relatedly, Kuribayashi et al. (2022) show that reducing the amount of context available to neural language models can improve their fit to human reading behaviour. This is compatible with the idea that unrestricted access to long contexts, although beneficial for next-word prediction in document analysis or coding, may not be cognitively plausible as a model of human sentence comprehension.

Lastly, it remains unclear how surprisal relates to preview effects in eye-tracking data (Section 3.2.6). If the reader acquires partial information $\tilde{w}_i$ about the word w_i at position $i-1$, then their internal hypothesis about that word could be assumed to take this information into account in the form of $-\log P(\, w_i \mid w_{<i}, \tilde{w}_i\,)$. This would also be the case if word predictability was derived from explicit parses as partial information could rule out those structural interpretations that do not contain the observed word prefix or fragment. On the other hand, if surprisal already manifests at a low level such as lexical identification, one could expect slow-down due to surprising prefixes in the preview. On the other hand, as the effect of surprisal appears to be stronger for eye-tracking than for self-paced reading times despite eye-tracking allowing for preview (Mielczarek et al., 2025), it might play only a limited role. A character-based language model or a marginalisation technique such as the one proposed by Giulianelli et al. (2024) could be a suitable option to investigate this issue further.

4.2.2 Entropy Reduction Hypothesis

This section will give a short overview over the other major expectation-based sentence processing theory, the *entropy reduction hypothesis* (ERH) (Hale, 2003; Hale, 2006). It proposes that processing difficulty during comprehension is determined by the reduction in uncertainty about the parse of the sentence when a new word is encountered. Given prefix $w_1 \ldots w_i$, this uncertainty is quantified by entropy (cf. Definition 2.24):

$$\mathrm{H}(P_i) = -\sum_{T \in \mathscr{T}} P_i(T) \log_b P_i(T). \tag{38}$$

When the next word w_{i+1} is encountered, the distribution over possible parses changes from P_i to P_{i+1}. The entropy reduction at position $i+1$ is defined as the decrease in this uncertainty after observing the next word (Hale, 2006):

$$\mathrm{ER}(w_i, w_{<i}) \coloneqq \max\{\mathrm{H}(P_{i-1}) - \mathrm{H}(P_i), 0\}\,. \tag{39}$$

It is assumed that the processing cost for a word is proportional to the reduction in syntactic entropy. Note that no processing difficulty is assumed if uncertainty is increased.

Hypothesis 4.7 (Entropy reduction linking hypothesis)**.**

$$\mathrm{RT}(w_i, w_{<i}) \propto \mathrm{ER}(w_i, w_{<i}) \qquad \forall\, w_1 \ldots w_i \in V^*.$$

Unlike surprisal, which can be defined directly in terms of the conditional probability of the next word, entropy reduction is inherently defined over a distribution of hidden structural representations (Levy, 2008).

Corollary 4.2. Entropy reduction is not, in general, uniquely determined by next-word probabilities alone.

Proof. It suffices to present two different distributions over parses that induce the same next-word probability but deviating entropy reductions.

Let $\mathscr{T} = \{\, T_1, T_2, T_3 \,\}$ and suppose

$$P(\, x \,|\, T_1 \,) = 1, \qquad P(\, x \,|\, T_2 \,) = 1, \qquad P(\, y \,|\, T_3 \,) = 1. \tag{40}$$

for words x, y. Consider the two parse distributions P^a_{i-1} and P^b_{i-1} before seeing the word specified by

$$P^a_{i-1}(T_1) = \tfrac{1}{2}, \quad P^a_{i-1}(T_2) = 0, \quad P^a_{i-1}(T_3) = \tfrac{1}{2}, \tag{41}$$

and

$$P^b_{i-1}(T_1) = \tfrac{1}{4}, \quad P^b_{i-1}(T_2) = \tfrac{1}{4}, \quad P^b_{i-1}(T_3) = \tfrac{1}{2}. \tag{42}$$

The next-word probability is obtained by marginalising over parses:

$$P^a(\, x \,|\, w_{<i} \,) = \sum_{T \in \mathscr{T}} P(\, x \,|\, T \,) P^a_{i-1}(T) = \tfrac{1}{2} = \sum_{T \in \mathscr{T}} P(\, x \,|\, T \,) P^b_{i-1}(T) = P^b(\, W_i = x \,|\, w_{<i} \,), \tag{43}$$

Then both models assign the same next-word probability to x. After observing x, the posterior over parses is given by Bayes' rule:

$$P_i(T) = \frac{P(\, x \,|\, T\,) P_{i-1}(T)}{\sum_{T'} P(\, x \,|\, T'\,) P_{i-1}(T')}. \tag{44}$$

This yields

$$P_i^a(T_1) = 1, \tag{45}$$

and

$$P_i^b(T_1) = \tfrac{1}{2}, \qquad P_i^b(T_2) = \tfrac{1}{2}. \tag{46}$$

Using Definition 2.24, one obtains

$$\mathrm{H}(P_{i-1}^a) = 1, \quad \mathrm{H}(P_i^a) = 0, \tag{47}$$

and

$$\mathrm{H}(P_{i-1}^b) = \tfrac{3}{2}, \quad \mathrm{H}(P_i^b) = 1. \tag{48}$$

Thus,

$$\mathrm{H}(P_{i-1}^a) - \mathrm{H}(P_i^a) = 1, \qquad \mathrm{H}(P_{i-1}^b) - \mathrm{H}(P_i^b) = \tfrac{1}{2}. \tag{49}$$

Therefore, entropy reduction differs although the next-word probability is identical. □

Empirical work on the predictability of the ERH for processing difficulty is limited precisely due to its reliance on a concrete specified grammar. Hale (2006) successfully relate entropy reduction to repetition accuracy and Wu et al. (2010) find an effect of entropy reduction over prefix structures on reading times. Linzen and Jaeger (2014) observe that in contrast to surprisal, entropy reduction does not predict slow-downs at garden-path disambiguating sites since due to increasing entropy (i.e. $\mathrm{ER} = 0$) in such cases. Generally, the field still lacks evaluations of entropy reduction on naturalistic reading time corpora.

Other researchers have used an alternative notion of *lexical entropy reduction* over successive next-word probability distributions (e.g. Frank, 2013; Song et al., 2024). In contrast to ER on syntactic distributions which can be interpreted to quantify the work performed to reduce the uncertainty about a single matter (Hale, 2003), this method compares entropies for two different kinds of events – the occurrence of a word at position i and at position $i + 1$. Frank (2013) note that measure should mainly be understood as a helful simplification abstracting away from the unknown nature of cognitive representations.

For this alternative formulation, Frank (2013) observe positive correlation with RT and Song et al. (2024) find that it is an effective predictor of neural responses, significantly complementing surprisal. This suggests that language processing is influenced by additional information-theoretic factors beyond negative word probability.

4.3 Memory-based sentence processing theories

Memory-based theories of sentence processing attribute slow-down in reading to limitations in encoding and retrieving representations during incremental comprehension. These theories vary in their assumed mechanisms and therefore make different predictions about reading patterns. Two influential approaches are particularly prominent.

First, *Dependency Locality Theory* (DLT) proposes that integrating new material into an evolving syntactic structure incurs higher *integration costs* when the relevant dependencies are long (Gibson, 2000). Second, *cue-based retrieval* theories (Van Dyke and Lewis, 2003) assume that incoming material provides cues that probe memory for previously processed items (words, constituents, referents). When multiple candidates match the cues, retrieval suffers from *similarity-based interference*, yielding increased processing time (Gordon et al., 2002; Gordon et al., 2006).

In the following, I will explain these two theories in detail, report on empirical evaluations and explore their relations, as well as their connections to the expectation-based theories outlined in Section 4.2.

4.3.1 Dependency Locality Theory

Dependency Locality Theory (DLT) (Gibson, 1998; Gibson, 2000) presupposes that humans incrementally construct syntactic relationships between newly encountered words and already seen context while reading texts and listening to speech, similarly to the garden-path model discussed in Section 4.1.5. It assumes that this process of *integration* demands cognitive resources, with its central prediction being that syntactic relationships with larger distances are more difficult to process than relationships with small distances. This is motivated by observations of increased reading times at the closing sites of constructions such as center embeddings, known as *locality effects* (Gibson, 2000).

The novel contribution brought forward by the DLT concerns limitations of working memory for performing three types of operations necessary for incremental parsing: (1) the introduction of new referents, (2) the integration of referents into the existing analysis of the sentence prefix and (3) maintaining unconnected structures (Reichle, 2021, Chapter 4). Gibson (2000) assume that these actions consumes resorces measured in abstract *energy units* (EUs) and *memory units* (MUs). The linking hypothesis relates these units to reading times.

Gibson (2000) use Webber (1988)'s definition of referents being entities that can be referred to via anaphora after their introduction into the discourse. They are identified by being the heads of noun or verb phrases that have introduced new discourse elements (Gibson, 2000). A central assumption of DLT is that words which do not refer to discourse referents inflict substantially lower processing costs (Gibson, 2000).

4.3.1.1 Syntactic framework

The underlying structural representation used by DLT was originally defined to be phrase-structure grammar with nodes being assigned lexical heads (Gibson, 1998; Gibson, 2000). Since the actions of the framework are defined in relation to these heads and not dependent on specifications of internal nodes, it is equivalently applicable to dependency structures (Demberg and Keller, 2008). Later implementations therefore use dependency annotations for the extraction of DLT-style costs (Demberg and Keller, 2008; Rathi, 2021; Mielczarek et al., 2025).[16]

[16] A constituent tree could be converted without loss of the necessary head structure following Johansson and Nugues (2007).

4.3.1.2 Actions

Introduction and integration Gibson (2000) assume that introducing a new referent requires one energy unit. The second assumption is that integration raises a cost based on the distance between the new referent to integrate and the site of integration in the maintained structure. This distance is measured by the number of integration actions that have happened in between those two items (Gibson, 2000). The assumption is that integration operations track the degradation of activations of memory items (see Section 4.1.2). As reactivation requres resources, items with a large distance are more difficult to access (Reichle, 2021, Chapter 4).

Usually, this is simplified by counting all intermediate verbs and nouns. The resulting number then makes up the EUs consumed for integration at the respective memory site (Demberg and Keller, 2008). Since multiple integrations with unconnected items ifrom the memory can take place for any word w_i, the respective distances are computed separately and then summed to represent the composite *integration cost* for w_i. An additional simplification concerns the fact that only words that introduce new discourse items are attribtued with integration costs, based on the assumption that costs for other words are very small (Gibson, 2000).

Let D_i be the set of dependency-arcs completed at position i. For $\delta \in D_i$, let $d(\delta)$ denote the number of intervening *discourse referents* between the two linked positions that were introduced in this sentence DLT integration cost can be written as

$$\text{integration}_i := \begin{cases} 1 + \sum_{\delta \in D_i} d(\delta) & w_i \text{ introduces a new discourse referent,} \\ 0 & \text{else.} \end{cases} \tag{50}$$

Storage cost Lastly, Dependency Locality Theory assumes that resources are necessary to maintain incomplete structures. For each processing step, the minimal number of syntactic heads still necessary to complete is defined as the *storage costs* at that point (Gibson, 2000). Since a sentence with minimal length consists of at least a noun and a verb, each sentence starts with a storage cost of 2 (Reichle, 2021, Chapter 4). In every subsequent step, storage costs are updated to reflect newly established expectations about the number of dependencies still to close, and reduced if those referents are found (Reichle, 2021, Chapter 4).

Storage costs are measured in *memory units* (MUs) (Gibson, 2000) with each open syntactic dependency contributing one MU. Let storage_i denote the number of open syntactic dependencies at position i.

4.3.1.3 Linking hypothesis

Combining both components yields the following simplified linking hypothesis:

Hypothesis 4.8 (Simplified DLT hypothesis)**.** There exist $\alpha, \beta > 0$ such that

$$\text{RT}(\, w_i \,|\, w_{<i} \,) \propto \alpha \cdot \text{storage}_i + \beta \cdot \text{integration}_i \qquad \forall\, w_1 \ldots w_i \in V^*.$$

As Gibson (2000) do not specify the relationship between EUs and MUs, the rescaling variables α and β are included in the hypothesis.

In practice, many empirical studies focus primarily on integration cost because storage cost is harder to operationalise and thought to play only a minor role in natural texts (Demberg and Keller, 2008). It will also be the focus for the remainder of this thesis.

4.3.1.4 Empirical support

DLT makes several predictions with respect to certain grammatical constructions that have been investigated through qualitative and quantitative measures. First, as object relative clauses require integrating across more intervening referents than subject relatives, it predicts greater difficulty at the former.

See Example (3), taken from Gibson (2000). In (3a), integration cost would be low with the exception of the main verb *hoped* where a relation to the subject *the reporter* needs to be established, crossing three intervening referents (*photographer*, *sent* and *editor*).

In (3b), DLT predicts reading times to be fast in the first five words but slow down on the verb *sent* where the integration into the modified noun structure *the reporter* is increased in distance by the relative clause subject *the photographer*. After that, reading times would decrease again, rise at the main verb *hoped* and finally speed up for the third time.

(3) a. The reporter who sent the photographer to the editor hoped for a good story.

b. The reporter who the photographer sent to the editor hoped for a good story.

Multiple empirical reading time measurements reviewed by Gibson (2000) support these predictions of DLT both for English and Japanese, finding higher reading times for object-extracted than for subject-extracted relative clauses.

A second classic domain concerns nested centre embeddings where DLT posits higher processing difficulty the larger the number of embedded clauses. The location of slow-down is attributed to the closing location, with reading times progressively increasing as embedding depth decreases due to the number of intervening discourse referents growing (Gibson, 2000). See (4) for an example, again taken from Gibson (2000). DLT predicts that integration costs are larger for the two verbs *attacked* and *disliked* in (4b) than in (4a) due to the intervening referents *John* and *met* in the second embedding level.

(4) a. The reporter who the senator attacked disliked the editor.

b. The reporter who the senator who John met attacked disliked the editor.

Empirical data shows that nested centre embeddings are difficult to process for humans, with sentences that have an embedding depth of three or larger often being nearly incomprehensible Miller and Chomsky (1963). Measurements by Warren and Gibson (2002) show elevated processing cost at the verbs that close embedded dependencies, supporting the predictions of DLT.

4.3.1.5 Relation to other theories

The following subsections will relate the Dependency Locality Theory to the architectural paradigms introduced in Section 4.1.5 (serial/parallel, modular/interactive) and to surprisal theory which was presented in Section 4.2.1.

Architectural debates With respect to serial and parallel processing, DLT is compatible with incremental and serial, word-by-word processing of the input in a serial manner. However, it assumes that multiple incomplete syntactic dependencies may be maintained simultaneously in working memory. In this sense, DLT allows a limited form of parallelism at the level of representations but not with respect to alternative analyses. At the same time, because storage cost increases with the number of active dependencies, DLT effectively penalises the maintenance of many parallel structures.

With respect to modular or interactive processing, the original formulation of DLT is largely agnostic on this matter. It assumes syntactic constructions to be retrieved but does not specify whether and how its actions are influence by non-syntactic factors Gibson (2000). The cost of introduction and integration actions are dependent on whether new items refer to new discourse referents, extending beyond purely syntactic structure. However, this specifies an output of the parsing action and not the parsing decision itself as assumed in interactive models.

Surprisal DLT and surprisal often make similar qualitative predictions for certain syntactic constructions but attribute them to different mechanisms and to different locations in the phrase. These cases are investigated and analysed in depth by Levy (2008). The following selection will give an overview over the most relevant results.

One important source of deviations between surprisal and Dependency Locality Theory are head-final local syntactic dependencies in languages such as German or Hindi. DLT predicts greater processing difficulty for words with a higher number of left dependents (Konieczny, 2000) while surprisal predicts the opposite: faster reading times due to more information accumulating about the missing head (Levy, 2008). Levy (2008) show that surprisal models reading times closer than DLT for German verb-final phrases. See Example (5) taken from Konieczny (2000).

(5) a. *Er hat den Abgeordneten begleitet, und ...*
He has the delegate escorted, and ...

'He escorted the delegate, and ...'

b. *Er hat den Abgeordneten ans Rednerpult begleitet, und ...*
He has the delegate to_the lectern escorted, and ...

'He escorted the delegate to the lectern, and ...'

c. *Er hat den Abgeordneten an das große Rednerpult begleitet, und ...*
He has the delegate to the big lectern escorted, and ...

'He escorted the delegate to the big lectern, and ...'

As Konieczny (2000) explain, the verb *begleitet* should have the lowest reading time in case (5a) according to DLT due to fewer dependents and intervening referents compared with (5b) and (5c). Accordingly, (5c) would be hardest to process. However, the reading times measured by Konieczny (2000) exhibit the opposite tendency, establishing what is known as an *anti-locality effect*.

Surprisal, estimated by Levy (2008) using a PCFG extracted from a natural German corpus, does model the facilitatory effect of prepositional phrase insertion. Levy (2008) explain this preverbal facilitatory effect noting that the observation of constituents of a certain type in a head-final clause decreases the expectation of a second occurrence of that type, thus increasing the expectation of a final head. Concretely, seeing the goal prepositional phrase *ans Rednerpult* in (5b) eliminates the possibility of a goal prepositional phrase appearing as the next item.

Levy (2008) demonstrate that this effect is not reliant on the insertions hinting towards the identity of the upcoming verb by showing that it also holds for English subject-modifying relative clauses. The longer the clause, the faster the following main verb is read, despite the relative clause modifying the verb and having no semantic relation to the verb (Levy, 2008).

For object relative and subject relative clauses as in Example (4), surprisal and DLT make similar predictions. As Levy (2008) note, surprisal predicts a slow-down for object relative clauses since they appear less often in text. The predictions of DLT were explained in Section 4.3.1.4. However, processing slow-down is attributed to different locations in the two theories: surprisal locates it at the subject before the relative clause verb since that word indicates the appearence of an object relative clause while DLT predicts it at the integration site, i.e. at the relative clause verb (Levy, 2008). Furthermore, the object relative clause verb is read faster than the verb in the subject relative clause, according to surprisal theory, due to the evidence provided by the preceding noun (Levy, 2008). Demberg and Keller (2008) find only low correlations between DLT integration costs and surprisal which indicates that they might reflect two different aspects of processing. Levy (2008) suggest while reading times hint towards major surprisal effects, non-local, long-distance dependencies (relativisation, *wh*-questions, ...) may be subject to a different mechanism than local dependencies, giving rise to noticeable integration and retrieval costs.

According to Levy (2008), empirical data supports the DLT analysis with respect to the location of the processing slow down in object relative clauses. Grodner and Gibson (2005) tested whether the observed effect can be explained with spillover of surprisal from the noun onto the verb by inserting noun modifiers between noun and verb into the object relative clause. However, no elevated reading times could be detected in the modifier region, making spillover unlikely. Furthermore, in agreement with DLT and despite prior findings with respect to accumulation of evidence in main clause head-final constructions, verb reading times increase with the insertion of these noun-modifiers.

While these comparisons appear to support the DLT in many respects, they are focused on precisely those constructions that originally motivated the theory. In order to gain a reliable and representative assessment of DLT's plausibility – including diverse kinds of syntactic constructions – more recent research has performed regression analyses on naturalistic corpora of extended text. The results paint a more nuanced picture and is discussed in the following section.

4.3.1.6 Limitations and extensions

As discussed above, the main challenge for DLT comes from anti-locality effects. Broad-coverage corpus work paints a mixed picture. Using the Dundee corpus, Demberg and Keller (2008) found no evidence of a general effect of DLT-style integration cost on reading times. Contrarily, they observed significant facilitation of processing in cases of long dependencies. A positive effect was observed, however, for nouns and auxiliary verbs. This observation is supported by Rathi (2021)'s findings which reveal a significant negative overall effect of integration cost on eye-tracking measures in a model that includes surprisal as a predictor. This suggests that the observed anti-locality effects might not be explainable purely by the accumulation of probabilistic evidence (Levy, 2008) which is measured by surprisal. Again, modelling only the reading times of noun-phrases yields a significant positive effect. Mixed evidence is provided by Mielczarek et al. (2025) who report negative effects of DLT-inspired costs on eye-tracking data but small positive effects on self-paced reading times.

In summary, overall evidence from naturalistic corpora suggests that DLT's predictions are either true only for a subset of syntactic constructions or that they are masked by other contributors to reading times, blurring its effects. Both Levy (2008) and Rathi (2021) advocate for developing joint frameworks that operationalise both prediction-based and locality-based paradigms, with limited development in this direction so far. This approach is non-trivial to implement due to two factors: (1) surprisal assumes parallelism (see Section 4.2.1.7) while DLT does not and (2) it is not apparent what amendments to enact on DLT in the light of its predictions on the verbal category, the site of the phenomena that originally motivated its development, being incorrect.

Another possibility is that the syntactic formalism assumed by DLT and/or the distance measure of activation decay proposed by it are improper. Against this backdrop Isono (2024) propose an extension of DLT to syntactic distance based on Combinatory Categorial Grammar (Steedman, 1996), reporting a significant effect on self-paced reading times even in the presence of surprisal. Another route is taken by Mielczarek et al. (2025), extracting a distance-based measure not from annotated parse trees or from the output of a performant parser that includes bidirectional contexts but from an incremental transformer model that has been trained to conduct parsing in its attention mechanism. By doing this, they aim to model costs due to mistakes in integration, e.g. in ambiguous cases. While they observe positive effects on self-paced reading times, it is unclear whether their findings can be generalised since they use an approximation that only takes into account the length of the longest dependency.

4.3.2 Cue-based retrieval

In the preceding section, I presented Dependency Locality Theory which explains processing difficulty in terms of the distance between syntactic dependencies and the number of intervening discourse referents. However, this account leaves open an important question: *how* previously processed linguistic material is accessed when a dependency is resolved. Cue-based retrieval models address this aspect by providing an explicit computational account of memory access during sentence processing (Van Dyke and Lewis, 2003; Lewis and Vasishth, 2005; Van Dyke and McElree, 2006). In these models, integration is treated as a process of retrieving previously encoded representations from a *content-addressable memory system* (McElree et al., 2003) using a set of retrieval cues. Processing difficulty arises not (only) from the distance between dependent

elements, but from the interaction between retrieval cues and competing memory representations.

This perspective allows for a more fine-grained explanation of processing phenomena. In particular, it predicts that difficulty depends on the similarity between relevant items in memory. As a result, cue-based retrieval models can account for effects that are not straightforwardly predicted by distance-based theories, such as similarity-based interference and retrieval errors.

In the following, the theoretical assumptions of cue-based retrieval are introduced, followed by a description of a prominent implementation and a formal linking hypothesis. Lastly, its empirical support, relations to other theories and limitations are discussed

4.3.2.1 Motivation and assumptions

The central motivation for the cue-based retrieval approach is the phenomenon of *similarity-based interference*. Gordon et al. (2001) observed that processing relative clauses us more difficult when the nominal phrases involved are similar. See for instance Example (6) from Gordon et al. (2001) where both NPs *The banker* and *the barber* are equal in grammatical number. Gordon et al. (2001) find increased reading times at the verbs *praised* and *climbed*, which relate to the nominal phrases syntactically.

(6) The banker that the barber praised climbed the mountain.

Gordon et al. (2002) made comparable observations for cleft-structures. Thus, in certain contexts, the complexity of phrases can be increased due to multiple words with similar characteristics. Cue-based retrieval attributes this phenomenon to ambiguity in memory access.

Structurally, cue-based retrieval assumes that comprehenders incrementally retrieve a constituent parse tree while processing a sentence through the means of integration (Van Dyke and Lewis, 2003) – similar to DLT. When enountering a new word that should be integrated, retrieval of previously established constituents from memory has to be performed. This is done by issuing a query to memory and matching the features of the newly enountered word with the features of previous items (Van Dyke and Lewis, 2003). Following the original proposal of Van Dyke and Lewis (2003), these are grammatical features such as part-of-speech, number, genus and case, since they identify syntactically distinct analyses in cases of ambiguities.

A defining assumption of this approach is that memory access is *content-addressable*: all stored items are simultaneously available for retrieval and are evaluated directly with respect to the retrieval cues, rather than being accessed via a serial search process. The motivation for this are results from a study conducted by McElree et al. (2003), challenging the locality-based account of processing difficulty. Thus, Van Dyke and Lewis (2003) conclude that items in memory are addressable via a parallel matching operation. As this may be intractable, they assume that items in memory maintain an activation level and seize to be available for matching if their activation falls below a threshold.

The probability of retrieving a particular item depends on the degree to which its stored features match the retrieval cues. For instance, the verb *leaves* would need to be integrated with a preceding subject in singular form. The cues might include {+N, +singular} and retrieval checks which items in memory have these features. However, if multiple items in memory (partially) match the cues, these items compete during retrieval since and the probability of returning the wrong item

is increased. This prolongs the retrieval process, explaining the effects observed by Gordon et al. (2001) and Gordon et al. (2002).

Van Dyke and Lewis (2003) also propose a repair mechanism for syntactic revision in cases such as garden-paths which requires additional retrieval operations and the deactivation of already established constituents in memory after disambiguating words. This shall not be the focus here.

4.3.2.2 Implementation

Lewis and Vasishth (2005) provide a computational formalisation of cue-based retrieval that largely follows the principles laid out by Van Dyke and Lewis (2003). For this, they use ACT-R, a general framework of cognition. In the following, I will first introduce ACT-R before explaining how Lewis and Vasishth (2005) use it to implement cue-based retrieval and finally I will formalise their linking hypothesis.

Adaptive control of thought-rational (ACT-R) *Adaptive control of thought-rational* (ACT-R) is an important and widely used general theoretical framework motivated by the considerations on internal representations and activations discussed in Section 4.1.5 (Anderson, 1996; Anderson, 2005; Anderson et al., 2004). It is based on the assumption that humans solve problems by minimising processing costs while maximising reward (Whitehill, 2013). ACT-Rs predictions are not restricted to sentence processing but refer to human behaviour on other domains as well, such as problem solving or memorisation.

ACT-R assumes a modular organisation of cognition consisting of several interacting memory systems with limited capacity (Reichle, 2021, Appendix A). A central distinction is made between *declarative memory*, which stores knowledge (e.g. facts and meanings) in the form of discrete *chunks*, and *procedural memory*, which consists of *productions*, i.e. condition-action rules that govern behaviour (Reichle, 2021, Appendix A). Processing happens serially in discrete *production cycles*. During each cycle, information available in the system's buffers is matched against the conditions of productions in parallel (Anderson et al., 2004). Finally, a *goal module* keeps track of the current intent and state of processing (Anderson et al., 2004).

Procedural memory At each processing step, one production is selected from multiple available options and executed. Following Anderson et al. (2004), the selection is performed based on the productions' learned utility defined as

$$U_i = p_i \cdot G - C_i, \tag{51}$$

for production i, where p_i is the estimated probability that the production will lead to the current goal of the system being achieved, G is the value of the goal, and C_i is the associated estimated cost to achieve the goal. Anderson et al. (2004) assume that the probabilities p_i are optimised following a Bayesian framework.

The probability of a production takes into account the utility of all productions options. Let $\mathcal{P}$ denote the set of applicable productions. The probability of selecting production $i \in \mathcal{P}$ is given

by a choice rule that normalises utilities to a probability simplex via the softmax function,

$$P(\text{apply } i) = \frac{\exp\left(\frac{U_i}{t}\right)}{\sum_{j \in \mathcal{P}} \exp\left(\frac{U_j}{t}\right)}, \tag{52}$$

where t is a free parameter controlling the degree of stochasticity (softmax *temperature*, see Section 5.4.1.1) (Anderson et al., 2004).

The procedural memory contains *production rules*, pairs of condition and action (Anderson, 1996). Production actions can be outside-directed, such as moving the hand, or internal, such as the retrieval of an item from declarative memory, modifying a value in the goal, changing the goal (Vasishth and Lewis, 2004) or activating the meaning of a word (Reichle, 2021, Appendix A).

Declarative memory Declarative knowledge is represented as chunks, i.e. units of information. At each cycle, typically one unit can be retrieved from memory and put into a buffer or written to memory, if triggered by a production (Anderson et al., 2004). Each chunk i is associated with an activation value A_i that determines its accessibility (Anderson et al., 2004) and is defined as

$$A_i = B_i + \sum_j W_j S_{j,i}, \tag{53}$$

where B_i is the base-level activation, W_j are attention weights over elements that are currently in the goal module and $S_{j,i}$ denotes associative strengths between goal elements and the chunk in memory (Anderson et al., 2004). W_j are usually set uniformly to $\frac{1}{m}$ with m being the number of total elements in the goal (Reichle, 2021, Appendix A). The base-level activation B_i captures effects of recency:

$$B_i = \log \sum_{k=1}^{n} t_k^{-d}, \tag{54}$$

where t_k denotes the time that has passed since the k-th activation of the chunk, n is the number of past retrievals and d is a free decay parameter (Anderson et al., 2004). Thus, the more time has passed since the last activation of a chunk, the smaller its base-level activation is. Importantly, decay is not just a monotonic function of the recency of creating the memory entry (Lewis and Vasishth, 2005).

Finally, the values of the associative strengths $S_{j,i}$ between goal element j and memory chunk i are specified by

$$S_{j,i} = S - \log \mathit{fan}_j, \tag{55}$$

where S is another free parameter determining the upper limit for the associative strength between two items and fan_j denotes the number of chunks in memory that correspond to the goal item j. Thus, the more items in memory a goal chunk is associated with, the smaller the strength of association with any one of them ends up (Schneider and Anderson, 2012). This accounts for the so-called *fan-effect* (Schneider and Anderson, 2012).

Finally, the probability of successfully retrieving a chunk increases with its activation, while retrieval latency decreases as a function of activation. This establishes a direct link between memory accessibility and processing time. Given a chunk i with activation A_i, ACT-R assumes that

retrieval is stochastic (Anderson et al., 2004). The probability that it is successfully retrieved is modelled using a logistic/sigmoid function:

$$P(\text{retrieve } i) = \frac{1}{1 + \exp\left(-\frac{A_i - \tau}{s}\right)}, \tag{56}$$

where $s > 0$ is a free parameter controlling the degree of stochasticity in retrieval and τ is a threshold (Anderson et al., 2004). If retrieval is successful, the latency of retrieving chunk i is assumed to decrease exponentially with activation:

$$T_i = F \cdot \exp(-A_i), \tag{57}$$

where $F > 0$ is a scaling parameter (Reichle, 2021). This latency is assumed to govern human response and processing times.

ACT-R for cue-based retrieval Lewis and Vasishth (2005) use ACT-R as a framework for implementing a cue-based retrieval parser model of sentence processing. They use the declarative memory as both a long-term and a short-term memory: lexicon entries for lexical items are saved in memory as well as information on the words just processed. The actual short-term memory is comparably small, consisting only of single-item retrieval, control (goal) and lexical input buffers (Lewis and Vasishth, 2005). All remaining items have to be retrieved from declarative memory if needed.

The constituent structure assigned incrementally to the sentence prefix is represented via chunks in declarative memory, with each chunk being associated with a lexical item and containing a set of features (Lewis and Vasishth, 2005). Some features might describe syntactic relations defined between chunks. Lewis and Vasishth (2005) specify that one chunk is retrieved in each step, amounting to a parsing decision. From this follows that only one parse tree is maintained at any given point in time and that each new word is immediately integrated into the existing structure. This corresponds to strict incrementallity and connectedness.

The goal matched with the memory consists of the cues, i.e. features derived from the new lexical item and the current state of processing (Lewis and Vasishth, 2005). In each processing step, the item to be retrieved from declarative memory is determined by the highest activation $\arg\max_i A_i$ and the overall latency of the retrieval operation is given by T_i. Since A_i is lower in cases of more competition, the resulting latency is higher. Thus, similarity-based interference is a manifestation of the fan-effect in this framework. Here, fan_j gives the number of items in memory that have features corresponding to cue j of the set of cues.

With respect to procedural specifications, Lewis and Vasishth (2005) define a serial process defined by the following sequence: lexical access for a newly read word, creating cues, performing retrieval on the declarative memory, creating new structure, adding it to memory and updating the goal. Cues are created based on the new lexical item and the current state of processing to be adapted to the given the context. New structure on the other hand is established based on the new lexical item and the retrieved item, pushing not only syntactic features of the new item to memory but potentially also pointers to the retrieved chunk.

Linking hypothesis Omitting processing difficulty caused by the establishment of cues and the creation of new structure after retrieval, both assumed by Lewis and Vasishth (2005) to contribute to reading times, the cue-based retrieval hypothesis in ACT-R is given by the following equation.

Hypothesis 4.9 (Cue-based retrieval hypothesis)**.**

$$\mathrm{RT}(w_i, w_{<i}) \propto T_{\arg\max_i A_i}, \qquad \forall\, w_1 \ldots w_i \in V^*$$

Under this hypothesis, reading time at position i increases when the activation of the target item is low or when competing items reduce its relative activation.

4.3.2.3 Empirical support

Cue-based retrieval models make a range of empirically testable predictions regarding reading times and error patterns during sentence processing. A central prediction concerns *similarity-based interference*: when multiple memory items share features with the retrieval cues, they compete during retrieval, reducing the activation of the target item. This results in increased retrieval latency and higher probability of misretrieval. Experimental evidence from self-paced reading and eye-tracking studies supports this prediction. For instance, Van Dyke and McElree (2006) and Van Dyke (2007) show that structurally irrelevant but feature-matching distractors can increase reading times at dependency resolution sites, indicating that interference arises from feature overlap.

Another phenomenon shown to be explainable with a cue-based retrieval account is subject-verb agreement attraction: sentences such as *The key to the cabinets are on the table* (Bock and Miller, 1991) are read without slow-down and judged acceptable despite the agreement error between *key* and *are*. Wagers et al. (2009) demonstrates that this can be accounted for by assuming that *cabinets* yields higher associative strength than *key* when processing *are* due to agreement in number between *are* and *cabinets*. Thus, feature-matching distractors can lead to faulty parses that are faster to process.

Overall, these results provide broad evidence that content-addressable retrieval with competition offers a plausible mechanistic account of memory effects in sentence processing.

4.3.2.4 Relation to other theories

The following subsections will relate cue-based retrieval to the architectural paradigms introduced in Section 4.1.5 (serial/parallel, modular/interactive) and to surprisal theory (4.2.1), entropy reduction theory (4.2.2) and Dependency Locality Theory (4.3.1).

Architectural debates Cue-based retrieval assumes that memory access is content-addressable, such that all stored representations are evaluated simultaneously with respect to the retrieval cues. Retrieval therefore proceeds in parallel, with multiple items competing based on their degree of match to the cues, rather than via a serial search through memory. At the same time, sentence processing is assumed to be serial, with each retrieval operation returning one item from memory at which the most recently read word is integrated. Thus, only one analysis of the sentence is maintained at any point in time. If content turns out to be incompatible with this analysis, an explicit repair mechanism needs to be triggered to restructure this analysis (Van Dyke and Lewis, 2003). For this reason, cue-based retrieval is a serial paradigm.

The question of modularity or interactiveness depends on the set of features selected from the lexicon for the cues. The original proposal of Van Dyke and Lewis (2003) assumed that retrieval is primarily guided by syntactic factors which is why grammatical features were encoded. This

corresponds to a modular approach. However, there is no inherent restriction on the kind of featureset that could be represented in the framework. Thus, later research combines both syntactic and semantic features (Van Dyke, 2007), making them jointly available in an interactive setting.

Surprisal Surprisal theory and cue-based retrieval propose two only weakly related explanations for processing latency that can diverge substantially in their predictions. Surprisal assumes difficulty whenever a continuation is improbable, independent of the specification of memory operations. Cue-based retrieval, in contrast, makes explicit assumptions with repect to memory calls triggered by the continuation, with difficulty depending on the accessibility of the retrieval target and the similarity of competing representations, even when predictability of the continuation is high. See for instance Example (6): surprisal would support lower reading times at the second verb due to accumulated evidence of its occurrence, however, Gordon et al. (2001) find that RTs increase at both verbs, compatible with the assumption that both items trigger calls into memory which get prolonged through NP interference.

On the other hand, the theories can be treated as complentary rather than competing. Levy et al. (2013) report evidence that both theories complentarily explain reading times. They observe expectation-based difficulty at clause-initial regions and memory-based difficulty at integration sites in Russian relative clauses. Levy et al. (2013) suggest that both expectation-based and memory-based theories should be operationalised in a joint framework and that their relative strengths in explaining reading times might be language-dependent, for instance specific to word order.

However, such a joint integration raises theoretical questions since the interpretation of surprisal as quantifying the change in distributions over possible analyses of the sentence (Section 4.2.1.3) is not compatible with cue-based retrieval since it assumes maintaining multiple analyses for the sentence prefix in parallel. Thus, either such a framework has to be altered for retrieval to return several items that are used to construct and save different alternative analyses or surprisal needs to be interpreted as the result of a distinct prediction process conditioned on the elements available in the buffers, e.g. for increasing the speed of lexical identification (see Section 4.2.1).

Entropy reduction hypothesis Cue-based retrieval captures interference from competing representations in memory: larger numbers of constituents that share features with the most recently read item are linked to increased processing time. This competition measure is similar to entropy which captures the diffuseness of a distribution. In fact, it can even be expressed via a measure of entropy if taking $\mathrm{H}(P)$ of the retrieval distribution P defined in Equation 56. Recent models inspired by cue-based retrieval use entropy to measure competition of retrieval-like operations (discussed in Section 5.4.2). Despite this notion, cue-based retrieval should not be confused with the entropy reduction hypothesis.

The set of items over which uncertainty is quantified differs. Entropy reduction measures uncertainty over alternative analyses for the whole prefix maintained in parallel. Cue-based retrieval, on the other hand, measures diffuseness with respect to returning an element from memory that determines the continuation of the single maintained analysis. The underlying assumptions that uncertainty needs resources to resolve is shared. However, in cue-based retrieval, new uncertainty is introduced at each step, with high competition at position i leading to a higher reading time at i. Uncertainty is resolved immediately by deciding on one item to retrieve from memory. In ER,

high RTs are explained with a reduction in uncertainty between two processing steps, tracking a global sentence-wide process.

In the more recent entropy reduction defined on lexical continuation probabilities (e.g. Frank, 2013; Song et al., 2024), the entropy measure resets at each processing step, similarly to cue-based retrieval. However, aside from it being derived from a distribution over different items than cue-based retrieval, it does not quantify the reduction of uncertainty over a single set of events.

Dependency Locality Theory Cue-based retrieval and Dependency Locality Theory are closely related in that both attribute part of sentence processing difficulty to memory limitations. However, they differ at the explanatory level. DLT predicts that dependencies spanning more intervening referents should be harder to process, whereas cue-based retrieval predicts that difficulty depends on the degree to which intervening items match the retrieval cues. Thus, the contributors to distance are fixed in DLT to depend only on reference while they are dynamically dependent on their identity and the identity of the retrieval cues in cue-based retrieval. As a consequence, two dependencies of equal structural length may differ in difficulty under a cue-based account if the intervening material differs in feature overlap with the target. This is supported by studies showing that retrieval difficulty is modulated by distractor similarity while keeping distance constant (Van Dyke and Lewis, 2003). Additionally, the predictions might also differ in cases where the intervening item is located left of the item to retrieve for the analysis as the cue-based theory induces no fixed ordering on the memory. On the other hand, Dependency Locality Theory predicts longer reading times for long dependencies with intervening referents even if those referents exhibit no feature overlap with the retrieval target.

Another difference lies in the syntactic operationalisation: in cue-based retrieval, at each step, there is always an integration triggered to incrementally attach a new item to the single syntactic analysis. In DLT, on the other hand, storage costs insinuate that items still lacking syntactic heads may be stored as incomplete items (Gibson, 2000). This is necessary if only retrieving syntactic dependencies between surface items, since new words might provide no syntactically valid options for relations with preceding items (see also Mielczarek et al. (2025) on this matter who simulate integration with a dummy item in such cases). In this sense, cue-based retrieval is strictly incremental and connected, while DLT does not require the latter.

In practice, the two approaches are often correlated and make similar qualitative predictions in classic locality configurations. In relative-clause and other long-distance dependency constructions, both DLT and cue-based retrieval tend to predict greater difficulty when integration must take place over more intervening material. Experimental work on Russian relative clauses, for example, has been interpreted as consistent with both distance-based locality accounts and cue-based retrieval accounts (Levy et al., 2013). Generally, research struggles to compare predictions from both theories using corproa-based methods due to their similarity.

Lastly, it is important to note that the cue-based retrieval implementation of Lewis and Vasishth (2005) includes activation-based decay. Thus, latency for a retrieved chunk is influenced by the recency of its establishment and subsequent reactivations. This is not an original characteristic of cue-based retrieval as defined by Van Dyke and Lewis (2003) but a design choice motivated by success of the ACT-R framework. It makes the measure more similar to DLT's locality effect but not equivalent: (1) it measures distance not in intervening referents but in processing steps and (2) DLT's decay is monotonic whereas reactivations of chunks through retrieval can reduce high

locality-based costs for accesses that follow.

4.3.2.5 Limitations and extensions

Despite their explanatory power, cue-based retrieval models face several theoretical and empirical challenges. A first limitation concerns the scope of the theory. Cue-based retrieval provides a detailed mechanistic account of memory access during integration, but it does not in itself specify how syntactic structures are constructed. As a result, it is typically combined with parsing frameworks, such as left-corner parsing (Aho and Ullman, 1972) in Lewis and Vasishth (2005).

A second challenge concerns the relative contributions of interference and decay assumed by the ACT-R-based implementation of Lewis and Vasishth (2005). While the model predicts both effects, the empirical findings that motivated cue-based retrieval suggest immediacy in contenct-access regardless of distance or decay Gordon et al. (2001) and Gordon et al. (2002). This raises questions about the extent to which locality effects should be attributed to memory decay as opposed to competition between representations.

A practical issue is that cue-based retrieval models rely on the specification of features and cues, which are typically assumed rather than derived from independent principles. The choice of cues can substantially influence model predictions. This introduces a degree of flexibility that can make it difficult to distinguish between alternative explanations of empirical effects. This issue is especially pressing in the light of semantic interference effects found by Van Dyke (2007) although weaker than syntactic effects (Van Dyke and McElree, 2011). Semantic features are notourously hard to specify. Jäger et al. (2017)'s meta-analysis suggests that multiple modifications of Lewis and Vasishth (2005)'s model are necessary to account for interference effects widely varying across types of dependencies. Thus could also hint towards the selection of features being inadequate.

Motivated by this, a recent line of work relates cue-based retrieval to attention-based neural networks (transformers, see Section 5.3), combining retrieval competition with surprisal-based expectation measures (Ryu and Lewis, 2021; Ryu and Lewis, 2025). Thanks to their optimisation procedure, these models automatically learn continuous feature specifications as part of their parameterisation, doing away with the need for manual selection. These approaches provide a potential bridge between. A detailed discussion is given in Section 5.4.2).

Overall, similarly to surprisal and Dependency Locality Theory, there is evidence for the validaty of interference effects due to cue-based retrieval. At the same time, it does not provide a full account of processing difficulties with researchers having identified various explanatory gaps in the theory. While "all models are wrong" (Box, 1976), the preceding discussions suggest that the present accounts do not yet yield a sufficent model of human sentence processing reality. This stresses the need for a unified account.

5 Language models

In Section 4.2.1, surprisal theory was introduced as a linking hypothesis relating human reading times to the conditional probability of a word given its left context. In empirical work, these probabilities are typically approximated using computational *language models*.

Definition 5.1 (Language model)**.** Let V be a vocabulary. A *language model class with parameter space* Θ is a family of functions

$$\{\, P[\theta](\,\cdot \mid \cdot\,) \mid \theta \in \Theta \,\}$$

such that for each $\theta \in \Theta$ and each prefix $w_{<i} \in V^*$, the mapping

$$P[\theta](\,\cdot \mid w_{<i}\,)\colon V \to [0,1]$$

defines a probability distribution over the next token. Θ is the parameter space of the model, i.e. the set of all admissible parameter configurations θ specified by the model architecture. Each $\theta \in \Theta$ defines a *model instance*, i.e. a specific selection of conditional word-probabilities $P[\theta](\,\cdot \mid w_{<i}\,)$ for all prefixes $w_{<i} \in V^*$.

In the following, the term *model* can refer to a model class or a model instance, depending on context. When necessary, this distinction is made explicit.

A joint probability over complete sequences is induced via the chain rule (Jurafsky and Martin, 2026, Chapter 3.1):

$$P[\theta](\,w_1 \ldots w_n\,) = \prod_{i=1}^{n} P[\theta](\,w_i \mid w_{<i}\,). \tag{58}$$

The parameters $\theta \in \Theta$ are estimated from a large corpus of natural text using an estimation algorithm.

Definition 5.2 (Estimation algorithm)**.** An *estimation/training algorithm* $\mathcal{A}_\Theta$ for a vocabulary V and a language model class with parameter space Θ is a computational procedure that can be defined as a function mapping the character sequence of a corpus to a parameter configuration,

$$\mathcal{A}_\Theta\colon V^* \to \Theta.$$

Thus, for a language model with parameter space Θ, the corpus and the training algorithm jointly determine the fitted parameters $\theta \in \Theta$. Using corpora comes with the assumption that they reflect the linguistic variety representative of the language that should be modelled.

The usual approach for algorithm $\mathcal{A}$ is *maximum likelihood* (ML) *estimation* (Jurafsky and Martin, 2026, Chapte 3.1). Given a corpus string s over vocabulary V, the likelihood of the data under a model instance $P[\theta](\,\cdot \mid \cdot\,)$ is defined as

$$\ell(\,\theta \mid s\,) \coloneqq P[\theta](\,s\,). \tag{59}$$

The likelihood is the probability that the model with parameters θ assigns to the observed training sequence, viewed as a function of the parameters. Note that it does not constitute a probability distribution over possible parameters $\theta \in \Theta$. The goal of the estimation process is to find the

parameters that produce the maximum likelihood to get a model that most accurately describes the training data (Bishop, 2006, Chapter 1.2).

Thus, the maximum likelihood estimate is given by

$$\theta^* \coloneqq \arg\max_{\theta \in \Theta} \ell(\, \theta \mid s \,). \tag{60}$$

It is common to minimise the logarithm of the likelihood instead since this allows to express the joint probability as an addition (Bishop, 2006, Chapter 1.2). This is equivalent to Equation 60 since the logarithm is a monotonic function.

$$\begin{aligned} \theta^* &= \arg\min\nolimits_{\theta \in \Theta} \mathcal{L}[\theta](\, s \,), \\ \text{with} \quad \mathcal{L}[\theta](\, s \,) &\coloneqq -\log \ell(\, \theta \mid s \,) = -\textstyle\sum_{i=1}^{|s|} \log P[\theta](\, s_i \mid s_{<i} \,) = \sum_{i=1}^{|s|} \mathrm{I}_{P[\theta]}(\, s_i \mid s_{<i} \,). \end{aligned} \tag{61}$$

The *log-likelihood function* $\mathcal{L}$ is also called an *error function* (Bishop, 2006, Chapter 1.2), a *loss function* (Goldberg, 2017, Chapter 2.7) or an *objective function*. Thus, maximum likelihood estimation corresponds to minimising the sum of negative log-probabilities assigned to the observed tokens in the corpus. Equivalently, the term is often defined as the mean loss per training instance by dividing it through $|s|$ (Goldberg, 2017, Chapter 2.7) and can therefore also be interpreted as minimising the average negative log-probability of the next token given its context, i.e. the average conditional surprisal.

Following Goldberg (2017, Chapter 2.7), the loss function can also be interpreted as the sum of *cross-entropies* (Definition 2.28) between the model distributions $P[\theta]$ and the target distributions P^*:

$$\mathcal{L}[\theta](\, P^*, P[\theta] \,) = \sum_{i=1}^{|s|} \mathrm{H}(P^*(\, \cdot \mid w_{<i} \,), P[\theta](\, \cdot \mid w_{<i} \,)). \tag{62}$$

The intuition is that the measure quantifies the difference between model output and goal output, which should be minimised. In the case of language modelling, the target distribution yields 1 for the single token w_i actually found in the training data and 0 for all other tokens. Thus, cross-entropy simplifies to $-\log P[\theta](\, w_i \,)$ which is equal to $\mathcal{L}[\theta](\, s \,)$ (Equation 61).

Broadly, language models can be divided into (1) traditional statistical models, such as n-gram models, and (2) neural network-based models. In the following, the classical n-gram approach is briefly reviewed as a conceptually simple baseline. Subsequently, neural language models are introduced, describing recurrent architectures schematically and defining transformer-based models in detail. The latter will provide the basis for the experiments in this work.

5.1 n-gram models

A conceptually simple and historically important class of language models is given by *n-gram models*. They are based on a finite-context assumption, according to which the probability of a token depends only on a fixed number of preceding tokens (Jurafsky and Martin, 2026, Chapter 3.1). Early work on this concept includes Shannon's (1948) seminal paper where they introduced the idea of approximating English using finite-order Markov processes.

5.1.1 Parameterisation

Definition 5.3 (n-gram model)**.** Let V be a vocabulary and let $n \in \mathbb{N}$. An *n-gram model* is a language model class in which the conditional probabilities form a Markov chain of order $n-1$ (Jurafsky and Martin, 2026, Chapter 3.1), i.e. for all $w_{<i} \in V^*$,

$$P[\theta](\, w_i \,|\, w_{<i} \,) = P[\theta](\, w_i \,|\, w_{i-n+1} \ldots w_{i-1} \,), \tag{63}$$

where contexts shorter than $n-1$ are handled by suitable boundary conventions.

Thus, the model conditions only on the last $n-1$ tokens of the prefix. Consequently, 1-gram models quantify word frequency.

Following this idea, the parameters of an n-gram model instance simply consist of a table of conditional probability distributions, one for each context $c \in V^{n-1}$. This can be operationalised via probability simplexes.

Formally, the parameter space is given by

$$\Theta = \{\, \theta : V^{n-1} \to \Delta^{|V|} \,\}.$$

Given a parameter configuration $\theta \in \Theta$, the conditional probabilities of the language model are defined by

$$P[\theta](\, w_i \,|\, w_{<i} \,) = P[\theta](\, w_i \,|\, w_{i-n+1} \ldots w_{i-1} \,) = \theta\big(w_{i-n+1} \ldots w_{i-1}\big)_{(\mathrm{idx}_V(w_i))}, \tag{64}$$

where idx_V induces an ordering on V.

5.1.2 Estimation

Let $\mathrm{count}(cw)$ denote the number of occurrences of the n-gram cw in a corpus and $\mathrm{count}(c)$ the number of occurrences of the context c. Then, following Jurafsky and Martin (2026, Chapter 3.1), the maximum likelihood estimate is given by

$$\theta(c)^*_{(\mathrm{idx}_V(w))} = \frac{\mathrm{count}(cw)}{\mathrm{count}(c)}. \tag{65}$$

This estimator is intuitive but suffers from data sparsity: many possible n-grams may not be observed in the corpus, leading to probabilities of zero. This problem becomes even more pressing for larger values of n since this leads to more unobserved contexts cw, increasing sparsity.

To address this, *smoothing* techniques modify the maximum likelihood estimate to assign non-zero probability mass to unseen events. A simple example is *add-one* (Laplace) smoothing which increases all counts of cw by one (Jurafsky and Martin, 2026, Chapter 3.6):

$$\theta(c)^*_{(\mathrm{idx}_V(w))} = \frac{\mathrm{count}(cw) + 1}{\mathrm{count}(c) + |V|}. \tag{66}$$

More advanced methods, such as *Kneser-Ney smoothing*, redistribute probability mass based on lower-order models and have been shown to perform substantially better in practice (Kneser and Ney, 1995; Chen and Goodman, 1999).

5.1.3 Limitations

n-grams yield a simple and tractable language model but are limited in their ability to capture long-range dependencies due to the restriction to a finite context of size $n-1$. Take for instance Example (3a) where we as human processors identify that *reported* and *hoped* are related by syntactic dependency while replacing *hopes* with *go* violates grammatical rules due to disagreement in number. Thus, the event of a word appearing at the position of *hoped* is not independent of the item 8 positions prior. Sentences like these appear thousands of times in typical training material and any model trained on such data is bound to generate ill-formed strings if it cannot condition on appropriate context lengths.

Increasing the context to an appropriate length not only comes with the issue of sparsity discussed in the preceding section but also with the problem that the number of parameters grows exponentially with n since the number of counts to track is determined by $|V|^n$ (Jurafsky and Martin, 2026, Chapter 3.1). The model needs to save a number for all imaginable contexts of size n.

Most contexts we encounter and produce in our daily lifes are entirely novel and unseen but we are still able to understand them and complete them to well-formed expressions. Thus, it is necessary to find modelling assumptions that can generalise from observed contexts to new but similar situations (Jurafsky and Martin, 2026, Chapter 3.1).

5.2 Recurrent neural networks

Neural language models parameterise the conditional distribution $P(w_{i+1} \mid w_1, \ldots, w_i)$ using an artificial neural network. In contrast to n-gram models, which rely on simple statistics extracted through counting, artificial neural networks exhibit learned continuous vector representations that are less interpretable but more flexible. They make it possible to compress context representations to allow for contexts of variable size and enable generalisation across similar contexts.

An *artificial neural network* is a parameterised function $f[\theta](\,\cdot\,)\colon \mathbb{R}^d \to \mathbb{R}^k$ for an input and an output dimensionality $d, k \in \mathbb{N}$ defined as a composition of affine transformations and element-wise nonlinearities. The following exposition assumes familiarity with fundamental concepts in neural networks and their optimisation. For a comprehensive introduction to neural networks in natural language processing, see Goldberg (2017).

We seek two model components for language modelling: (1) a parameterised *model function* $\phi[\theta](\,\cdot\,)$ that maps a prefix $w_1 \ldots w_{i-1}$ to a continuous contextual representation $\mathbf{h}_{i-1} \in \mathbb{R}^d$ with some dimensionality $d \in \mathbb{N}$:

$$\mathbf{h}_{i-1} = \phi[\theta_\phi](\, w_1 \ldots w_{i-1}\,) \tag{67}$$

and (ii) a *head function* $\gamma[\theta_\gamma](\,\cdot\,)$ that maps $\mathbf{h}_i$ to a probability simplex over the vocabulary V, e.g. via a linear projection to logits followed by normalisation:

$$P[\theta](\, w_i \mid w_{<i}\,) = \gamma[\theta_\gamma](\, \mathbf{h}_{i-1}\,)_{(\mathrm{idx}_V(w_i))}, \qquad \theta \coloneqq \langle\, \theta_\phi, \theta_\gamma \,\rangle. \tag{68}$$

The neural language model thus consists of the components $\langle \phi, \gamma \rangle$ and learnable parameters θ.

In the following, I will provide an abstracted schematic of recurrent neural networks focused on their general mode of operation which will help to understand differences between transformer

models and recurrent architectures later. The reader is again referred to Goldberg (2017) for a complete specification.

5.2.1 Abstract notion

Early neural language models were based on *recurrent neural networks* (RNN), including architectures such as *Long Short-Term Memory* (LSTM) networks (Hochreiter and Schmidhuber, 1997). In these models, the contextual representation $\phi[\theta_\phi](\, w_1 \ldots w_{i-1}\,) = \mathbf{h}_{i-1}$ is computed via a recurrent function ρ (Goldberg, 2017, Chapter 14.1):

$$\begin{aligned} &\mathbf{h}_1 \coloneqq \mathbf{0} \qquad \text{(or a learned initial state)}, \\ &\mathbf{h}_j \coloneqq \rho[\theta_\phi](\, \mathbf{h}_{j-1},\, w_j\,) \qquad \forall j \in \{\, 2, \ldots, i-1\,\}. \end{aligned} \tag{69}$$

This introduces an inherent sequential dependency: computing the state at position j requires computing the state at position $j-1$ first. As a result, it is not possible to parallelise model training across sequence positions. This quickly becomes a problem when combining several LSTMs in a stacked system.

5.2.2 Estimation

Model training involves producing for each token instance w_i in the training sequence with $i > 2$ a predictive distribution $P[\theta](\,\cdot \mid w_1 \ldots w_i\,)$.This results in a series of distributions for all next-token targets $w_2, \ldots, w_{|w|}$. Then, the likelihood of the model under parameters θ is defined as

$$\ell(\,\theta \mid w\,) = \prod_{i=1}^{|w|-1} P[\theta](\, w_{i+1} \mid w_1 \ldots w_i\,) = \prod_{i=1}^{|w|-1} \gamma[\theta_\gamma](\, \phi[\theta_\phi](\, w_1 \ldots w_i\,)\,)_{(\mathrm{idx}_V(w_{i+1}))} \tag{70}$$

The goal of the training process is to find the parameters that produce the maximal likelihood to get a model that most accurately describes the observed data. For this, gradients are computed by backpropagation through the computational graph defined by ϕ and γ, and parameters are updated using a variant of stochastic gradient descent (Goldberg, 2017, Chapter 2.8).

The parameters θ are tuned during training. In contrast, quantities such as the dimensionality d, architectural choices (e.g. whether several recurrent networks should be stacked on top of each other), tokenisation strategies and optimisation settings (e.g. learning rate and batch size) are fixed before training and are called *hyperparameters*. These hyperparameters determine the parameter space Θ by fixing the architecture and dimensionality of the model.

5.2.3 Limitations and extensions

Unlike n-gram models, RNNs do not impose a fixed context size. Instead, the hidden state $\mathbf{h}_i$ serves as a compressed representation of the entire prefix $w_{<i}$. This allows, in principle, modelling of long-range dependencies (Goldberg, 2017, Chapter 14).

However, in practice, standard RNNs suffer from difficulties in learning such dependencies due to vanishing and exploding gradients (Bengio et al., 1994). This limitation motivated the development of more advanced recurrent architectures such as *long short-term memory* (LSTM) networks

(Hochreiter and Schmidhuber, 1997) and *gated recurrent units* (GRUs) (Cho et al., 2014) and ultimately led to the establishment of alternative parallel architectures such as the *transformer* (Vaswani et al., 2017) which dominate the field today and are discussed in the next section.

5.3 Transformer

Transformer-based architectures (Vaswani et al., 2017) compute contextual representations by repeatedly updating token representations through stacked layers. At each layer, the representation of a token is computed by attending to representations of all tokens in the preceding layer. Unlike recurrent models, which summarise the prefix in a single hidden state, transformers maintain a representation for each position and allow direct interactions between all positions via attention (Vaswani et al., 2017).

The transformer architecture is the backbone of modern large language models and many contemporary *natural language processing* (NLP) systems. Its success is largely attributed to its scalability and its ability to model long-range dependencies. The architecture allows for effective parallelised training of highly expressive models with billions of parameters (Jurafsky and Martin, 2026, Chapter 8). It has motivated extensive research into which architectural design choices make transformers particularly well suited for modelling language and to what extent these systems may serve not only as models of language production but also of language processing (see Section 5.4).

In the following, I will first give an abstract intuition of the transformer architecture that aligns with the functional notation introduced in the preceding sections. After that, I will define the transformer architecture in detail.

5.3.1 Abstract notion

For an input sequence $w_1 \ldots w_i$ and a model with l layers, intermediate representations $\mathbf{h}_{k,j}$ are defined for each layer $k \leq l$ and each position $j \leq i$. The model function yields the representation of the final layer at the last prefix position:

$$\phi[\theta_\phi](\, w_1 \ldots w_i \,) \coloneqq \mathbf{h}_{l,i}. \tag{71}$$

The layer-wise computation of a decoder-only, causal transformer is defined as

$$\begin{aligned} \mathbf{h}_{0,j} &\coloneqq \mathrm{emb}(w_j), \\ \mathbf{h}_{k,j} &\coloneqq \beta[\theta_k](\, \mathbf{h}_{k-1,1}, \ldots, \mathbf{h}_{k-1,j} \,) \qquad \forall\, k \in [l],\ j \in [i], \end{aligned} \tag{72}$$

where $\beta[\theta_k]$ denotes a transformer block that combines the representations of all input words in the left context, including j, at the preceding (lower) layer $k-1$ and $\mathrm{emb}(w_j)$ is a continuous representation of w_j.

Unlike recurrent models, all positions within a layer can be processed in parallel; the computation is sequential only across layers, not across tokens. Note that this does not mean that the output representations of two tokens in a sequence do not share computational graphs. At position i, a layer takes into account input representations for positions $\{\, 1, \ldots, i-1 \,\}$ which are also used to compute the layer representation for item $i-1$.

5.3.2 Parameterisation

This section introduces the components of the transformer architecture as originally proposed by Vaswani et al. (2017). For each component, the hyperparameter configurations of `GPT-2 Small` and `GPT-2 Medium` – the 124 million and the 355 million parameter variants of Radford et al. (2019)[17] – are provided as an example and reference for the architecture used in the experimental part of this work (see Section 7). In some cases, specifics are not reported in Radford and Narasimhan (2018) and Radford and Narasimhan (2018). In these cases, I will consult the original implementations found at https://github.com/openai/finetune-transformer-lm and https://github.com/openai/gpt-2 and remark this explicitly.

The original transformer was introduced in the context of sequence-to-sequence tasks (Vaswani et al., 2017). It follows an encoder–decoder architecture in which an input sequence $x_1 \ldots x_n$ is first mapped to a sequence of continuous representations $\mathbf{z}_1, \ldots, \mathbf{z}_n$. Conditioned on these representations, the decoder generates an output sequence $y_1 \ldots y_m$ in an autoregressive manner, i.e. each output token is predicted using the previously generated tokens as an additional input (Vaswani et al., 2017).

Subsequent work demonstrated that for language modelling, the encoder component is not required. Decoder-only transformer architectures trained with an autoregressive objective were shown to yield strong performance (Liu et al., 2018). Scaling this paradigm substantially led to the *generative pre-trained transformer* (`GPT`) series of models, which established that large-scale self-supervised[18] language modelling can serve as an effective pretraining strategy for downstream supervised NLP tasks (Radford and Narasimhan, 2018; Radford et al., 2019). Here, I formalise the decoder-only variant exclusively.

5.3.2.1 Dropout

Before defining the components of the transformer, it is necessary to introduce the *dropout* method since it is applied in several places in the model. Dropout is a regularisation technique that was first introduced by Hinton et al. (2012) and Srivastava et al. (2014) to reduce overfitting in neural networks. During training, dropout randomly sets a subset of activations to zero, thereby preventing the model from relying too strongly on specific weights. Intuitively, this forces the network to distribute information across multiple features rather than relying on a small number of highly specialised neurons (Goldberg, 2017, Chapter 4.6).

Definition 5.4 (Dropout)**.** Formally, let $\mathbf{x} \in \mathbb{R}^d$ be an input vector and let $p \in [0, 1)$ denote the dropout probability. The dropout operator is defined as

$$\mathrm{Drop}_p(\mathbf{x}) \coloneqq \frac{1}{1-p}\, \mathbf{m} \odot \mathbf{x},$$

where $\mathbf{m} \in \{\, 0, 1 \,\}^d$ is a stochastic masking vector with independent components sampled from a

[17] Referred to as 117M and 345M in the original paper. The actual parameter counts are slightly larger than reported; see official implementation at https://huggingface.co/openai-community/gpt2-medium.

[18] Originally referred to as *unsupervised* by Radford and Narasimhan (2018). However the training algorithm utilises a supervision signal in the form of the following word. Thus, the term *self-supervised* is usually used (Jurafsky and Martin, 2026, Chapter 7)

Bernoulli distribution

$$\mathbf{m}_{(i)} \sim \mathcal{B}(1-p).$$

At inference time, dropout is disabled and the identity function is applied. The scaling factor $\frac{1}{1-p}$ ensures that

$$\mathbb{E}[\text{Drop}_p(\mathbf{x})] = \mathbf{x}, \tag{73}$$

so that the expected activation magnitude remains unchanged between training and inference (Hinton et al., 2012).

In transformer architectures, dropout is applied in four places (Vaswani et al., 2017) which will be explained in more detail in the next sections. It is applied to the

1. attention weight matrix after the softmax operation (attention dropout, p_{att}),
2. output of the attention mechanism (residual dropout, p_{resid}),
3. output of the feed-forward function in each layer (feed-forward dropout, p_{ff}),
4. sum of token embeddings and positional encodings (embedding dropout, p_{emb}).

For the base transformer model, Vaswani et al. (2017) used a uniform dropout rate of $p = 0.1$ for residual dropout, feed-forward dropout and embedding dropout. Attention dropout appears first in literature with the `GPT-1` model (Radford and Narasimhan, 2018) where it was combined with the other three types of dropout. The rate $p = 0.1$ was also adopted. `GPT-2` follows the same strategy.[19]

5.3.2.2 Tokenisation

The choice of tokenisation has important consequences for the modelling assumptions. The easiest form of tokenisation, also called *word-level tokenisation* (e.g. Alrefaie et al., 2024), separates sentences into word units at white-spaces. While this representation is intuitive and seems to align with traditional units in formal grammar modelling, it leads to very large vocabularies and suffers from severe sparsity, as rare or unseen words cannot be assigned reliable probabilities – especially for languages using rich inflectional morphology. Furthermore, it is not applicable to languages written without white-spaces and a model trained on tokens that were split on the word-level can only be applied to a word that it has not seen during training by replacing it with special `<unk>` token (Gaddy et al., 2018). This gets rid of any useful information the word provides, even if the model has reliable representations for orthographically similar words.

To address these issues, modern language models commonly employ *subword tokenisation* schemes, which decompose words into smaller units (Jurafsky and Martin, 2026, Chapter 2). A widely used approach is *Byte Pair Encoding* (BPE) (Gage, 1994), which iteratively merges frequent adjacent symbol pairs in a corpus to construct a vocabulary of subword units. Starting from a character-level representation, BPE builds larger units by replacing the most frequent pair of

[19] Radford et al. (2019) do not report the exact dropout rates and they only provide code for model inference in their public repository – not for training. They state that `GPT-2` generally follows the specification of `GPT-1` and explicitly list a range of modifications which does refer to dropout.

neighbouring symbols with a new symbol representing its concatenation, repeating this process until a predefined vocabulary size is reached (Gage, 1994).

This procedure yields a tokenisation that balances two competing goals: on the one hand, frequent words may be represented as single tokens, reducing sequence length compared to character-based representations. On the other hand, rare or unseen words can be represented by smaller subword units, allowing the model to generalise to novel forms (Jurafsky and Martin, 2026, Chapter 2). Furthermore, subword tokenisation reduces the vocabulary to a manageable size – 50.257 for `GPT-2` (Radford et al., 2019).

In language modelling, BPE merges are typically restricted to occur within white-space boundaries The BPE-training procedure is given in Algorithm 1 following Sennrich et al. (2016). It requires a sequence of words already split at white-spaces or other appropriate positions possibly provided by labelling, via an *initial tokenisation*. This ensures that the subtokens identified by the BPE algorithm do not span several words.

Definition 5.5 (Initial white-space tokenisation)**.** Let V be a character vocabulary. The *initial tokenisation* is a mapping

$$\tau_0 \colon V^* \to ((V^1)^*)^*.$$

For a character sequence $s \in V^*$, let

$$\langle\, u_1, \ldots, u_n \,\rangle$$

be the sequence of maximal non-empty substrings of s delimited by word boundaries, such that $u_1 \ldots u_n = s$, where word boundaries are given either by explicit annotations or inferred from whitespace. For each word $u_i = c_{i,1} \ldots c_{i,m_i}$ with $c_{i,j} \in V$, define

$$\tau_0(s) \coloneqq \langle\, \langle\, c_{1,1}, \ldots, c_{1,m_1} \,\rangle, \ldots, \langle\, c_{n,1}, \ldots, c_{n,m_n} \,\rangle \,\rangle.$$

Thus, each word is represented as a sequence of singleton character sequences.

Algorithm 1 BPE training following Sennrich et al. (2016)

Input: corpus words $\langle\, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_n \,\rangle$ with each $\boldsymbol{\omega}_j$ being a sequence of initial symbol singletons, merge number m

Output: merge list $\mathcal{M}$

Note: Pair counts are computed over all words but only for adjacent symbols within individual words.

$\mathcal{M} \leftarrow \langle\ \rangle$
for $r = 1$ to m **do**
  Count all adjacent pairs $\langle\, x, y \,\rangle$ occurring within words
  $(x^*, y^*) \leftarrow$ most frequent pair
  if no such pair exists **then**
    break
  Append (x^*, y^*) to $\mathcal{M}$
  for each word representation **do**
    Replace all occurrences $\langle\, \ldots, x^*, y^*, \ldots \,\rangle$ in all words with $\langle\, \ldots, x^*y^*, \ldots \,\rangle$
return $\mathcal{M}$

Given a trained merge list $\mathcal{M}$, BPE tokenisation of a corpus consists of repeated merging of the adjacent string pairs prescribed by $\mathcal{M}$. Formally, this is defined as a composition of merge operations.

Definition 5.6 (Merge operator)**.** Let $\langle\, x, y \,\rangle \in (V^*)^2$. The merge operator

$$r_{(x,y)} : (V^*)^* \to (V^*)^*$$

maps a tuple $\boldsymbol{\omega}$ of character sequences to a new tuple obtained by replacing every adjacent occurrence of the pair (x, y) by the single token xy.

Formally, $r_{(x,y)}$ scans $\boldsymbol{\omega}$ from left to right and applies the following rule:

$$\langle \ldots, x, y, \ldots \rangle \;\mapsto\; \langle \ldots, xy, \ldots \rangle$$

where no replacements overlap.

Definition 5.7 (BPE tokenisation)**.** Let

$$\mathcal{M} = \langle\, (x_1, y_1), \ldots, (x_{|\mathcal{M}|}, y_{|\mathcal{M}|}) \,\rangle$$

be a sequence of merge pairs. The BPE tokenisation function is defined as repeated merging performed independently inside each word:

$$\tau^{\mathcal{M}}_{\mathrm{BPE}} \colon V^* \to (V^*)^*$$
$$s \mapsto \bigcirc\left(\left(r_{(x_{|\mathcal{M}|}, y_{|\mathcal{M}|})} \circ_2 \cdots \circ_2 r_{(x_1,y_1)} \circ_2 \tau_0\right)(s)\right).$$

Example 5.1. Let $s = \texttt{low}$ and

$$\mathcal{M} = \langle\, \langle\, \texttt{l}, \texttt{o} \,\rangle, \langle\, \texttt{lo}, \texttt{w} \,\rangle \,\rangle.$$

Then the application of $\tau^{\mathcal{M}}_{\mathrm{BPE}}$ on s can be decomposed into

$$\begin{aligned}
\tau_0(s) &= \langle\, \langle\, \texttt{l}, \texttt{o}, \texttt{w} \,\rangle \,\rangle \\
r_{(\texttt{l},\texttt{o})}(2\, \tau_0(s)) &= \langle\, \langle\, \texttt{lo}, \texttt{w} \,\rangle \,\rangle \\
r_{(\texttt{lo},\texttt{w})}(2\, r_{(\texttt{l},\texttt{o})}(2\, \tau_0(s))) &= \langle\, \langle\, \texttt{low} \,\rangle \,\rangle \\
\tau^{\mathcal{M}}_{\mathrm{BPE}}(s) = \bigcirc(r_{(\texttt{lo},\texttt{w})}(2\, r_{(\texttt{l},\texttt{o})}(2\, \tau_0(s)))) &= \langle\, \texttt{low} \,\rangle.
\end{aligned}$$

5.3.2.3 Word embeddings

To process a sequence of tokens $w_1 \ldots w_n$ computationally, a suitable numerical representation is required. This mapping from discrete symbols to continuous vectors is referred to as an *embedding* (Goldberg, 2017, Chapter 4.8).

mi

Definition 5.8 (Vocabulary indexing)**.** Let V be a vocabulary. A bijective function $\mathrm{idx}_V \colon V \to [|V|]$ is called a *vocabulary indexing*. It induces an ordering over V.

Definition 5.9 (Embedding matrix)**.** For a vocabulary V and an embedding dimensionality $d \in \mathbb{N}$, a matrix $\mathbf{E} \in \mathbb{R}^{|V| \times d}$ is called an *embedding matrix*. Given a vocabulary indexing idx_V, for any word $w \in V$, the $\mathrm{idx}_V(w)$-th row of $\mathbf{E}$ is called the *embedding of* w.

This notion can be used to define an embedding function over V:

$$\mathrm{emb}_{\mathrm{idx}_V}[\mathbf{E}](\, w \,) \coloneqq \mathbf{E}_{(\mathrm{idx}_V(w))} \tag{74}$$

In neural networks, the embedding matrix is treated as a learnable parameter of the embedding function since it is usually randomly initialised and jointly optimised with the remaining network during training (Goldberg, 2017, Chapter 4.8). Using this strategy, the embeddings gradually converge towards ideal representations of the vocabulary items for the given optimisation goal without the need for any manual specification of token features.

For `GPT-2 Small`, the embedding dimensionality d is 768 while `GPT-2 Medium` has a larger dimensionality of 1024 (Radford et al., 2019).

5.3.2.4 Positional information

Because the transfomer's attention mechanism contains neither recurrence nor convolution, it lacks an inherent mechanism to encode token order. Positional information must therefore be incorporated into the input representations by means of *positional encoding*.

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ be embeddings of the input sequence. A positional encoding mechanism augments each $\mathbf{x}_i$ with $i \in [n]$ using information about its position i. In general, positional encodings can be formalised as a function

$$\mathrm{pe} \colon \mathbb{R}^d \times \mathbb{N} \to \mathbb{R}^d. \tag{75}$$

This function can be naturally extended to sequences of input vectors $\mathbf{X} = \left(\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(n)}\right)$ where the positions correspond to the sequence indices such that,

$$\mathrm{pe}^*(\mathbf{X})_{(i)} \coloneqq \mathrm{pe}(\mathbf{X}_{(i)}, i). \tag{76}$$

Most commonly, pe is defined as an addition of input vector $\mathbf{x}$ and a position-dependent vector $\mathbf{p}_i$. Two principal approaches have been proposed in the field:

Sinusoidal encodings The original transformer employs fixed sinusoidal encodings (Vaswani et al., 2017).

Definition 5.10 (Sinusoidal encoding)**.** For embedding dimensionality $d \in \mathbb{N}$, the sinusoidal encoding function maps a pair of position and dimension $i, k \in \mathbb{N}$ with $k \leq d$ to $[-1,1]$:

$$\mathrm{se}_d(i,k) \colon \mathbb{N} \times [d] \to [-1,1]$$

$$\mathrm{se}_d(i,k) \coloneqq \begin{cases} \sin(\gamma) & \text{if } k \text{ is even,} \\ \cos(\gamma) & \text{if } k \text{ is odd,} \end{cases}$$

$$\text{where} \quad \gamma = i \cdot 10000^{-\frac{2\lfloor k/2 \rfloor}{d}}.$$

In practice, one instantiates a fixed encoding matrix $\mathbf{S} \in \mathbb{R}^l \times \mathbb{R}^d$ for a maximal sequence length l such that

$$\mathbf{S}_{(i,k)} \coloneqq \mathrm{se}_d(i,k)$$

Now, the sinusoidal positional encoding function is defined as

$$\mathrm{pe}_{\sin}(\mathbf{x}, i) \coloneqq \sqrt{d} \cdot \mathbf{x} + \mathbf{S}_{(i)} \tag{77}$$

According to Vaswani et al. (2017), the scaling factor $\sqrt{d}$ ensures that the magnitude of the vector token representation is not dominated by the positional encoding.

Positional embeddings An alternative approach replaces fixed encodings with learnable position vectors. This method has been used by `GPT-1` (Radford and Narasimhan, 2018) and `GPT-2` (Radford et al., 2019). Formally, one introduces an embedding function over the vocabulary of positions $[l]$ for some maximum sequence length l.

Definition 5.11 (Positional embedding)**.** The positional embedding encoding function is defined as

$$\mathrm{pe}_{\mathrm{emb}}[\mathbf{P}](\,\mathbf{x}, i\,) \coloneqq \mathbf{x} + \mathrm{emb}_{\mathrm{idx}_{[l]}}[\mathbf{P}](\,i\,), \tag{78}$$

where $\mathbf{x} \in \mathbf{R}^d$ is an input embedding, i is a position and $\mathbf{P} \in \mathbb{R}^{l \times d}$ is the learnable positional embedding matrix.

Here, positional representations are optimised jointly with the remaining model parameters.

Comparison The sinusoidal encodings proposed by Vaswani et al. (2017) have several conceptual advantages: they are parameter-free, one can theoretically extrapolate them to arbitrary sequence lengths not observed during training and they can represent relative position due to the periodic structure of the sine and cosine functions (Vaswani et al., 2017). On the other hand, learned positional embeddings allow for greater adaptivity to tasks and do not presuppose a form of positional information that might be suboptimal for the model.

Empirically, learned and sinusoidal encodings yield comparable performance in many settings (Vaswani et al., 2017). This might be due to the fact that positional embeddings learned as part of generative transformers appear to adopt a circular pattern quite similar in effect to sinusoidal encodings, as pointed out by Yedidia (2023a) and Yedidia (2023b)

Newer large language models such as `Llama` (Touvron et al., 2023) or `DeepSeek` (DeepSeek-AI et al., 2024) predominantly use novel approaches based on *rotary position embeddings* (RoPE) (Su et al., 2023).

`GPT-2` embedding layer Finally, the complete embedding layer used in GPT-2, embeds each token in the input, adds position encodings and applies position-wise dropout as follows.

Definition 5.12 (Embedding function)**.** Let $d \in \mathbb{N}$ be the model dimensionality and $l \in \mathbb{N}$ a maximum sequence length. The input embedding function yields a matrix in $\mathbb{R}^{n \times d_{\mathrm{model}}}$ for any input $w_1 \ldots w_n \in V^*$ and is defined as

$$\mathrm{EmbLayer}[\mathbf{E}, \mathbf{P}](\,w_1 \ldots w_n\,) \coloneqq \mathrm{Drop}_{p_{\mathrm{emb}}}(2\,\mathrm{pe}^*[\mathbf{P}](\,\mathrm{emb}_{\mathrm{idx}_V}[\mathbf{E}](\,w_1 \ldots w_n\,))),$$

$$\mathbf{E} \in \mathbb{R}^{|V| \times d},\ \mathbf{P} \in \mathbb{R}^{l \times d}.$$

5.3.2.5 Attention

In transformer models, the mechanism of *attention* is used to perform interactions between representations of different tokens. The central idea of attention is to match a *query* against a set of *keys* in order to compute compatibility scores, which are then used to form a weighted sum of *values* corresponding to the keys. Formally, an attention function maps a query and a pair of key and value collections of equal length to an output vector of the same type as the values. The weights

are obtained by applying a normalisation function to compatibility scores between the query and each key (Vaswani et al., 2017).

Given a compatibility function $\psi\colon \mathbb{R}^{d_\mathrm{k}} \times \mathbb{R}^{d_\mathrm{k}} \to \mathbb{R}$ for some dimensionality $d_\mathrm{k} \in \mathbb{N}$ and a normalisation function $\mu\colon \mathbb{R}^n \to \Delta^n$ with $n \in \mathbb{N}$ being the number of keys/queries, attention can be formalised as

$$\mathrm{Attention}_{\mu,\psi}(\mathbf{q}, \mathbf{K}, \mathbf{V}) = \mu\left(\begin{bmatrix} \psi(\mathbf{q}, \mathbf{K}_{(1)}) \\ \vdots \\ \psi(\mathbf{q}, \mathbf{K}_{(n)}) \end{bmatrix}\right)^\mathsf{T} \mathbf{V}, \tag{79}$$

Here, $\mathbf{q} \in \mathbb{R}^{d_\mathrm{k}}$ is called the *query* vector, $\mathbf{K} \in \mathbb{R}^{n \times d_\mathrm{k}}, \mathbf{V} \in \mathbb{R}^{n \times d_\mathrm{v}}$ are matrices of *keys* and *values* respectively, d_k is the key/query dimensionality and d_v is the value dimensionality which determines the output shape. Thus, the $\mathrm{Attention}_{\mu,\psi}$ function computes a weighted sum of the vectors in $\mathbf{V}$ based on the probability simplex created from the compatibility scores between $\mathbf{q}$ and the items in $\mathbf{K}$.

In self-attentive settings where compatibility is computed between a query token from a sequence and the items preceding it in that sequence, the token at position i must not attend to positions $j > i$. This is achieved via masked attention (Vaswani et al., 2017). Formally, for position $i \leq n$ of query $\mathbf{q}$,

$$\begin{aligned} \mathrm{MaskedAtt}^i_{\mu,\psi}(\mathbf{q}, \mathbf{K}, \mathbf{V}) = \mu\left(\begin{bmatrix} \psi(\mathbf{q}, \mathbf{K}_{(1)}) \\ \vdots \\ \psi(\mathbf{q}, \mathbf{K}_{(i)}) \end{bmatrix}\right)^\mathsf{T} \mathbf{V}_{(1:i)} &= \left[\begin{array}{c} \mu\left(\begin{bmatrix} \psi(\mathbf{q}, \mathbf{K}_{(1)}) \\ \vdots \\ \psi(\mathbf{q}, \mathbf{K}_{(i)}) \end{bmatrix}\right) \\ \hline 0 \\ \vdots \\ 0 \end{array}\right]^\mathsf{T} \mathbf{V} \end{aligned} \tag{80}$$

If the normalisation function μ admits a null-element 0_μ, masking can be equivalently implemented by defining a masking function:

$$\begin{aligned} \mathrm{m}_i &\colon \mathbb{R}^n \to \mathbb{R}^n \\ \mathrm{m}_i(\mathbf{x})_{(j)} &\coloneqq \begin{cases} \mathbf{x}_{(j)} & \text{if } j \leq i \\ 0_\mu & \text{else} \end{cases}. \end{aligned} \tag{81}$$

Then, Equation 80 can be reformulated as

$$\mathrm{MaskedAtt}^i_{\mu,\psi}(\mathbf{q}, \mathbf{K}, \mathbf{V}) = \mu \circ \mathrm{m}_i\left(\begin{bmatrix} \psi(\mathbf{q}, \mathbf{K}_{(1)}) \\ \vdots \\ \psi(\mathbf{q}, \mathbf{K}_{(n)}) \end{bmatrix}\right)^\mathsf{T} \mathbf{V}. \tag{82}$$

It is common to apply a dropout layer to the normalised compatibility scores during training to randomly mask the contribution of some of the context items (Vaswani et al., 2017). This is known

as *attention dropout*.

$$\text{MaskedAtt}^{i}_{\mu,\psi}(\mathbf{q},\mathbf{K},\mathbf{V}) = \text{Drop}_{p_{\text{att}}}\left(\mu \circ \text{m}_i\left(\begin{bmatrix}\psi(\mathbf{q},\mathbf{K}_{(1)})\\ \vdots \\ \psi(\mathbf{q},\mathbf{K}_{(n)})\end{bmatrix}^{\mathsf{T}}\right)\right)\mathbf{V}. \tag{83}$$

Dot-product Attention functions are designed to quantify the compatibility between two vector representations of the same dimensionality. They were initially introduced by Bahdanau et al. (2016) in the field of neural machine translation and later used in parsing mechanisms (Dozat and Manning, 2017), before being included in the transformer architecture. Two widely used compatibility functions are additive attention (Bahdanau et al., 2016) and multiplicative (dot-product) attention (Vaswani et al., 2017).

Additive attention employs a small feed-forward network to compute the compatibility between two vectors, whereas dot-product attention is parameter-free and more efficient to implement. For these reasons, the transformer uses dot-product attention (Vaswani et al., 2017).

The *dot product/inner product* between two vectors $\mathbf{q},\mathbf{k} \in \mathbb{R}^{d_\text{k}}$ is defined as

$$\text{dp}(\mathbf{q},\mathbf{k}) \coloneqq \sum_{i=1}^{d} \mathbf{q}_{(i)}\mathbf{k}_{(i)}. \tag{84}$$

One can also write it as a matrix product: $\mathbf{q}^{\mathsf{T}}\mathbf{k}$. It maps two d_k-dimensional vectors to a scalar in $\mathbb{R}$. Its magnitude depends on both the lengths of the vectors and the angle between them. More concretely: $\text{dp}(\mathbf{q},\mathbf{k}) = \|\mathbf{q}\|_2 \cdot \|\mathbf{k}\|_2 \cdot \cos\theta$, where θ denotes the angle between $\mathbf{q}$ and $\mathbf{k}$. It follows that the dot-product is larger than zero if the angle is smaller than $90°$, equal to zero, if the vectors are orthogonal and smaller than zero if $\theta > 90°$. Thus, the dot product measures directional alignment: it is large when the vectors point in similar directions, small (or negative) when they are dissimilar. If both vectors are have length 1, the dot product reduces to cosine similarity.

In the context of attention, queries and keys are learned representations in a shared embedding space. The dot product therefore can be understood to quantify how strongly a query points toward a key. Large positive values indicate high compatibility and lead – after normalisation – to large attention weights.

The dot product is particularly suitable as a compatibility function for two reasons: Firstly, it is smooth, making gradient-based optimisation straightforward. Secondly, for a sequence of queries $\mathbf{Q} \in \mathbb{R}^{n_1 \times d_\text{k}}$ and a key matrix $\mathbf{K} \in \mathbb{R}^{n_2 \times d_\text{k}}$ with lengths $n_1, n_2 \in \mathbb{N}$, all pairwise compatibilities can be computed simultaneously as a single matrix multiplication $\mathbf{Q}\mathbf{K}^{\top}$, which is highly optimised on modern hardware (Vaswani et al., 2017).

Scaled attention Since the dot product assigns values in $\mathbb{R}$, including negative numbers, the transformer uses softmax for normalisation. However, when the dimensionality d_k is large and the vector components have non-trivial variance, the magnitude of $\mathbf{q}^{\mathsf{T}}\mathbf{K}_{(i)}$ grows proportionally to d_k, possibly pushing the softmax function into regions with very small gradients with respect to the loss. This motivates the introduction of a scaling factor $1/\sqrt{d}$, which stabilises gradients by keeping the variance of the scores approximately constant (Vaswani et al., 2017).

Definition 5.13 (Scaled dot-product)**.** The *scaled dot product* between two vectors $\mathbf{q}, \mathbf{k} \in \mathbb{R}^{d_\mathrm{k}}$ is defined as

$$\mathrm{sdp}(\mathbf{q}, \mathbf{k}) \coloneqq \frac{\mathrm{dp}(\mathbf{q}, \mathbf{k})}{\sqrt{d_\mathrm{k}}}.$$

In practice, dot-product attention is computed simultaneously on a sequence of queries contained in a query matrix $\mathbf{Q} \in \mathbb{R}^{n_1 \times d_\mathrm{k}}$ and it will be helpful for the following definitions to operate on such a sequence of input vectors.

Definition 5.14 (Scaled dot-product attention)**.** For query sequence length n_1 and key/value sequence length n_2, scaled dot-product attention is a function

$$\mathrm{Attention} \colon \mathbb{R}^{n_1 \times d_\mathrm{k}} \times \mathbb{R}^{n_2 \times d_\mathrm{k}} \times \mathbb{R}^{n_2 \times d_\mathrm{v}} \to \mathbb{R}^{n_1 \times d_\mathrm{v}}$$

defined by

$$\begin{aligned}\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \coloneqq &\begin{pmatrix} \mathrm{MaskedAtt}^1_{\mathrm{softmax,sdp}}(\mathbf{Q}_{(1)}, \mathbf{K}, \mathbf{V}) \\ \vdots \\ \mathrm{MaskedAtt}^{n_1}_{\mathrm{softmax,sdp}}(\mathbf{Q}_{(n_1)}, \mathbf{K}, \mathbf{V}) \end{pmatrix} \\ &= \mathrm{Drop}_{p_\mathrm{att}}\left(\mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\mathsf{T}}{\sqrt{d_\mathrm{k}}} + \mathbf{M}\right)\right)\mathbf{V},\end{aligned}$$

where $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ is a mask matrix with

$$\mathbf{M}_{(i,j)} = \begin{cases} 0 & \text{if } j \leq i \\ -\infty & \text{if } j > i \end{cases}$$

in the self-attentive case and $\mathbf{M} = \mathbf{0}$ else. For self-attentive training, queries and keys/values have the same length, $n_1 = n_2$.

The mask matrix contains $-\infty$ entries at for key/value positions to mask due to it being the null-element of the $\mathrm{softmax}$ function (see Section 2.2). For convenience it assumed that $x - \infty = -\infty$ for any $x \in \mathbb{R}$.

Parameterised attention So far, attention has been defined directly on the sequence representations $\mathbf{Q}, \mathbf{K}, \mathbf{V}$. However, the transformer architecture does not maintain three separate embeddings for each vocabulary item. Instead, input sequence representations are transformed into query, key and value spaces by a learned affine map (Vaswani et al., 2017). This introduces additional flexibility and allows the model to learn distinct similarity and aggregation structures.

Formally, an affine transformation f of a representation $\mathbf{x} \in \mathbb{R}^{d_1}$ into the space $\mathbb{R}^{d_2}$ with $d_1, d_2 \in \mathbb{N}$ can be written as

$$f(\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathbf{W} + \mathbf{b}, \tag{85}$$

with $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$ being called the *weight* matrix and $\mathbf{b} \in \mathbb{R}^{d_2}$ the *bias* term (Goldberg, 2017, Chapter 2.7).

The formula can be reduced to a single matrix multiplication by subsuming the bias term into the weight matrix. For this, it is necessary to augment the input representation with a constant feature. For any dimensionality $d \in \mathbb{N}$ define the $_{/\mathbf{1}}$ operator as follows,

$$
\begin{aligned}
{}_{/\mathbf{1}} &\colon \mathbb{R}^d \to \mathbb{R}^{d+1} \\
\mathbf{x}_{/\mathbf{1}} &\coloneqq \begin{pmatrix} \mathbf{x}_{(1)} \\ \vdots \\ \mathbf{x}_{(d)} \\ 1 \end{pmatrix}.
\end{aligned}
\tag{86}
$$

Now, the affine transformation f can be expressed as:

$$
f(\mathbf{x}) = \mathbf{x}_{/\mathbf{1}}^{\mathsf{T}} \tilde{\mathbf{W}}, \tag{87}
$$

with $\tilde{\mathbf{W}}$ being decomposable as

$$
\tilde{\mathbf{W}} = \left(\begin{array}{c} \mathbf{W} \\ \hline \mathbf{b}^{\mathsf{T}} \end{array} \right). \tag{88}
$$

Finally, the transformation can be applied to multiple input representations $\mathbf{X} \in \mathbb{R}^{n \times d_1}$ using a single matrix multiplication by extending the augmentation to matrices: $\mathbf{X}_{/\mathbf{1}} \coloneqq \mathbf{X} \circ_2 \mathbf{1} \in \mathbb{R}^{n \times (d_1+1)}$, where $\mathbf{1} \in \mathbb{R}^{n \times 1}$ is a 1-column matrix of ones. Then, parallel application can be achieved by

$$
f({}_2\,\mathbf{X}) = \mathbf{X}_{/\mathbf{1}} \tilde{\mathbf{W}},
$$

which corresponds to a row-wise broadcasted bias addition when substituting in the decomposition in Equation 88:

$$
\mathbf{X}_{/\mathbf{1}} \tilde{\mathbf{W}} = \left(\begin{array}{c|c} \mathbf{X} & \mathbf{1} \end{array} \right) \left(\begin{array}{c} \mathbf{W} \\ \hline \mathbf{b}^{\mathsf{T}} \end{array} \right) = \mathbf{X}\mathbf{W} + \mathbf{1}\mathbf{b}^{\mathsf{T}}.
$$

Now, in order to define general parameterised attention, let $\mathbf{X} \in \mathbb{R}^{n_1 \times d}$, $\mathbf{Y} \in \mathbb{R}^{n_2 \times d}$ denote two sequences of input representations. Queries, keys and values are obtained via affine maps (Vaswani et al., 2017):

$$
\begin{gathered}
\mathbf{Q} = \mathbf{X}_{/\mathbf{1}} \tilde{\mathbf{W}}^{\mathrm{Q}}, \quad \mathbf{K} = \mathbf{Y}_{/\mathbf{1}} \tilde{\mathbf{W}}^{\mathrm{K}}, \quad \mathbf{V} = \mathbf{Y}_{/\mathbf{1}} \tilde{\mathbf{W}}^{\mathrm{V}}, \\
\tilde{\mathbf{W}}^{\mathrm{Q}}, \tilde{\mathbf{W}}^{\mathrm{K}} \in \mathbb{R}^{(d+1) \times d_k}, \quad \tilde{\mathbf{W}}^{\mathrm{V}} \in \mathbb{R}^{(d+1) \times d_{\mathrm{v}}}.
\end{gathered}
\tag{89}
$$

Thus, the items encoded in $\mathbf{X}$ query into the sequence $\mathbf{Y}$. The matrices $\tilde{\mathbf{W}}^{\mathrm{Q}}, \tilde{\mathbf{W}}^{\mathrm{K}}, \tilde{\mathbf{W}}^{\mathrm{V}}$ are optimised during model training. While it is common practice in expositions of the transformer architecture to omit specifying the optional bias terms, simply using $\mathbf{X}^{\mathrm{Q}}, \mathbf{W}^{\mathrm{K}}, \mathbf{W}^{\mathrm{V}}$, the following definitions will include them for reasons of clarity.

Now, using the parameterised matrices, the fully parameterised attention mechanism as employed in the architecture of (Vaswani et al., 2017) is defined as.

Definition 5.15 (Parameterised scaled dot-product attention)**.** Parameterised scaled dot-product attention is defined as

$$
\begin{gathered}
\mathrm{ParamAttention}[\tilde{\mathbf{W}}^{\mathrm{Q}}, \tilde{\mathbf{W}}^{\mathrm{K}}, \tilde{\mathbf{W}}^{\mathrm{V}}](\mathbf{X}, \mathbf{Y}) \coloneqq \mathrm{Attention}(\mathbf{X}_{/\mathbf{1}} \tilde{\mathbf{W}}^{\mathrm{Q}}, \mathbf{Y}_{/\mathbf{1}} \tilde{\mathbf{W}}^{\mathrm{K}}, \mathbf{Y}_{/\mathbf{1}} \tilde{\mathbf{W}}^{\mathrm{V}}), \\
\mathbf{X} \in \mathbb{R}^{n_1 \times d}, \mathbf{Y} \in \mathbb{R}^{n_2 \times d}, \tilde{\mathbf{W}}^{\mathrm{Q}}, \tilde{\mathbf{W}}^{\mathrm{K}} \in \mathbb{R}^{(d+1) \times d_{\mathrm{k}}}, \tilde{\mathbf{W}}^{\mathrm{V}} \in \mathbb{R}^{(d+1) \times d_{\mathrm{v}}}.
\end{gathered}
$$

Self-attention is a special case of attention in which queries, keys and values are obtained from the same sequence representation $\mathbf{X}$. Here, each token representation issues a query over the tokens in its context – which is precicely what is used for language modelling purposes (Radford and Narasimhan, 2018).

Definition 5.16 (Parameterised scaled dot-product self-attention)**.**

$$\text{ParamSelfAttention}[\tilde{\mathbf{W}}^{\text{Q}}, \tilde{\mathbf{W}}^{\text{K}}, \tilde{\mathbf{W}}^{\text{V}}](\,\mathbf{X}\,) \coloneqq \text{ParamAttention}[\tilde{\mathbf{W}}^{\text{Q}}, \tilde{\mathbf{W}}^{\text{K}}, \tilde{\mathbf{W}}^{\text{V}}](\,\mathbf{X}, \mathbf{X}\,)$$
$$\mathbf{X} \in \mathbb{R}^{n \times d},\ \tilde{\mathbf{W}}^{\text{Q}},\ \tilde{\mathbf{W}}^{\text{K}} \in \mathbb{R}^{(d+1) \times d_{\text{k}}},\ \tilde{\mathbf{W}}^{\text{V}} \in \mathbb{R}^{(d+1) \times d_{\text{v}}}.$$

The reason for using projections $\tilde{\mathbf{W}}^{\text{Q}}$ and $\tilde{\mathbf{W}}^{\text{K}}$ instead of computing dot products directly in the common input representation space is that they allow to distinguish the functional roles of queries and keys, implementing a form of feature selection dependent on the item's role. In self-attention, without separate transformations, the compatibility between any two items $\mathbf{x}$ and $\mathbf{y}$ would be the same, irrespective of their order, due to the commutativity of the dot-product. The value transformation $\tilde{\mathbf{W}}^{\text{V}}$ allows the model to suppress features irrelevant for the subsequent layer(s) and to map the result to a different output dimensionality d_{v}.

5.3.2.6 Multi-Head Attention

A single attention operation computes one weighted aggregation of values based on a single learned similarity space. However, different linguistic or structural relations may require distinct similarity notions (e.g. agreement, coreference, syntactic dependency). The transformer therefore employs *multi-head attention* (MHA), which runs several attention mechanisms in parallel, each operating in a lower-dimensional subspace compared to embedding dimensionality, and subsequently combines their outputs (Vaswani et al., 2017). Intuitively, each head learns its own projections into query, key and value spaces and can therefore specialise to a different compatibility/similarity structure (Tsai et al., 2019).

Let $m \in \mathbb{N}$ denote the number of attention heads. Now, for multi-head attention, define m separate sets of $\tilde{\mathbf{W}}^{\text{Q}}, \tilde{\mathbf{W}}^{\text{K}}, \tilde{\mathbf{W}}^{\text{V}}$ transformations. This can be notated using three-dimensional tensors with the first dimension having size m.

Definition 5.17 (Parameterised multi-head self-attention)**.**

$$\text{ParamSelfMHA}[\tilde{\mathbf{W}}^{\text{Q}}, \tilde{\mathbf{W}}^{\text{K}}, \tilde{\mathbf{W}}^{\text{V}}, \tilde{\mathbf{W}}^{\text{O}}](\,\mathbf{X}\,) \coloneqq \text{Drop}_{p_{\text{resid}}}(\bigcirc(\langle\,\text{head}_1, \ldots, \text{head}_m\,\rangle)_{/\mathbf{1}}\,\tilde{\mathbf{W}}^{\text{O}}),$$
$$\text{where}\quad \text{head}_h = \text{ParamSelfAttention}[\tilde{\mathbf{W}}^{\text{Q}}_{(h)}, \tilde{\mathbf{W}}^{\text{K}}_{(h)}, \tilde{\mathbf{W}}^{\text{V}}_{(h)}](\,\mathbf{X}\,) \qquad \forall\, h \in [m],$$
$$\mathbf{X} \in \mathbb{R}^{n \times d},\ \tilde{\mathbf{W}}^{\text{Q}}, \tilde{\mathbf{W}}^{\text{K}} \in \mathbb{R}^{m \times (d+1) \times d_{\text{k}}},\ \tilde{\mathbf{W}}^{\text{V}} \in \mathbb{R}^{m \times (d+1) \times d_{\text{v}}},$$
$$\tilde{\mathbf{W}}^{\text{O}} \in \mathbb{R}^{(m \cdot d_{\text{v}}+1) \times d},\ \text{head}_h \in \mathbb{R}^{n \times d_{\text{v}}}.$$

The outputs of each attention mechanism are concatenated and the result is projected back to the model space $\mathbb{R}^d$ using the output projection $\tilde{\mathbf{W}}^{\text{O}}$. Usually, it holds that

$$d_{\text{k}} = d_{\text{v}} = \frac{d}{m}, \tag{90}$$

so that the attention heads operate in lower dimensionality (Vaswani et al., 2017). This is also the case for `GPT-2` (Radford et al., 2019). Furthermore, `GPT-2 small` and `GPT-2 medium`

employ MHA layers with 12 attention heads each (Radford et al., 2019; Radford and Narasimhan, 2018).

Efficient implementations do not compute each head independently but instead they first generate a sequence of combined query-key-value vectors,

$$\mathbf{Z} = \mathbf{X}_{/\mathbf{1}} \tilde{\mathbf{W}}^{\mathrm{QKV}}, \tag{91}$$

with $\tilde{\mathbf{W}}^{\mathrm{QKV}} \in \mathbb{R}^{(d+1)\times(3\cdot m\cdot d_\mathrm{k})}$, and then split along the second axis into separe query, key and value matrices

$$\mathbf{Z} = \left(\begin{array}{c|c|c} \mathbf{Q} & \mathbf{K} & \mathbf{V} \end{array} \right) \tag{92}$$

where each block lies in $\mathbb{R}^{n\times m\cdot d_k}$. These are then reshaped into tensors of shape $m \times n \times d_\mathrm{k}$. Then, the head-wise matrix multiplication between queries and keys can be performed using a single batched multiplication over the first axis (Wolf et al., 2020).

This computational optimisation allows for a single large matrix multiplication leading to better *graphics processing unit* (GPU) utilisation and fewer memory reads. For notational brevity, I will use the single composed matrix $\tilde{\mathbf{W}}^{\mathrm{QKV}}$ when specifying parameters of higher-level transformer components below.

Lastly, transformer implementations are usually trained with batched input (e.g. Radford and Narasimhan, 2018), i.e. multiple input sequences at once corresponding to tensors of shape $\mathbf{X} \in \mathbb{R}^{b\times n\times d}$ with b being the number of sequences in the batch. This allows to take even stronger advantage of GPU acceleration for matrix operations. The definitions in this work can be prepended with a batch axis to make them compatible with this view.

Low-rank structure in (multi-head) attention It is worth noting that the parameterised dot-product for input sequence representations $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ can be rewritten in biaffine form:

$$\psi(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}_{/\mathbf{1}}^\mathsf{T} \overbrace{\tilde{\mathbf{W}}^{\mathrm{Q}}(\tilde{\mathbf{W}}^{\mathrm{K}})^\mathsf{T}}^{\mathbf{S}} \mathbf{y}_{/\mathbf{1}}}{\sqrt{d_\mathrm{k}}} \tag{93}$$

where $\mathbf{S} \in \mathbb{R}^{d+1\times d+1}$. Thus, each attention head learns a separate suitable biaffine compatibility function which is strictly more expressive than a simple dot product (Elhage et al., 2021) and similar to the biaffine form used by Dozat and Manning (2017).

Given that one can express this transformation using a single matrix, it might seem peculiar that the implementation employs two separate transformations. However, note that the inner dimension of the transformation $\mathbf{S} = \tilde{\mathbf{W}}^{\mathrm{Q}}(\tilde{\mathbf{W}}^{\mathrm{K}})^\mathsf{T}$ is $d_\mathrm{k} \leq d$, so that $\mathrm{rank}(\mathbf{S}) \leq d_\mathrm{k}$. Hence, each head realises a *low-rank* biaffine form in the model space (Elhage et al., 2021). If one were to employ a single learnable $(d+1)\times(d+1)$ transformation matrix, the rank contraint would get lost.

Similarly, following Definition 5.17, the value-transformation in head $h \in [m]$ followed by the output projection $\mathbf{W}^{\mathrm{O}}$ can be rewritten as

$$\mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\mathsf{T}}{\sqrt{d_\mathrm{k}}}\right) \mathbf{X}_{/\mathbf{1}} \underbrace{\tilde{\mathbf{W}}^{V}_{(h)} \mathbf{W}^{O}_{(((h-1)d_v+1):h\cdot d_v)}}_{\mathbf{T}} \tag{94}$$

where $\mathbf{W}^{\mathrm{O}}_{(((h-1)d_{\mathrm{v}}+1):h\cdot d_{\mathrm{v}})}$ denotes the slice of $\mathbf{W}^{O}$ corresponding to head h (Cordonnier et al., 2019; Kobayashi et al., 2020). Note that the output bias term is omitted here. Again, the composite matrix $\mathbf{T}$ has rank of at most $d_{\mathrm{v}} \leq d$.

This architectural choice has two consequences. First, for MHA with more than 2 heads, the number of parameters is reduced compared to learning arbitrary full-rank transformations. Second, the low-rank structure acts as an inductive bias and potentially as regularisation.

5.3.2.7 Feed-forward network

In addition to multi-head attention, responsible for the interaction between different tokens in the input, transformers employ *feed-forward networks* that are applied independently to the representations of each input position. These networks combine affine transformations (such as $\tilde{\mathbf{W}}^{\mathrm{O}}$ above) and nonlinear *activation functions* to perform computations on the outputs of the MHA layer (Vaswani et al., 2017).

A feed-forward network or *multi-layer perceptron* (Goldberg, 2017, Chapter 4.2) in its most general form is defined as an alternating composition of affine tranformations and non-linear activations, closed with a final affined transformation, where the affine weights are trainable parameters.

Definition 5.18 (Feed-forward network)**.** A feed-forward network with $n \in \mathbb{N}$ layers and activation function $\chi\colon \mathbb{R} \to \mathbb{R}$ is a function

$$\begin{aligned} \mathrm{FF}_{\chi}[\tilde{\boldsymbol{\upsilon}}](\,\mathbf{x}\,) &\coloneqq \left(g \circ \bigcirc_{i=1}^{n-1} f_{\chi}^{i}\right)(\mathbf{x}) \\ \text{with} \quad f_{\chi}^{i}(\mathbf{a}) &= \chi(\mathbf{a}_{/\mathbf{1}}\tilde{\boldsymbol{\upsilon}}_{(i)}) \qquad \forall\, i \in [n-1], \\ g(\mathbf{a}) &= \mathbf{a}_{/\mathbf{1}}\tilde{\boldsymbol{\upsilon}}_{(n)}, \\ \mathbf{x} \in \mathbb{R}^{d_1},\ \tilde{\upsilon} \in \bigtimes_{i=1}^{n} \mathbb{R}^{(d_i+1)\times d_{i+1}},\ &\langle d_1, ..., d_{n+1}\rangle \in \mathbb{N}^{n+1}. \end{aligned}$$

In transformer architectures, the feed-forward network is applied independently to each row of a sequence representation $\mathbf{X} \in \mathbb{R}^{n\times d}$ with n being the length of the sequence. That is,

$$\mathrm{FF}_{\chi}[\tilde{\omega}](_2\,\mathbf{X}\,) = \begin{pmatrix} \mathrm{FF}_{\xi}(\mathbf{X}_{(i)}) \\ \vdots \\ \mathrm{FF}_{\xi}(\mathbf{X}_{(n)}) \end{pmatrix}.$$

In the original transformer, this network consists of two linear layers with an intermediate ReLU activation (see Definition 5.20), followed by dropout with probability p_{ff} (Vaswani et al., 2017).

Definition 5.19 (Transformer feed-forward network)**.**

$$\begin{aligned} \mathrm{FF}^{\mathrm{Transformer}}[\langle\, \tilde{\mathbf{W}}_1, \tilde{\mathbf{W}}_2\,\rangle](\,\mathbf{x}\,) &\coloneqq \mathrm{Drop}_{p_{\mathrm{ff}}}(\mathrm{FF}_{\mathrm{ReLU}}[\langle\, \tilde{\mathbf{W}}_1, \tilde{\mathbf{W}}_2\,\rangle](\,\mathbf{x}\,)), \\ \tilde{\mathbf{W}}_1 \in \mathbb{R}^{(d+1)\times d_{\mathrm{ff}}},\ &\tilde{\mathbf{W}}_2 \in \mathbb{R}^{(d_{\mathrm{ff}}+1)\times d}. \end{aligned} \tag{95}$$

Typically, $d_{\mathrm{ff}} > d$, so that the network expands the representation into a higher-dimensional space before applying the non-linearity. Vaswani et al. (2017) fix $d_{\mathrm{ff}} = f_{\mathrm{ff}} \cdot d$ with factor $f_{\mathrm{ff}} = 4$ in their original proposal which was later adopted by `GPT-1` and `GPT-2` (Radford and Narasimhan, 2018; Radford et al., 2019). This expansion increases expressive capacity while preserving the overall model dimensionality.

Activation functions Activation functions introduce nonlinearities into neural networks. This is crucial because compositions of affine transformations are themselves affine. Hence, a network without nonlinearities can only represent functions of the form

$$f(x) = \mathbf{W}x + \mathbf{b}, \tag{96}$$

regardless of its depth (possibly with low-rank weight $\mathbf{W}$). Such models are therefore severely limited in their expressive power (Huchette et al., 2025). Nonlinear activation functions enable neural networks to approximate a wider class of functions. In particular, feed-forward networks with at least one hidden layer and a suitable non-linearity are *universal function approximators* (Hornik et al., 1989).

An activation function is a function on scalars of the form $\mathbb{R} \to \mathbb{R}$. Traditionally, the sigmoid function and the hyperbolic tangent were widely used as activation functions. However, these functions suffer from *gradient saturation*: for large positive or negative inputs, their derivatives approach zero, which leads to vanishing gradients during backpropagation and makes optimisation difficult in deep networks (Glorot and Bengio, 2010; Goodfellow et al., 2016). For this reason, they have largely been replaced by partly linear activation functions such as the *Rectified Linear Unit* (ReLU), which exhibit more favourable optimisation properties (Goldberg, 2017, Chapter 4.4).

Definition 5.20 (ReLU)**.** The ReLU is defined as

$$\begin{aligned} \mathrm{ReLU}\colon \mathbb{R} &\to \{\, y \in \mathbb{R} \mid y \geq 0 \,\} \\ x &\mapsto \max(\{\, 0, x \,\}). \end{aligned}$$

The linearity of ReLU above 0 is a major advantage with respect to gradient estimation. However, while ReLU is computationally efficient, it is not differentiable at $x = 0$ and yields zero gradients for negative inputs, which can lead to so-called *dead parameters* that never activate (Horuz et al., 2025, Chapter 6.3). Therefore, `GPT-2` uses a related activation function called *Gaussian Error Linear Unit* (GELU) (Hendrycks and Gimpel, 2016).

Definition 5.21 (Gaussian Error Linear Unit (GELU))**.** The GELU is defined as

$$\begin{aligned} \mathrm{GELU}\colon \mathbb{R} &\to \mathbb{R} \\ x &\mapsto x \cdot \Phi(x), \end{aligned}$$

where $\Phi(x)$ denotes the cumulative distribution function of the standard normal distribution,

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{t^2}{2}\right) dt.$$

GELU is a smooth approximation of the ReLU function that is differentiable everywhere. This smoothness can lead improved empirical performance in large transformer models (Hendrycks and Gimpel, 2016). For computational efficiency, GELU is often implemented using an approximation (Hendrycks and Gimpel, 2016).

5.3.2.8 Layer normalisation

In addition to multi-head attention and a feed-forward network, a transformer layer consists of two components: *layer normalisation* and *residual connections* (see Section 5.3.2.9). Layer normalisation (Ba et al., 2016) standardises the values of a representation vector along its dimensions by

centering and scaling. In contrast to batch normalisation, which computes statistics across a batch of examples, layer normalisation operates independently on each individual input vector. This makes it particularly suitable for sequence models where batch statistics may vary across time steps due to different sequence lengths (Ba et al., 2016).

Definition 5.22 (Layer Normalisation)**.** Let $\mathbf{x} \in \mathbb{R}^d$ be an input representation. Define the mean and standard deviation across its feature dimensions as

$$\operatorname{avg}(\mathbf{x}) \coloneqq \frac{1}{d}\sum_{i=1}^{d}\mathbf{x}_{(i)}, \qquad \operatorname{std}(\mathbf{x}) \coloneqq \sqrt{\frac{1}{d}\sum_{i=1}^{d}(\mathbf{x}_{(i)} - \operatorname{avg}(\mathbf{x}))^2}.$$

Layer normalisation is then defined as

$$\operatorname{LayerNorm}[\gamma, \beta](\mathbf{x}) \coloneqq \gamma \odot \frac{\mathbf{x} - \operatorname{avg}(\mathbf{x})}{\operatorname{std}(\mathbf{x})} + \beta,$$

where $\gamma, \beta \in \mathbb{R}^d$ are learned parameters.

The normalisation step ensures that the transformed vector $(\mathbf{x} - \operatorname{avg}(\mathbf{x}))/\operatorname{std}(\mathbf{x})$ has empirical mean 0 and variance 1 across its coordinates. The parameters γ and β implement a separate affine transformation for each dimension that allows for adaptive rescaling and shifting intended to prevent the layer normalisation from compromising the expressive capacity of the network (Ba et al., 2016). From an optimisation perspective, the empricial success of layer normalisation is thought to originate from a stabilisation of the magnitude of the vector representations across layers (Ba et al., 2016), from re-centering and scaling gradients to prevent gradient explosions (Xu et al., 2019) and from the introduction of a strong non-linearity (Ni et al., 2024). This empirically enables stable training of transformer models (Zhang and Sennrich, 2019).

5.3.2.9 Residual connections

So-called *residual*, *skip* or *shortcut connections* introduce additive identity mappings between layers. They were originally proposed by He et al. (2016) to facilitate the training of very deep neural networks.

A central difficulty in deep architectures is that gradients propagated through many nonlinear layers may either decay or grow exponentially, leading to vanishing or exploding gradients (Bengio et al., 1994; Kolen and Kremer, 2001; Glorot and Bengio, 2010). Residual connections mitigate this problem by providing direct gradient paths that bypass intermediate transformations.

Definition 5.23 (Residual connection)**.** The residual connection can be defined as a family of operators on functions. Given a domain X, an addition operation $+_X$ defined for the elements of X and the identity function Id_X, a residual connection operator is defined as

$$\begin{aligned} \mathcal{R}_X &\colon X^X \to X^X \\ f &\mapsto \mathrm{Id}_X +_X f. \end{aligned}$$

Residual connections reparameterise the target function g^* approximated by the network as

$$f^*(\mathbf{x}) = \mathbf{x} + f(\mathbf{x}), \tag{97}$$

so that it learns the residual error $f(\mathbf{x}) = f^*(\mathbf{x}) - \mathbf{x}$ instead of f^* itself.

Residual connections also simplify the learning of identity mappings. If the optimal transformation is close to the identity, it suffices to learn $f(\mathbf{x}) \approx \mathbf{0}$, which is typically easier than learning $\mathbf{x} \mapsto \mathbf{x}$ through a stack of nonlinear layers (He et al., 2016). This observation explains why residual architectures can be trained reliably even when they are overparameterised in depth. In a transformer layer, multi-head attention and the feed-forward network are embedded in residual connections (Vaswani et al., 2017), made precise in the following section.

5.3.2.10 Transformer layer

Vaswani et al. (2017) combine a multi-head attention mechanism and a feed-forward network into a *transformer layer*. In this context, the attention mechanism and the feed-forward network are called *sublayers*.

Both sublayers are embedded in residual connections. Furthermore, Vaswani et al. (2017) employ layer normalisation on their respective outputs (*post-normalisation*). In contrast, `GPT-2` adopts a *pre-normalisation* (Radford et al., 2019) which improves gradient flow and stabilises training for very deep networks (Xiong et al., 2020): The inputs of both sublayers are normalised and the residual connection is wrapped around the respective composition of LayerNorm and sublayer.

Definition 5.24 (`GPT-2` transformer layer)**.** A `GPT-2` transformer layer is defined as a parameterised function:[20]

$$\begin{aligned}
&\mathrm{Layer}[\tilde{\mathbf{W}}^{\mathrm{QKV}}, \tilde{\mathbf{W}}^{\mathrm{O}}, \tilde{\mathbf{W}}_1, \tilde{\mathbf{W}}_2, \boldsymbol{\gamma}, \boldsymbol{\beta}](\mathbf{X}) = (\mathcal{R}(f_2) \circ \mathcal{R}(f_1))(\mathbf{X}),\\
\text{with}\quad &f_1(\mathbf{X}) = \mathrm{ParamSelfMHA}[\tilde{\mathbf{W}}^{\mathrm{QKV}}, \tilde{\mathbf{W}}^{\mathrm{O}}](\mathrm{LayerNorm}[\boldsymbol{\gamma}_{(1)}, \boldsymbol{\beta}_{(1)}](\mathbf{X})),\\
&f_2(\mathbf{X}) = \mathrm{FF}^{\mathrm{Transformer}}[\tilde{\mathbf{W}}_1, \tilde{\mathbf{W}}_2](\mathrm{LayerNorm}[\boldsymbol{\gamma}_{(2)}, \boldsymbol{\beta}_{(2)}](\mathbf{X})),\\
&\mathbf{X} \in \mathbb{R}^{n \times d},\ \tilde{\mathbf{W}}_1 \in \mathbb{R}^{(d+1) \times d_{\mathrm{ff}}},\ \tilde{\mathbf{W}}_2 \in \mathbb{R}^{(d_{\mathrm{ff}}+1) \times d},\\
&\tilde{\mathbf{W}}^{\mathrm{QKV}} \in \mathbb{R}^{(d+1) \times 3d},\ \tilde{\mathbf{W}}^{\mathrm{O}} \in \mathbb{R}^{(d+1) \times d},\ \boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^{2 \times d}.
\end{aligned}$$

In `GPT-2 small`, 12 layers such layers are stacked together in sequential order, with the output of one layer being used as the input of the next one. `GPT-2 medium` doubles the number of layers to 24 (Radford et al., 2019).

5.3.2.11 Language modelling head

In order to generate a probability distribution over the vocabulary V, it is necessary to transform the output of the transformer into $|V|$-dimensional space. Naturally, this can be done using another linear transformation of shape $\mathbf{W}^{\mathrm{M}} \in \mathbb{R}^{d \times |V|}$. Given a vector representation $\mathbf{x}$, one can rewrite the transformation as a column-wise dot-product over the matrix:

$$\mathbf{x}^{\mathsf{T}}\mathbf{W}^{\mathrm{M}} = \begin{pmatrix} \mathrm{dp}(\mathbf{x}, \mathbf{W}^{\mathrm{M}}_{(1)}) \\ \vdots \\ \mathrm{dp}(\mathbf{x}, \mathbf{W}^{\mathrm{M}}_{(|V|)}) \end{pmatrix} \tag{98}$$

[20] Note that the learnable parameters of the layer normalisation functions are generally not implemented as a single weight matrix across modules. This combined notation is used for reasons of brevity here and in the following two sections.

Note that $\mathbf{W}^{\mathrm{M}}$ has the same shape as the word embedding matrix defined in Section 5.3.2.3 – only transposed. This is not a coincidence: Given that each column of $\mathbf{W}^{\mathrm{M}}$ is associated with a token in the vocabulary, one can interpret $(\mathbf{W}^{\mathrm{M}})^{\mathsf{T}}$ as a word embedding matrix that learns a representation in the neural network output space for each token in the vocabulary. As the dot-product was esatblished as a measure of similarity (see Section 5.3.2.5), the model output $\mathbf{x}$ is matched against all token embeddings to estimate how close it is to the respective tokens. The training goal of the model is to make it most strongly associated with the column of $\mathbf{W}^{\mathrm{M}}$ corresponding to the actual next token.

Traditionally, input and output embedding matrices were kept separate in neural network architectures. However, motivated by Press and Wolf (2017), Vaswani et al. (2017) tie the token embedding matrix in the input to the output transformation, by fixing $\mathbf{W}^{\mathrm{M}} = \mathbf{E}^{\mathsf{T}}$. Therefore, the network contains only a single embedding matrix which results in the transformer being trained to align its output with the space of input representations. As embedding matrices can make up a significant fraction of a model's parameters, Press and Wolf (2017) show that this strategy drastically reduces model size while also lowering language modelling perplexity due to synergies in the optimisation process.

The transformer language modelling head consists of a final layer normalisation paired with the embedding matrix multiplication just discussed:

Definition 5.25 (LM head)**.** The transformer language modelling head is defined as

$$\mathrm{Head}[\mathbf{E}, \gamma, \beta](\,\mathbf{X}\,) \coloneqq \mathrm{LayerNorm}[\gamma, \beta](_2\,\mathbf{X}\,)\,\mathbf{E}^{\mathsf{T}},$$
$$\mathbf{X} \in \mathbb{R}^{n \times d},\ \mathbf{E} \in \mathbb{R}^{|V| \times d},\ \gamma, \beta \in \mathbb{R}^{d}.$$

This head function yields values in $\mathbb{R}^{|V|}$ that can be interpreted as compatibility scores between hidden representation and word embedding. These scores are also called *logits* (Goodfellow et al., 2016). In order to gain a probability distribution over the vocabulary, normalisation needs to be applied (see the following section).

5.3.2.12 Full transformer

The definition for the complete causal decoder-only transformer with pre-normalisation and positional embeddings used by Radford and Narasimhan (2018) is as follows.

Definition 5.26 (Transformer)**.** Given a number of layers $h \in \mathbb{N}$, a vocabulary V and a maximum sequence length $l \in \mathbb{N}$, a transformer is defined as a parameterised function from sequences $w_{\leq n} \in V^n$ with $n \leq l$ to matrices of logits of shape $n \times |V|$ by

$$\mathrm{Transformer}[\mathbf{E}, \mathbf{P}, \tilde{\mathbf{W}}^{\mathrm{QKV}}, \tilde{\mathbf{W}}^{\mathrm{O}}, \tilde{\mathbf{W}}_1, \tilde{\mathbf{W}}_2, \boldsymbol{\gamma}, \boldsymbol{\beta}](\,w_{\leq n}\,) \coloneqq \mathrm{Head}[\mathbf{E}, \boldsymbol{\gamma}_{(b+1)}, \boldsymbol{\beta}_{(b+1)}](\,\mathbf{A}\,),$$
$$\text{with } \mathbf{A} = \left(\bigcirc_{i=1}^{h} \mathrm{Layer}_i[\tilde{\mathbf{W}}^{\mathrm{QKV}}_{(i)}, \tilde{\mathbf{W}}^{\mathrm{O}}_{(i)}, \tilde{\mathbf{W}}_{1(i)}, \tilde{\mathbf{W}}_{2(i)}, \boldsymbol{\gamma}_{(i)}, \boldsymbol{\beta}_{(i)}]\right)(\mathrm{EmbLayer}[\mathbf{E}, \mathbf{P}](\,w_{\leq n}\,)),$$

$$\mathbf{E} \in \mathbb{R}^{|V| \times d},\ \tilde{\mathbf{W}}_1 \in \mathbb{R}^{h \times (d+1) \times d \cdot f_{\mathrm{ff}}},\ \tilde{\mathbf{W}}_2 \in \mathbb{R}^{h \times (d \cdot f_{\mathrm{ff}}+1) \times d},$$
$$\tilde{\mathbf{W}}^{\mathrm{QKV}} \in \mathbb{R}^{h \times (d+1) \times (3d)},\ \tilde{\mathbf{W}}^{\mathrm{O}} \in \mathbb{R}^{h \times (d+1) \times d},\ \boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^{h \times 2 \times d},\ \mathbf{P} \in \mathbb{R}^{l \times d}$$

In order to gain a probability simplex over the possible vocabulary items at position $i+1$ given the left context $w_1 \ldots w_i$ with $i \in [n]$, the row i in the output logit matrix is normalised using the $\mathrm{softmax}$ function:

$$P(\, t \,|\, w_{\leq i} \,) \coloneqq \mathrm{softmax}\left(\mathrm{Transformer}[\mathbf{E}, \mathbf{P}, \tilde{\mathbf{W}}^{QKV}, \tilde{\mathbf{W}}^{O}, \tilde{\mathbf{W}}_1, \tilde{\mathbf{W}}_2, \boldsymbol{\gamma}, \boldsymbol{\beta}](\, w_{\leq n} \,)_{(i)}\right)_{(\mathrm{idx}_V(t))}, \tag{99}$$

for any $t \in V$.

5.3.3 Estimation

In the following, I will first introduce the transformer loss term and methods for parameter initialisation. Then, optimisation techniques will be presented and finally, regularisation methods are introduced.

5.3.3.1 Loss

Radford and Narasimhan (2018) and Radford et al. (2019) implement the language modelling loss introduced in the introduction to Section 5, averaged by the number of tokens.[21] For a training sequence $w_1 \ldots w_n$ of length n, this amounts to

$$\mathcal{L}[\theta](\, w_1 \ldots w_n \,) = \frac{1}{n-1} \sum_{i=2}^{n-1} -\log P[\theta](\, w_i \,|\, w_{<i} \,) \tag{100}$$

For each position i, the output distribution is conditioned on the actual token at position $i+1$. Note that transformers cannot operate on an empty context and there is no data point for $n+1$ to condition on. Thus, for training material with length n, the length of the model's output is $n-1$.

Transformer models are trained with a pre-defined context window size l due to computation restrictions. For example, `GPT-2` uses a window of $l = 1024$ tokens. Note that this context does not respect sentence boundaries (Radford and Narasimhan, 2018), i.e. it contains several sentences.

Furthermore, the gradient of the loss function with respect to the parameters is approximated using subsets of the training data due to computational considerations. Radford et al. (2019) sample $b = 512$ contiguous subsequences of length l from the training data $w_1 \ldots w_n$ at each optimisation iteration. Such a bundle is called a *minibatch* (Goodfellow et al., 2016, Chapter 5.9) or simply *batch* (Radford et al., 2019).

Definition 5.27 (Batched loss)**.** Formally, let $B = \{\, w_{i_1} \ldots w_{i_1+l}, \ldots, w_{i_b} \ldots w_{i_b+l-1} \,\}$ denote a set of b subsequences of length l with start indices $1 \leq i_1, \ldots, i_b \leq n-l+1$. The corresponding minibatch loss is given by

$$\mathcal{L}[\theta](\, B \,) = \frac{1}{b} \sum_{j=1}^{b} \mathcal{L}[\theta](\, w_{i_j} \ldots w_{i_j+l-1} \,). \tag{101}$$

[21] https://github.com/openai/finetune-transformer-lm/blob/master/train.py

5.3.3.2 Initialisation

The parameters (weights and biases) of a neural network must be initialised prior to training. From a probabilistic perspective, parameter initialisation can be interpreted as specifying a prior distribution over model parameters. Training then corresponds to adjusting these parameters to better fit the observed data under this prior. However, in the light of model size and complexity, it is not transparent what priors to assume and the choice is usually based on proven heuristics (Goodfellow et al., 2016, Chapter 8.4).

Initialisation is necessarily random across the columns of weight matrices since otherwise the respective features of the output would result in exactly the same value and the weights would be deterministically updated in the same way. Random initialisation thus enables training dynamics in the first place (Goodfellow et al., 2016, Chapter 8.4). The choice of random initialisation influences the propagation of activations and gradients through the network and can therefore have a significant impact on optimisation dynamics. Classical initialisation schemes such as Glorot and Bengio (2010) and He et al. (2015) select the variance of weight distributions such that the magnitutes of vector representations are approximately preserved across layers by making the sampling distribution dependent on the respective weight dimensionalities. This is done to prevent activations from growing exponentially or vanishing at the start of training which can result in unstable training or no training at all.

This issue is less pressing in transformer architectures since $\mathrm{LayerNorm}$ is applied extensively. Due to the normalisation, the network becomes less sensitive to the precise scale of the initial weights. Nevertheless, appropriate initialisation remains helpul for stable signal propagation (Xiong et al., 2020).

`GPT-1` (Radford and Narasimhan, 2018) employs a simple normal initialisation for all weight matrices,[22]

$$\mathbf{W}_{(i,j)} \sim \mathcal{N}(0, \sigma^2), \qquad \sigma = 0.02. \tag{102}$$

Bias parameters are initialised to zero. For layer normalisation, the gain parameters are initialised to one and the shift parameters to zero.

In addition, `GPT-2` introduced a depth-dependent scaling to the final weights of both the attention sublayer and the feed-forward sublayer that accounts for the accumulation of residual contributions across model depth (Radford et al., 2019). Concretely, these weights are initialised as

$$\mathbf{W}_{(i,j)} \sim \mathcal{N}\left(0, \frac{\sigma^2}{2h}\right), \tag{103}$$

where h denotes the number of transformer layers and the factor of 2 accounts for the two residual connections per layer. This ensures that the variance of accumulated residual contributions remains approximately constant across depth which can be assumed to stabilise training further.

5.3.3.3 Optimisation

The loss term defined in Section 5.3.3.1 is minimised by computing the gradients of the tunable parameters with respect to it and then adjusting these parameters in the direction of a decrease in

[22] Radford and Narasimhan (2018) specify the distribution to be $\mathcal{N}(0, 0.02)$ which is ambiguous. The naming of the parameters in their implementation suggests that 0.02 referred to the standard deviation (https://github.com/openai/finetune-transformer-lm).

loss (Goldberg, 2017, Chapter 2.7).

Stochastic gradient descent The basic method for training a neural network is *stochastic gradient descent* (SGD) which adjusts the parameters iteratively. Given training data $w_1 \ldots w_n$ and a learning rate $\eta_t > 0$ at iteration t, the parameters are updated according to

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta \mathcal{L}(\, B_t \,|\, \theta_t \,), \tag{104}$$

where the operator ∇_θ yields the gradient with respect to θ, $B_t \sim w_1 \ldots w_n$ denotes the minibatch of examples sampled uniformly from the training corpus at step t and θ_0 comprises the initial parameters discussed in Section 5.3.3.2 (Goodfellow et al., 2016, Chapter 5).

SGD provides an unbiased estimate of the true gradient $\nabla_\theta \mathcal{L}(\theta)$ when minibatches are sampled uniformly, but may suffer from slower convergence due to less precise gradient estimates. On the other hand, the likelihood of stalled training due to local minima of the loss function is reduced (Goodfellow et al., 2016, Chapter 8).

In traditional settings, in cases where the training data already consists of individual sequences, the sampling is realised by randomly shuffling the sequences and then iterating through the data in chunks of b items (Goldberg, 2017, Chapter 5). A full iteration through the training data is called an *epoch*. After an epoch is finished, the dataset is again shuffled and the next epoch initiated. Radford and Narasimhan (2018) use the same epoch-terminology by specifying that `GPT` was trained for 100 epochs. However, it is not entirely clear how the data was partitioned into fixed-length windows. It is possible that it was repartitioned before each epoch or before starting training and it is possible that they used ordinary random sampling which allows for repetition (and overlap) or simple shuffling before an epoch.

Adaptive optimisation (Adam) To address limitations of SDG regarding slower convergence and fluctuating gradients, adaptive optimisation methods adjust the learning rate individually for each parameter. A widely used method is *Adam* (Kingma and Ba, 2017), which maintains exponential moving averages of recent gradients and recent squared gradients. The former smooths the update direction, while the latter is used to adapt the step size of each parameter. This often leads to faster convergence (Pan and Li, 2023) and reduced sensitivity to the choice of hyperparameters (Goodfellow et al., 2016, Chapter 8) compared to SGD. The original transformer implementation used Adam an its optimiser (Vaswani et al., 2017).

Let $g_t = \nabla_\theta \mathcal{L}[\theta_t](\, B_t \,)$ denote the gradient at step t. Adam computes

$$s_t = \rho_1 s_{t-1} + (1-\rho_1) g_t, \qquad r_t = \rho_2 r_{t-1} + (1-\rho_2) g_t \odot g_t, \tag{105}$$

where $\rho_1, \rho_2 \in [0,1)$ control the decay rates of the moving averages (Goodfellow et al., 2016, Chapter 8).[23]

To correct for the bias introduced by initialisation at zero, Adam uses bias-corrected estimates

$$\hat{s}_t = \frac{s_t}{1-\rho_1^t}, \qquad \hat{r}_t = \frac{r_t}{1-\rho_2^t}. \tag{106}$$

[23] The operations are assumed to be performed pointwise over the matrices/vectors part of the parameter collection.

The parameters are then updated as

$$\theta_{t+1} = \theta_t - \eta_t \frac{\hat{s}_t}{\sqrt{\hat{r}_t} + \epsilon}, \tag{107}$$

where $\epsilon > 0$ is a small constant for numerical stability.

Learning rate scheduling For `GPT-1`, the learning rate is scheduled by a linear warm-up followed by a gradual reduction. The maximum learning rate is $\eta_{\max} = 2.5 \cdot 10^{-4}$, and the warm-up phase lasts $t_{\text{warmup}} = 2000$ optimisation steps (Radford and Narasimhan, 2018). Let t_{total} denote the total number of optimisation steps. Following the implementation, during warm-up, the learning rate determined by:

$$\eta_t = \begin{cases} \eta_{\max} \frac{t}{t_{\text{warmup}}} & t \leq t_{\text{warmup}}, \\ \eta_{\max} \cdot \frac{1}{2} \left(1 + \cos\left(\pi \frac{t}{t_{\text{total}}}\right)\right) & t > t_{\text{warmup}}. \end{cases} \tag{108}$$

The cosine, applied after t_{warmup} iterations, has the effect that the learning rate first decreases slowly, faster in the middle, and slowly again near the end. This enables stable initial training and smoother convergence at later phases of training (Loshchilov and Hutter, 2017). For `GPT-2` (Radford et al., 2019), the exact learning rate schedule is not fully specified in the paper.

5.3.3.4 Regularisation

Regularisation techniques are employed to improve the generalisation performance of neural networks by preventing overfitting to the training data. Without such constraints, highly parameterised models may learn spurious patterns or noise in the data, leading to poor performance on unseen inputs (Goodfellow et al., 2016, Chapter 7). Two forms of regularisation were already introduced above: layer normalisation and dropout. In addition to that, `GPT` models make use of so-called *weight decay* which scale downs the magnitude of weights during training.

L^2 regularisation L^2 regularisation is frequently used in neural network training and discourages large parameter values by adding a penalty term to the training objective (Goodfellow et al., 2016, Chapter 7.1),

$$\mathcal{L}_{\text{reg}}[\theta](\, B_i \,) = \mathcal{L}[\theta](\, B_i \,) + \frac{\lambda}{2} \|\theta\|_2^2. \tag{109}$$

Usually, bias terms are not regularised, thus if $\theta = \langle\, \mathbf{W}_1, \ldots, \mathbf{W}_p, \ldots \,\rangle$ enumerates all weights and biases from the collection of model parameters, then

$$\|\theta\|_2^2 \coloneqq \sum_{k=1}^{p} \|\mathbf{W}_k\|_F^2,$$

usually excluding bias terms and layer normalisation parameters, where $\|\cdot\|_F$ denotes the Frobenius norm.

Taking the gradient of the squared L2 penalty in standard SGD (Equation 104) yields

$$\nabla_\theta \frac{\lambda}{2} \|\theta\|_2^2 = \lambda\theta. \tag{110}$$

Thus, under SGD L^2 regularisation is equivalent to shrinking the parameter by a constant factor at each training iteration, which is also known as *weight decay* (Goodfellow et al., 2016, Chapter 7.1). This encourages small parameter norms and improves generalisation by reducing model variance (Goodfellow et al., 2016, Chapter 7.1).

Decoupled Adam For adaptive optimisers such as Adam, L^2 regularisation is not equivalent to weight decay as defined above because the L^2 contribution is included in the gradient and is therefore affected by the optimiser's adaptive rescaling (see Equations 105 to 107). For this reason, Loshchilov and Hutter (2017) proposed a decoupled formulation (*AdamW*), in which parameter shrinkage is applied independently of the gradient update. This is employed in `GPT-1` and `GPT-2` (Radford and Narasimhan, 2018; Radford et al., 2019).

A weight decay coefficient $\lambda = 0.01$ is applied to all weight parameters except bias terms and layer normalisation parameters. Following Loshchilov and Hutter (2017), the update modifies Equation 107 by adding a decoupled decay component, taking the form

$$\theta_{t+1} = \theta_t - \eta_t \lambda \theta_t - \eta_t \frac{\hat{s}_t}{\sqrt{\hat{r}_t} + \epsilon}. \tag{111}$$

5.3.4 Limitations and extensions

Despite their empirical success, transformer language models exhibit several structural limitations. A central limitation is the quadratic complexity of the self-attention mechanism, $\mathcal{O}(n^2 d)$ (Vaswani et al., 2017), which restricts the maximum sequence length n that can be processed in practice. This is usually not particularly problematic for modelling long-range dependencies in natural language, since context windows for contemporary transformer models extend well beyond normal sentence lengths (`GPT-2`: 1024 tokens (Radford et al., 2019)). However, for modelling discourse, scanning longer documents or generating code, this be a practical restriction.

To mitigate the quadratic complexity of self-attention and allow for longer context lengths, various approaches approximate or sparsify the attention computation. These include restricting attention to local windows (Beltagy et al., 2020), using low-rank approximations (Wang et al., 2020), or employing kernel-based methods to linearise the attention operation (Choromanski et al., 2022). Other models have proposed to incorporate recurrence with respect to larger sections of texts (Dai et al., 2019) or external long-term memory (Rae et al., 2019; Wu et al., 2022) into transformers. Finally, recent sequence-model architectures such as selective state-space models aim to replace attention with mechanisms whose cost scales linearly in sequence length, as in `Mamba` (Gu and Dao, 2024).

From the perspective of syntactic and semantic processing in cognitive modelling, another potential caveat to consider concerns the implicit nature of the representations learned by transformer language models.[24] Unlike formal grammars or probabilistic parsers, standard transformers do not explicitly construct symbolic syntactic objects such as constituent trees or dependency graphs. Their predictions are instead based on distributed vector representations. This does not mean that transformers are unable to encode syntactic information: probing studies suggest that contextual

[24] This presupposes that LMs, as models of generation, suggest themselves as objects of investigation for sentence processing despite not being explicitly trained to comprehend language as we assume a human to do. See Section 5.4 for a motivating discussion.

representations can contain substantial information about syntactic structure (Hewitt and Manning, 2019; Li et al., 2023). However, such structure is not represented in an explicit, directly inspectable form.

This matters for the present work since it complicates the development of interpretable structural linking hypotheses. In grammar-based or parser-based models, a structural processing cost can usually be defined directly in terms of the parser state, for example by counting incomplete dependencies or measuring dependency length (see Section 4). In standard transformers, by contrast, any analogous structural cost has to be extracted from continuous and distributed internal quantities such as attention patterns or hidden states. When extractic such a measure and correlating it with experimental data (e.g. reading times), it might be difficult to see what it could teach us about human cognition if we cannot explain how it came about in the first place.

Related to this is the issue of systematic generalisation. Since transformers are not explicitly constrained to represent syntactic relations as hierarchical structures, they may learn generalisations that fit the training data well but do not coincide with the structural rules assumed in linguistic theory. For example, a model might rely on a surface heuristic such as agreement with the nearest noun, even though the grammatically correct rule depends on the hierarchical subject structure. Controlled experiments show that transformers sometimes fail in such cases where hierarchical and surface-based generalisations make different predictions (McCoy et al., 2020; Petty and Frank, 2021; Mueller et al., 2022). This suggests that a lack of bias combined with overparameterisation prevents transformers to adopt economical hierarchical generalisations, instead making it depend on surface heuristics.

For this reason, several lines of work have explored ways of adding stronger structural inductive biases to transformer models. Some approaches augment attention with hierarchical constraints, while others combine transformers with explicit grammar-based mechanisms (Wang et al., 2019; Sartran et al., 2022; Gessler and Schneider, 2023). These modifications are motivated not only by performance considerations with respect to language modelling, but also by the desire to obtain representations whose relation to linguistic structure is more direct. However, so far, results in this line of research are mixed, with recent evaluations raising doubts with respect to the structural consistency of these approaches (Arps et al., 2025).

5.4 Transformers in sentence processing

The strong performance of transformer language models has made them an important tool in computational psycholinguistics. However, their use requires some caution. A neural language model is, in the first instance, a model of language as a statistical phenomenon: it estimates probability distributions over linguistic sequences from patterns in the training data. It is not, by construction, a model of human language processing or comprehension. Any interpretation of its internal computations as approximations of cognitive operations therefore requires a linking hypothesis.

For surprisal (see Section 4.2.1), this link is relatively direct: language models estimate conditional word probabilities which can be used as an approximation for the unobserved probability that the comprehender assigns to the incoming word.

For other quantities, such as attention entropy, which will be discussed in the following sections, the connection is more indirect. These measures are motivated by the hypothesis that successful language prediction requires the model to represent, store and access information about preceding

context in ways that may partly overlap with the memory operations involved in human sentence processing (Oh and Schuler, 2025).

In the following, I will outline the most influential links relating transformer characteristics with psycholinguistic measurements that research has proposed thus far.

5.4.1 Next-token probability

Transformer language models are trained to produce a probability distribution over next tokens given a left context. This is done by conditioning on the minimisation of surprisal with respect to a training corpus of text (Equation 61). Their parameterisation allows the models to generalise to unseen contexts. As a result, they provide general estimates of the empirical conditional distribution $P(\, w_i \,|\, w_{<i} \,)$ and can be used to approximate the theoretical and unobservable human distribution assumed by surprisal theory.

Surprisal extraction from such transformer is conceptually straightforward. Given a sequence $w_1 \ldots w_n$ for which surprisal should be computed, the model is evaluated in parallel on each prefix $w_{<i}$ with $i < n$, and the probability assigned to the actual next token w_{i+1} is recorded. Surprisal is then obtained as the negative logarithm of this probability. This procedure is standard in psycholinguistic applications of transformer models (e.g. Wilcox et al., 2023; Oh and Schuler, 2023a).

5.4.1.1 Model quality

For the following discussion, it is necessary to introduce the measure of *perplexity*.

Definition 5.28 (Perplexity of a language model)**.** The *perplexity of language model* $P[\theta]$ *on a sequence* $w_1 \ldots w_n$ is defined as the exponentiation of the average surprisal of $P[\theta]$ on the corpus:

$$\mathrm{PPL}(w_1 \ldots w_n, P[\theta]) := b^{\frac{1}{n} \sum_{i=1}^{n} \mathrm{I}_{P[\theta]}(\, w_i \,|\, w_{<i} \,)},$$

with $b > 1$ being a base.

Note that this measure amounts to the inverse geometric mean of the corpus probability (Jurafsky and Martin, 2026, Chapter 3.3):

$$\begin{aligned} b^{\left(\frac{1}{n} \sum_{i=1}^{n} \mathrm{I}_{P[\theta]}(\, w_i \,|\, w_{<i} \,)\right)} &= \prod_{i=1}^{n} b^{\left(-\frac{1}{n} \log_b P[\theta](\, w_i \,|\, w_{<i} \,)\right)} \\ &= \prod_{i=1}^{n} P[\theta](\, w_i \,|\, w_{<i} \,)^{-\frac{1}{n}} = \frac{1}{\sqrt[n]{P[\theta](\, w_1 \ldots w_n \,)}}. \end{aligned} \quad (112)$$

Thus, lower perplexity corresponds to lower average surprisal and indicates a better fit of the model to the distribution of the data. Since $P(w_1 \ldots w_n)$ decreases exponentially with sequence length n, the normalisation by $\sqrt[n]{}$ makes perplexity comparable across sentences. As negative log-probability is the standard objective function used in language modelling (see Section 5.3.3.1), perplexity is simply a log-transformation of the loss of the model.

Perplexity can be interpreted as the number of choices the model is uncertain about at each token position (Jurafsky and Martin, 2026, Chapter 3.3.1). For instance, for a perplexity of 50, the result

is as good as choosing uniformly from 50 equally likely word options at each step. Perplexity on held-out data is often used as an intrinsic measure of language modelling quality. However, note that it is not strictly comparable across different models due to differences in their underlying vocabulary changing the prediction events of the measurements (Jurafsky and Martin, 2026, Chapter 3.3). Furthermore, perplexity is generally discouraged to be used as the sole metric for evaluating language model quality. Instead, evaluation should also be performed on a real task (Jurafsky and Martin, 2026, Chapter 3.3).

Quality-power hypothesis The *quality-power (QP) hypothesis* posits that higher quality language models provide more accurate reading time predictors (Wilcox et al., 2025). The intuition of this assumption is that better LMs are closer to the assumed cognitive distribution underlying surprisal theory.

Early tests of computational language models indicated that the relationship between word predictability and reading times is logarithmic (Smith and Levy, 2013) or linear (Goodkind and Bicknell, 2018). In particular, Goodkind and Bicknell (2018) relate LM perplexity of n-gram models and recurrent neural networks with the Δ_{llh} (difference in log-likelihood, cf. Section 6) of linear models that fit surprisal from the respective LM to human reading times. They found that models with lower perplexity on held-out data produce more predictive surprisal values. Wilcox et al. (2020) confirm these results, including a transformer model of the `GPT-2` architecture that they trained from scratch. Note however, that the amount of data they use to train it is significantly smaller than the material that was used by Radford et al. (2019) to train the `GPT-2` model family. Additionally, they observe that a model's syntactic knowledge does not correlate with its psychometric predictive power.

More recent evaluations report a non-monotonic relationship between quality and psychometric predictive power. Wilcox et al. (2025) find that language model quality and fit to reading times are correlated for a majority of investigated languages – but not for Finnish and German. Using n-gram models, LSTMs and transformers, Kuribayashi et al. (2021) observe that the proposed linear relationship does not hold for Japanese.

The introduction of large-scale transformer models with greatly reduced perplexity has provided additional evidence: using models of between 124 million (`GPT-2 small`) and 66 billion parameters, Oh and Schuler (2023b) find that larger architectures yield poorer RT fit than their smaller counterparts. Their analysis reveals that this mainly stems from underprediction of reading times for nouns and adjectives, especially named entities and predicative adjectives, and overprediction for function words. Liu et al. (2024) confirm the decrease in predictive power for the larger variants of the `GPT-2` model family. Similarly, Shain et al. (2022) find that `GPT-3` (Brown et al., 2020) predicts RTs worse than `GPT-2 small` and Oh and Schuler (2025) confirm this tendency for `Mamba-2` state space models (Dao and Gu, 2024). A complementary result is reported by Oh and Schuler (2023a), who train transformer language models while controlling the architecture and varying the amount of training data. They find that surprisal predicts human reading times best after approximately two billion training tokens. Beyond this point, held-out perplexity continues to improve, whereas psychometric predictive power decreases. This suggests that the loss of human-like predictiveness in very strong language models cannot be attributed only to model size but that the amount of linguistic exposure during training may also play an important role.

A possible explanation is investigated by Wilcox et al. (2025), who note that the issue might be

rooted in leakage during model training, i.e. duplicate data in training and validation splits. Since the training data of large models is usually not made publicly available, they perform a simulation by training models with custom leaked data, finding that Δ_{llh} does not increase in cases of large duplication rates. This suggests that the reports are not rooted in compromised statistics.

Oh and Schuler (2023b) attribute the phenomenon to large transformers' tendency to memorise rare sequences, enabled by high model capacity. A related explanation would be that the predictive effect in human sentence processing is constrained by limited capacities. In this case, large transformers could have already reached superhuman language modelling performance.

An explanation put forward by Liu et al. (2024) proposes that very large language models are more confident of their predictations than humans, i.e. they assign higher probabilities to items than ground-truth would support, due to having seen significantly more data during training than humans observe in the course of language acquisition. They approach this hypothesis via *temperature-scaling*. This common technique in the field of model calibration tunes the temperature parameter of the softmax function (see Definition 2.21) in a pre-trained language model, changing the degree of stochasticity of the predictive distribution that surprisal depends on. Liu et al. (2024) find that the poorer fit of surprisal from large language models to RTs cannot be explained by a degradation in calibration, since it actually improves with model size. However, a temperature of $t \approx 2.5$ (as opposed to $t = 1$ in regular surprisal) improves their reading time predictions significantly, with the optimal temperature increasing with LM size. Thus, larger models require stronger manual flattening of their next-token distributions. Closer investigation showed that named entities profit the most from this scaling, largely consistent with the observations of Oh and Schuler (2023b). Furthermore, smoothing increases very low probabilities, also shown to improve RT fit (Liu et al., 2024).

Finally, Liu et al. (2024) show that the benefit of temperature scaling is strongly tied to subword tokenisation. Word-level surprisal is usually computed by summing the surprisals of all subword tokens belonging to a word (Equation 15). Temperature scaling affects this sum differently for words represented by a single token and words represented by multiple tokens. Liu et al. (2024) find that the improvement in reading-time prediction is mainly driven by multi-token words, for which temperature scaling increases word-level surprisal in the vast majority of cases. This raises the possibility that the psycholinguistic benefit of temperature scaling is partly an artefact of subword tokenisation rather than direct evidence that human predictive distributions are uniformly flatter than those of large language models. Multi-token words also tend to be longer or rarer and may require more visual or lexical processing independently of predictive probability. Thus, temperature scaling may improve reading-time fit because it indirectly compensates for tokenisation- and length-related factors (Liu et al., 2024).

5.4.1.2 Tokenisation

So far, I have assumed that language model units (tokens) align with human processing units (called *regions of interest* (ROI), Giulianelli et al. (2024)). In practice, reading time measurements for English data are mostly provided for white-space separated regions, corresponding to what is commonly assumed to constitute a word. Since transformer language model tokenisers (e.g. BPE tokenisers, see Section 5.3.2.2) are configured not to produce tokens that span spaces, a reading time ROI usually consists of a concatenation of one or more language model tokens. As was shown in Section 4.2.1, the surprisal for this region can be gained in a straightforward fashion by

summing over the token surprisals.

Despite this simple solution, tokenisation is a non-trivial modelling choice and empirical work shows that it matters with respect to surprisal's psychometric predictive power. Initial hints were provided by Oh et al. (2021) who demonstrate that incorporating morphological transformations into a parser-based model improves the fit of surprisal to reading times. Nair and Resnik (2023) investigate this issue for explicit language modelling with n-grams by fitting data tokenised using BPE, morphologically motivated splits and word-level (whitespace) tokenisation but find no statistically relevant differences in predictive power between these representations. On the surface, this suggests that tokenisation choice may not substantially affect surprisal-based models of reading times. However, Nair and Resnik (2023) find that most ROIs in the Dundee and Natural Stories corpora correspond to single-token items and that aggregate results hide the fact that when restricting the analysis to segmented words only, surprisal derived from BPE tokenisation yields significantly worse predictions of RTs than morpheme-based surprisal.

Even when having settled for a tokenisation scheme such as BPE, the chosen *granularity*, i.e. the number of distinct items to produce via merge operations, further affects results. Oh and Schuler (2025) train `Mamba-2` state-space models (Dao and Gu, 2024) with various vocabulary sizes and find that a size of 8000 yields the most predictive surprisal values – exceeding the predictors provided by `GPT-2` despite its far greater vocabulary size of 50.257 tokens. They suggest that the effect of more fine-grained tokens on the LM might be two-fold: (i) altering the initial distribution to be closer to a model determined by word-length and (ii) by inducing a bias during training.

Surprisingly, a qualitative analysis shows that garden-path effects are better modelled using coarser tokensisation despite the aggregate improvements provided by a smaller vocabulary. Oh and Schuler (2025) suggest that this might be explained by associations between words (e.g. syntax) being more indicative for predictions in these cases, which might be harder to learn for a model that splits words into a larger amount of subtokens. Thus, altering token vocabulary sizes appears to be a double-edged sword with respect to psycholingusitic modelling.

On the other side of the spectrum, Giulianelli et al. (2024) take the stance that the choice of tokenisation is largely irrelevant as the resulting probabilistic model can be marginalised into a model over character sequences. Their proposal is of practical relevance for investigating ROIs that involve parts of words, where tokenisation and measurements cannot be expected to align. While researchers could conduct experiments aligned with BPE tokenisation, the resulting units would be rather unnatural and thus generate interference with reading processes (Giulianelli et al., 2024). Furthermore, computational psycholinguists usually use preexisting data which they need to process.

Finally, Giulianelli et al. (2024) propose a more comprehensive approach with respect to reading time modelling with language models, disentangling the one-to-one correspondance between ROI and LM surprisal region (which they call the *focal area*). Motivated by eye-tracking research on lexical processing and preview effects, Giulianelli et al. (2024) use their marginalisation technique to test whether RTs are better modelled using only a word's first n characters and if preview of the following word can improve RT predictions further. They find that smaller or forward-looking regions can outperform full-word surprisal, indicating sensitivity to subword and parafoveal information.

5.4.2 Attention structure

The preceding section discussed surprisal as a predictor of reading times derived from the next-word probabilities of large transformer language models. Surprisal therefore treats the model primarily as a source of conditional probability estimates. The following section shifts attention from the model's output distribution to its internal attention structure. This is motivated by the idea that self-attention may provide computationally accessible analogues of memory retrieval operations in sentence processing (Ryu and Lewis, 2021; Ryu and Lewis, 2022; Oh and Schuler, 2022).

5.4.2.1 Transformer attention as differentiable retrieval

Ryu and Lewis (2021) propose that transformer attention can be interpreted as a computational realisation of a cue-based retrieval mechanism (Section 4.3.2). They found that self-attention implements a structurally analogous operation to retrieval in continuous vector space, producing similar interference patterns.

Assume key and query matrices $\mathbf{K} = \mathbf{X}_{/\mathbf{1}}\tilde{\mathbf{W}}^{\mathrm{K}}$ and $\mathbf{Q} = \mathbf{X}_{/\mathbf{1}}\tilde{\mathbf{W}}^{\mathrm{Q}}$ for an input sequence $w_1 \ldots w_n$ encoded in $\mathbf{X}$, and a causal mask $\mathbf{M}$. At position $i \leq n$ masked self-attention computes a probability simplex $\mathbf{A}_{(i)}$ over positions $[n]$ defined as

$$\mathbf{A}_{(i)} = \mathrm{softmax}\left(\frac{\mathbf{Q}_{(i)}\mathbf{K}^{\mathsf{T}}}{\sqrt{d_{\mathrm{k}}}} + \mathbf{M}_{(i)}\right). \tag{113}$$

$\mathbf{A}_{(i,j)}$ is the *attention score* assigned to the context item j by the attention head when processing the item at position i.

Due to the causal mask, all positions larger than i are assigned a probability of 0, so that the distribution can be reinterpreted as a distribution over token positions $[i]$, i.e. $\mathbf{A}_{(i,1:i)}$. Thus, the items in the lower triangle of $\mathbf{A}$, including the diagonal, provide distributions over contextual token positions for all items of the input sequence.

In the following, I adopt terms from probability theory where convenient due to the rows of $\mathbf{A}$ being probability simplexes. Note hower, that this does not license a probabilistic interpretation under which attention heads take samples out of the left context (Ryu and Lewis, 2021). Rather, the raw dot-products $\mathbf{Q}_{(i)}\mathbf{K}^{\mathsf{T}}$ record the compatibilities between the query and the context items and the application of the $\mathrm{softmax}$ established a comparison between compatibility levels.

Now, under analogy with ACT-R cue-based retrieval[25] (Section 4.3.2.2), at processing step i, the *cues* of the goal item correspond to the entries in the vector $\mathbf{Q}_{(i)}$. In the transformer model, these are selected from the lower-level representation of the item just read via the affine transformation $\tilde{\mathbf{W}}^{\mathrm{Q}}$ and represented continuously in d_{k}-dimensional space. In Lewis and Vasishth (2005)'s ACT-R implementation, they are derived from the lexicon entry of the new item and the current goal, which represents a form of recurrence.

In the same fashion, the features of the chunks in declarative memory correspond to the values of the vectors in $\mathbf{K}$. In both formalisms, each memory item is associated with one lexical item.

[25] Lewis and Vasishth (2005)'s ACT-R model is only one possible implementation of the ideas outlined by Van Dyke and Lewis (2003) and comes with additional design choices. However, it allows for an explicit comparison of formalised objects, aiding in the illustration of the striking similarities between cue-based retrieval and attention.

Additionally, the keys/features remain the same across all queries/retrievals. In attention, they are encoded by a transformation of the lower-level representation of the respective tokens using $\tilde{\mathbf{W}}^{\mathrm{K}}$, while they are determined by an item's lexicon entry and the chunk retrieved from memory at that position in ACT-R.

Finally, the attention score $\mathbf{A}_{(i,j)}$ would have its analogue in ACT-R in the normalised sum of associative strenghts $\sum_k W_k S_{k,j}$ (Equation 53) with $k \in [d_{\mathrm{k}}]$ iterating through the values of the query. In the same way as the magnitude of the attention score decreases if other items in the context have a high compatibility with the query, the associative strengths in ACT-R are lower if the cues are shared by other chunks in memory.

The main difference lies in two aspects: first, in constrast to attention scores, the activations A_j for memory items $j < i$ do not sum to 1. They are assumed to live in logit-space. And second, feature/cue matching is binary in cue-based retrieval and determined by equality – either they are present and match or they don't and can be ignored – while individual features of query/key vectors are encoded in $\mathbb{R}$ in attention and have their compatiblity determined by multiplication, again yielding a value in $\mathbb{R}$. Thus, features are continuous and can be graded, allowing for greater expressive power.

Additionally, the retrieved content is determined by the weighted sum of value vectors in attention while it is equal to the single retrieved memory chunk in cue-based retrieval. Thus, cue-based retrieval assumes only a limited form of contextualisation compared to attention: all of the context is queried, but usually only one item can be used to influence the character of the memory representation for the currently processed item.

In contrast to traditional approaches to cue-based retrieval, in attention, features do not need to be assigned manually. They are learned as a side-product of the self-supervised language modelling objective of the transformer. This conveniently fills the lack of a theory of cues in cue-based retrieval accounts (Parker et al., 2017) and eliminates a significant factor for researcher degrees of freedom. However, high dimensional continuous cue vectors have the downside of reduced interpretabiltiy. Furthermore, using these vectors in the context of cue-based retrieval is connected to the assumption of task-otpimality, i.e. that the transformer language model learns representations which carry information that would likewise be needed cognitively for the goal of comprehension.

Lastly, in contrast to ACT-R based implementations of cue-based retrieval, attention does not incorporate an explicit form of activation decay. Preference to local items in context is either induced by relevant tokens appearing closer statistically (cf. Futrell et al., 2015) or through weights being sensitive to positional encodings.

5.4.2.2 Retrieval targets

If attention is interpreted as implementing a retrieval mechanism, a central question concerns what representations are retrieved and how these relate to linguistic structure. While Van Dyke and Lewis (2003) assume that the purpose of retrieval is to construct a parse, with the retrieved chunk being related to the newly read word by means of syntactic dependency, the transformer constructs no such symbolic parse. But what guides attention then?

Several probing studies suggest that syntactic information is encoded in the internal representations of transformer models. For instance, Hewitt and Manning (2019) show that dependency-tree distances can be recovered from a linear transformation of contextual word representations and

Tenney et al. (2019) find that dependency information is localised in middle layers. Analyses of attention weights show that individual heads often assign high mass to tokens standing in particular syntactic relations. Clark et al. (2019), for example, identify heads that attend to direct objects, determiners, objects of prepositions, and coreferent mentions, while Vig and Belinkov (2019) report that attention in `GPT-2` aligns most strongly with dependency relations in middle layers. Similar findings have been reported for transformer models in machine translation, where some heads specialise in syntactic relations (Voita et al., 2019).

Since attention weights are computed from query-key compatibilities, such findings suggest that some query and key projections induce a similarity space in which syntactically related items receive high compatibility scores. However, this behaviour is head-specific: many heads instead track positional relations, sentence boundaries, or broad contextual information and no single head appears to account for full sets of dependencies necessary to recover a dependency parse (Clark et al., 2019; Voita et al., 2019; Htut et al., 2019).

A complication arises from the fact that transformer attention also assigns probability mass to the current position. From the perspective of retrieval-based theories, this raises the question of whether attention to the current token should be interpreted as a form of retrieval at all. One possible interpretation is that self-attention to the current position does not reflect retrieval, but rather constitutes a form of residual processing with an additional value-transformation to combine memory items and the query item. Under this view, only attention to preceding positions corresponds to genuine retrieval operations. However, attention to the current item is not strictly necessary to generate a combined output, as a representation of the current item is also made available via the residual connection that embeds the attention layer.

Alternatively, the key for the current item could be a sort of baseline. In cases in which little or no preceding tokens match the current cues well, attention could be higher on the query item, preventing the retrieval of irrelevant information. Note that this remark is speculative.

5.4.2.3 Interference as attention diffuseness

The central prediction of cue-based retrieval theory is the emergence of interference effects in the presence of distractors, assumed to increase human reading times (Van Dyke and Lewis, 2003). If attention is interpreted as a form of cue-based retrieval, the question emerges how to quantify its interference effects. Classical cue-based retrieval inverts the activation value assigned to the single retrieved item to compute latency (see Equation 57). This measure decreases when the number of other items in the memory matching the cues increases. However, there is no single identifiable retrieved memory item in dot-product attention.

Therefore, Ryu and Lewis (2021) propose a method for capturing the competition between matches: attention entropy. If only a single memory item receives high attention, the entropy of the attention distribution is is low, reflecting low latency due to competition. But if attention is uniformly distributed across memory items, entropy is high, mirroring high latency due to strong competition.

Let's define this formally:

Definition 5.29 (Attention distribution)**.** Assume a transformer with parameters θ and weights $\tilde{\mathbf{W}}^{\mathrm{Q}}_{l,h}, \tilde{\mathbf{W}}^{\mathrm{K}}_{l,h} \in \theta$ for the attention head h at layer l. Let $w_1 \ldots w_n$ be an input sequence encoded into $\mathbf{X}$ by the preceding modules under parameterisation θ. For any position i in $[n]$, the function

Attn yields a vector of *attention scores* attributed to the input items in the context (positions $1, \ldots, i$):

$$\mathrm{Attn}_{l,h}[\theta](\, w_i, w_{<i}\,) \coloneqq \mathbf{A}_{(i,1:i)},$$

where

$$\mathbf{A} = \mathrm{softmax}\left(\frac{\mathbf{X}\tilde{\mathbf{W}}^{\mathrm{Q}}_{l,h}(\mathbf{X}\tilde{\mathbf{W}}^{\mathrm{K}}_{l,h})^{\mathsf{T}}}{\sqrt{d_{\mathrm{k}}}} + \mathbf{M}\right),$$

with $\mathbf{M}$ being a causal mask.[26]

In the following, the pair $\langle\, l, h\,\rangle$ will be used to refer to a specific attention head of a transformer and the parameter θ will be used without further specification to denote all necessary parameters for the respective attention head and all preceding layers defined by the transformer function's specification.

Note that $\mathrm{Attn}_{l,h}$ yields a probability simplex with every entry being larger than 0. This motivates the extension of entropy (see Definition 2.24) to vectors:

Definition 5.30 (Vector entropy)**.** Vector entropy for some base $b > 1$ and vector dimensionality $d \in \mathbb{N}$ is defined as a function,

$$\mathrm{H}_b \colon (0,1]^d \to \{\, x \in \mathbb{R} \mid x \geq 0 \,\},$$
$$\mathbf{v} \mapsto \textstyle\sum_{i=1}^{n} \mathbf{v}_{(i)} \log_b \mathbf{v}_{(i)}.$$

If $\mathbf{v}$ encodes the probabilities of a random variable, vector entropy is equal to the regular notion of entropy. However, note that $\sum_{i=1}^{|\mathbf{v}|} \mathbf{v}_{(i)} = 1$ is not strictly required here. Thus, rather than encouraging a probabilistic interpretation, this measure should rather be treated as a quantification of dispersion (Ryu and Lewis, 2021).

Finally, Ryu and Lewis (2021) define entropy over the left context for a token for each attention head as follows:

Definition 5.31 (Restricted attention entropy)**.** Given a transformer with attention head $\langle\, l, h\,\rangle$ and an input sequence $w_1 \ldots w_n$, for position $i \in \{\, 2, \ldots, n\,\}$, the function $\mathrm{entr}^{\mathrm{restr}}_{l,h}$ yields the entropy of the attention vector restricted to the left context $\{\, 1, \ldots, i-1\,\}$,[27]

$$\mathrm{entr}^{\mathrm{restr}}_{l,h}[\theta](\, w_i, w_{<i}\,) \coloneqq \mathrm{H}_b(\mathrm{Attn}_{l,h}[\theta](\, w_i, w_{<i}\,)_{(1:i-1)}),$$

Ignoring attention to the current position has the effect of reducing entropy when competition with attention to the query item itself is high. However, this does not purely confine the measure of competition to the items in strict left context. Take for instance a case where the memory consists of two items. Assume that one item receives 0 attention and the other one some amount a. Evidently, there is no competition between memory items. However, if the current item is also assigned attention a such that $a = 0.5$, then $\mathrm{entr}^{\mathrm{restr}}$ evaluates to roughly $0.5 \cdot \log_2 0.5 = 0.5$ (assuming base $b = 2$) instead of 0.

[26] Note that all other entries of the row $\mathbf{A}_{(i)}$ are equal to 0 due to the causal mask.

[27] Ryu and Lewis (2021) use base $b = 2$.

Ryu and Lewis (2021) use the measure of attention entropy to investigate the dispersion of attention over retrieval targets in in the `GPT-2 medium` transformer. Increased entropy is interpreted as reflecting similarity-based interference, as attention is distributed across multiple partially matching items.

To this end, they identify attention heads specialised for syntactic dependencies using a data-driven procedure adapted from Voita et al. (2019): Given a dependency-annotated dataset, for each attention head $\langle\, l, h \,\rangle$ a syntactic accuracy is computed by determining how often the context item with the highest attention is a syntactic dependent of the query (e.g. the subject for a verb). A head is considered specialised if its accuracy exceeds a principled baseline (10% in Ryu and Lewis 2021). Among the specialised heads, Ryu and Lewis (2021) select head $\langle\, 4, 3 \,\rangle$, which exhibits the highest accuracy, for further analysis.

They evaluate entropy at specific integration sites in input sequences corresponding to the resolution of syntactic dependencies (e.g. verbs appearing after their subjects and reflexive pronouns referring to some preceding noun). The integration site is called the *target* (Ryu and Lewis, 2021). For each item, minimal sentence pairs $w_1^{\text{interf}} \ldots w_i^{\text{interf}}$ and $w_1^{\text{none}} \ldots w_i^{\text{none}}$ with $w_i^{\text{interf}} = w_i^{\text{none}}$ are constructed that differ only in the presence or absence of an interfering *distractor* that shares some grammatical features with the target.

Let $j_{\text{target}} < i$ be the position of the target (equal in both versions of the sentence). At the critical position i, the attention distribution of head $\langle\, 4, 3 \,\rangle$ is computed, yielding both target attention,

$$A^x = \text{Attn}_{4,3}[\theta](\, w_i^x, {w^x}_{<i} \,)_{(j_{\text{target}})}, \tag{114}$$

and attention entropy,

$$H^x = \text{entr}_{4,3}^{\text{restr}}[\theta](\, w_i^x, {w^x}_{<i} \,), \tag{115}$$

with $x \in \{\, \text{interf}, \text{none} \,\}$.

These quantities are then compared across conditions at the same position:

$$H^{\text{interf}} > H^{\text{none}}, \qquad A^{\text{interf}} < A^{\text{none}}. \tag{116}$$

Using this method, Ryu and Lewis (2021) find significant interference effects at the level of individual words, showing that attention becomes more diffuse and less focused on the correct retrieval target precisely at the point of integration. This confirmed that heads specialised to retrieve syntactic dependents are subject to interference effects of non-related items with similar continuous features.

Prediction of reading times Naturally, the results of Ryu and Lewis (2021) raised the question whether interference effects measured using transformer attention would be indicative of processing difficulties reflected in human reading times. An initial investigation on this matter was conducted by Ryu and Lewis (2022). In contrast to earlier work, entropy is computed not only for specific attention heads but also aggregated across the whole transformer, although the precise strategy for aggregation is not reported.

Formally, processing difficulty as measured through self-paced reading times on the Natural Stories corpus is modelled as a joint function of surprisal and attention entropy from `GPT-2`. Attention entropy emerges as an independent predictor, albeit only as a spillover effect (Ryu and Lewis, 2022).

Motivated by this fact, Oh and Schuler (2022) propose a series of modifications to the entropy extraction method. First, they introduce two alternative strategies for calculating attention weights. These are based on observations that the contribution of a token to the output of an attention module is not only determined by the magnitude in dot-product attention but also by the value and MHA output transformations (Kobayashi et al., 2020) and by the residual connection and layer normalisation (Kobayashi et al., 2021). Thus, rather than only capturing competition with respect to the dot-product similarity measure, the methods proposed by Oh and Schuler (2022) aim to quantify relative contributional strength of items partaking in a form of weighted retrieval.

First, based on scaling strategies proposed by Kobayashi et al. (2020), *norm-aware* attention weights are introduced. For their definition, the normalisation function uscale will be necessary.

Definition 5.32 (Unit scaling)**.** For any dimensionality $d \in \mathbb{N}$, the function uscale is defined as

$$\begin{aligned} \text{uscale}\colon \{\, x \in \mathbb{R} \mid x \geq 0 \,\}^d &\to \Delta^d, \\ \mathbf{v} &\mapsto \frac{1}{\sum_{i=1}^{d} \mathbf{v}_{(i)}} \mathbf{v}. \end{aligned}$$

Now, given that the output of attention is a weighted sum of the entries of a value matrix $\mathbf{V}$, transformed in a second step by a slice of the multi-head attention output transformation $\mathbf{W}^{\text{O}}$ (see Section 5.3.2.6), the strength of the contribution of each context token can be measured by taking the norm of the vector before taking the sum (Oh and Schuler, 2022).

Definition 5.33 (Norm-aware attention)**.** Given a transformer with m parallel attention heads, parameters weights $\tilde{\mathbf{W}}^{\text{V}}_{l,h}, \tilde{\mathbf{W}}^{\text{O}}_{l}$ for attention head $\langle\, l, h \,\rangle$ and an input sequence $w_1 \ldots w_n$ encoded into $\mathbf{X}$ by the preceding modules, for any position $i \in [n]$, the function $\text{Attn}^{\text{N}}_{l,h}$ yields a probability simplex over positions in the left context (including w_i) defined as follows:

$$\text{Attn}^{\text{N}}_{l,h}[\theta](\, w_i, w_{<i} \,) \coloneqq \text{uscale}\Big(\|_2 \tilde{\mathbf{V}}_{l,h} \|_2 \odot \text{Attn}_{l,h}[\theta](\, w_i, w_{<i} \,) \Big)\,,$$

where

$$\tilde{\mathbf{V}}_{l,h} = \mathbf{X}_{/\mathbf{1}} \tilde{\mathbf{W}}^{\text{V}}_{l,h} \mathbf{W}^{\text{O}}_{l} + m^{-1} \mathbf{b}^{\text{O}}_{l},$$

and $\|_2 \cdot \|_2$ denotes the row-wise application of the vector norm. The optional output transformation bias/intercept vector $\mathbf{b}^{\text{O}}_{l}$ is distributed uniformly across heads or omitted for a transformer without bias terms.

Additionally, Kobayashi et al. (2021) demonstrated that both layer norm and residual connections can be equivalently applied to the value vectors before weighing and summing them using the attention scores. This motivates the introduction of *residual layernorm norm-aware attention* (Oh and Schuler, 2022).

Definition 5.34 (RL norm-aware attention)**.** Given a transformer with m parallel attention heads, head $\langle\, l, h \,\rangle$, layer norm shift and scale parameters $\boldsymbol{\gamma}_{l,2}, \boldsymbol{\beta}_{l,2}$ for the LayerNorm applied to the input of the feed-forward network in layer l and an input sequence $w_1 \ldots w_n$ encoded into $\mathbf{X}$ by the preceding modules, for any position $i \in [n]$, the function $\text{Attn}^{\text{RL-N}}_{l,h}$ yields a probability simplex over positions in the left context (including w_i) defined as follows:

$$\text{Attn}^{\text{RL-N}}_{l,h}[\theta](\, w_i, w_{<i} \,) \coloneqq \text{uscale}\Big(\|_2 \, g(\tilde{\mathbf{V}}_{l,h}) \|_2 \odot \text{Attn}_{l,h}[\theta](\, w_i, w_{<i} \,) \Big)\,,$$

with $\tilde{\mathbf{V}}_{l,h}$ defined as in Definition 5.33, the function g over matrices defined as

$$g(\mathbf{Y})_{(:,j)} \coloneqq \begin{cases} \mathbf{0} & \text{if } j > i, \\ \frac{\mathbf{Y}_{(:,j)} + f(\mathbf{Y})_{(:,j)}}{\mathrm{std}\left(\mathbf{X}^{\mathrm{resid}}_{(i)} + \sum_{h=1}^{m} \mathbf{a}_{l,h}^{\mathsf{T}} \tilde{\mathbf{V}}_{l,h}\right)} \odot \boldsymbol{\gamma} + m^{-1}\boldsymbol{\beta} & \text{if } j \leq i, \end{cases}$$

and function f over matrices defined as

$$f(\mathbf{Y})_{(:,j)} \coloneqq \begin{cases} -\operatorname{avg}\left(\mathbf{Y}_{(:,j)} + \frac{\mathbf{X}^{\mathrm{resid}}_{(i)}}{m \cdot \mathbf{a}_{l,h\,(i)}}\right) + \frac{\mathbf{X}^{\mathrm{resid}}_{(i)}}{m \cdot \mathbf{a}_{l,h\,(i)}} & \text{if } i = j, \\ -\operatorname{avg}\left(\mathbf{Y}_{(:,j)}\right) & \text{if } i \neq j \end{cases}$$

where avg and std are the mean and standard deviation defined on vectors, $\mathbf{a}_{l,h} = \mathrm{Attn}_{l,h}[\theta](\, w_i, w_{<i}\,)$ and $\mathbf{X}^{\mathrm{resid}}$ is the input of the MHA, before applying layer normalisation.

Based on the standard attention scores and the two preceding alternations, Oh and Schuler (2022) compute three different entropy measures. Compared to Ryu and Lewis (2021, here: Definition 5.31), they introduce two important modifications: (i) the distribution over preceding context is re-scaled to sum to 1 after removal of the attention score assigned to the current item and (ii) entropy for each token position is normalised by the maximum entropy achievable given the size of the respective context.

Definition 5.35 (Strict left-context attention)**.** The function $\underleftarrow{\mathrm{Attn}}^{\pi}_{l,h}$ with $\pi \in \{\, \epsilon, \mathrm{N}, \text{RL-N} \,\}$,

$$\underleftarrow{\mathrm{Attn}}^{\pi}_{l,h}[\theta](\, w_i, w_{<i}\,) \coloneqq \mathrm{uscale}\big(\mathrm{Attn}^{\pi}_{l,h}[\theta](\, w_i, w_{<i}\,)_{(1:i-1)}\big)$$

is defined as the *strict left-context attention distribution*. The ϵ-variant refers to the ordinary attention distribution $\mathrm{Attn}_{l,h}$.

Definition 5.36 (Strict left-context attention entropy)**.** Given a transformer and an input sequence $w_1 \ldots w_n$, for any position $i \in \{\, 2, \ldots, n \,\}$, the function $\underleftarrow{\mathrm{entr}}^{\pi}_{l,h}$ yields the entropy of the strict left-context attention produced by head $\langle\, l, h \,\rangle$,

$$\underleftarrow{\mathrm{entr}}^{\pi}_{l,h}[\theta](\, w_i, w_{<i}\,) \coloneqq \mathrm{H}_b(\underleftarrow{\mathrm{Attn}}^{\pi}_{l,h}[\theta](\, w_i, w_{<i}\,)),$$

for $\pi \in \{\, \epsilon, \mathrm{N}, \text{RL-N} \,\}$.

Definition 5.37 (Scaled strict left-context attention entropy)**.** Given a transformer and an input sequence $w_1 \ldots w_n$, for any position $i \in \{\, 2, \ldots, n \,\}$ the function $\widetilde{\underleftarrow{\mathrm{entr}}}^{\pi}_{l,h}$ yields the entropy of the strict left-context attention produced by head $\langle\, l, h \,\rangle$ normalised by the maximum entropy possible for a context of length $i - 1$,

$$\widetilde{\underleftarrow{\mathrm{entr}}}^{\pi}_{l,h}[\theta](\, w_i, w_{<i}\,) \coloneqq \frac{\underleftarrow{\mathrm{entr}}^{\pi}_{l,h}[\theta](\, w_i, w_{<i})\,)}{\log_b(i-1)},$$

for $\pi \in \{\, \epsilon, \mathrm{N}, \text{RL-N} \,\}$.

Dividing the entropy at position i by the inverse of the maximally possible entropy $\log_b(i-1)$ for the strict left context yields a value in $[0,1]$. The motivation for this is the observation that regular

entropy is expected to grow as the size of the context increases. Given a uniform distribution, entropy is precisely determined by $\log_b(i-1)$.

To illustrate this point, assume that for a query at position i, $K_i \sim \mathrm{Binom}(i-1, p)$ with $p > 0$ of the context elements are expected to match in features. This description is justified in the limit when the number of features is finite. Let H_i denote strict left-context attention entropy for position i and let $b = \mathrm{e}$, without loss of generality. If attention were distributed uniformly across these matching elements (and set to 0 for the remaining elements as a simplification), the expected entropy would be

$$\mathbb{E}[H_i] = \mathbb{E}[\ln K_i]\,, \tag{117}$$

i.e. the expected entropy for a uniform distribution with K_i elements, since we can just ignore the remaining elements. Taking a second order Taylor expansion, we get

$$\begin{aligned}\mathbb{E}[\ln K_i] \approx \ln(\mathbb{E}[K_i]) - \frac{\mathrm{Var}[K_i]}{2\mathbb{E}[K_i]^2} &= \log(p(i-1)) - \frac{(i-1)p(1-p)}{2(p(i-1))^2} \\ &= \ln(p(i-1)) - \frac{1-p}{2p(i-1)}.\end{aligned} \tag{118}$$

Hence, for fixed $k > 0$, the entropy still grows logarithmically with context length. Dividing by $\ln(i-1)$ yields

$$\begin{aligned}\mathbb{E}\left[\frac{-\ln K_i}{\ln(i-1)}\right] &\approx \frac{\ln(k(i-1))}{\ln(i-1)} - \frac{1-p}{2p(i-1)\ln(i-1)} \\ &= \frac{\ln(p) + \ln(i-1)}{\ln(i-1)} - \frac{1-p}{2p(i-1)\ln(i-1)} = 1 + \frac{\ln(p)}{\ln(i-1)} - \frac{1-p}{2p(i-1)\ln(i-1)}\end{aligned} \tag{119}$$

which converges to 1 as $i \to \infty$ since both ratios vanish with $i \to \infty$. Thus, under the assumption that a constant proportion of prior tokens remains in principle attendable, normalisation by $\log_b(i-1)$ removes the trivial growth in entropy induced solely by increasing context size. Deviations of the scaled entropy from this limiting behaviour can then be interpreted as reflecting changes in the concentration of attention rather than changes in the number of available context positions.

Oh and Schuler (2022) report that normalisation to a proper distribution over the left context and scaling are necessary for convergence of their linear-mixed effects model for reading time prediction.

Finally, aggregated attention entropy is defined as the sum of attention entropies of a selection of heads and used to model reading times. This comes with the assumption that all heads of that selection have an equal share in processing difficulty (Oh and Schuler, 2022).

Definition 5.38 (Aggregated scaled strict left-context attention entropy)**.** Given a transformer, a set of layer-head indices S and an input sequence $w_1 \ldots w_n$, for any position $i \in \{2, \ldots, n\}$, the function $\widetilde{\overleftarrow{\mathrm{entr}}}^{\pi}_{\mathrm{agg}:S}$ is defined as

$$\widetilde{\overleftarrow{\mathrm{entr}}}^{\pi}_{\mathrm{agg}:S}[\theta](w_i, w_{<i}) \coloneqq \sum_{s \in S} \widetilde{\overleftarrow{\mathrm{entr}}}^{\pi}_{s}[\theta](w_i, w_{<i}).$$

for $\pi \in \{\epsilon, \mathrm{N}, \mathrm{RL\text{-}N}\}$.

Oh and Schuler (2022) compute aggregated attention entropy for all heads of the topmost layer of `GPT-2 small` since they assume that the computations immediately preceding the language modelling to be most indicative of the contribution of context tokens to the next-token prediction. Lower layers are not taken into account since they are thought to resemble retrieval to a lesser degree and perform other structural computations to prepare the retrieval in the topmost layer (Oh and Schuler, 2022). Unfortunately, this view is not investigated empirically.

Oh and Schuler (2022) found aggregated scaled entropy to substantially improve the prediction of human reading times across datasets and model specifications. When incorporated into regression models that already included strong baseline predictors and surprisal estimates from `GPT-2`, it consistently yielded increases in model likelihood, indicating that it captures variance in processing difficulty not explained by expectation-based measures alone. This confirms that attention entropy is predictive of RTs beyond a single attention head and that the heads in the topmost layer allow for a shared psycholinguistic interpretation.

Among the different formulations of attention weights, the variant based on norm-weighted attention (N) exhibited the strongest predictive performance. This suggests that incorporating the magnitude of transformed representations, in addition to attention scires, provides a more accurate characterization of the contribution of contextual tokens and, consequently, of the resulting entropy measure.

The strength of the predictor, however, differed across experimental paradigms. On the Natural Stories corpus of self-paced reading times, it showed a large and statistically significant effect, with effect sizes exceeding those of `GPT-2` surprisal. In contrast, on the Dundee eye-tracking corpus, the effect of attention entropy was unstable and did not reach statistical significance. This discrepancy suggests that attention entropy may be more closely related to processing components reflected in self-paced reading – such as sustained integration difficulty (see Section 3.3).

Correlation analyses by Oh and Schuler (2022) further revealed that attention entropy is only weakly correlated with contextual surprisal from `GPT-2`, indicating that it captures aspects of processing difficulty that are largely distinct from predictability. This supports the interpretation of attention entropy as reflecting mechanisms related to memory retrieval rather than probabilistic expectation.

Finally, Ryu and Lewis (2025) conduct an analysis using an aggregated form of restricted attention entropy (Ryu and Lewis, 2021, here: Definition 5.31). They compute an average over multiple attention heads; however note that statistical results are equivalent for summed and averaged measures due to their linear relation. The relevant difference with respect to Oh and Schuler (2022)'s entropy measure lies in how attention to the current item is disregarded.

Definition 5.39 (Local restricted attention entropy)**.** Given a transformer, a set of layer-head indices S and an input sequence $w_1 \ldots w_n$, for any position $\in \{2, \ldots, n\}$, the function $\mathrm{entr}^{\mathrm{restr}}_{\mathrm{loc}:S}$ yields the yields an average of the restricted attention entropy over the attention heads S, defined as

$$\mathrm{entr}^{\mathrm{restr}}_{\mathrm{loc}:S}[\theta](w_i, w_{<i}) := \frac{1}{|S|} \sum_{s \in S} \mathrm{entr}^{\mathrm{restr}}_{s}[\theta](w_i, w_{<i}).$$

Ryu and Lewis (2025) compare the measure calculated for the 20 most specialised `GPT-2 small` attention heads with respect to syntactic dependencies, determined by the method of Voita et al. (2019) outlined above, to an aggregate measure over all heads and to an aggregation over

non-specialised heads. They find that the inclusion of syntactically less important attention heads introduces noise which impedes psychometric predictive power.

Further evaluation on the Natural Stories and the GECO corpus using the top 20 heads confirms that attention entropy contributes predictive power above and beyond surprisal (Ryu and Lewis, 2025). More specifically, a positive current-word entropy effect is reported for self-paced reading times and for go-past times, while in first-fixation durations the entropy effect appears primarily at the previous word, i.e. as spillover, further supporting its role in later integrative memory processes.

Against the backdrop of these findings, Ryu and Lewis (2025) suggest a division of labour between the two measures: surprisal captures early processes related to probabilistic word identification, whereas attention entropy reflects later computation related to memory retrieval and integration. This implies that surprisal mainly takes on a facilitatory role for lexical access (Ryu and Lewis, 2025) as opposed to its interpretation as a causal bottleneck reflecting costs associated with a shift in resource allocation to sentence representations (see Section 4.2.1).

Overall, there is converging evidence that attention entropy captures a significant aspect of human sentence processing that is not explainable through the means of surprisal, and that – to the delight of any cognitive modeller – transformer language models provide a unified computational framework in which expectation-based and memory-based accounts are jointly realised (Ryu and Lewis, 2025). Furthermore, there is evidence that multiple transformer heads jointly contribute to prediction and that their effects should be combined when modelling processing difficulty. At the same time, it remains unclear why interference effects are a characteristic of only a specific selection of attention heads and to what extent this fact allows for a psycholinguistic interpretation.

Context windows Initial proposals investigating attention entropy, such as Ryu and Lewis (2021) and Oh and Schuler (2022), computed the measure using the full context window taken to compute surprisal, i.e. including the whole context available in the psycholinguistic dataset up to the maximum context length permissible by the transformer model. However, this appears to deviate from psycholinguistic theory which generally assumes that surface forms are only available in memory for a short duration, with items further away being abstracted into representations of meaning and discourse (Sachs, 1967).

By computing attention entropy with various context sizes, Ryu and Lewis (2025) find that a limited context window of 30 yields the strongest fit to human reading times. This might provide evidence for traditional psycholinguistic assumptions. On the other hand, since Ryu and Lewis (2025) do not scale their attention entropy measure by maximum entropy, it might also be an indication of degradation of the predictive power of entropy caused by trivial context-size induced effects.

Additionally, the method proposed by Ryu and Lewis (2025) requires performing transformer inference twice with different context sizes – once with a large context for generating surprisal and once with a small context to extract attention entropy. Ideally, a joint model, as Ryu and Lewis (2025) claim transformers to be, should account for both dimensions in parallel.

5.4.2.4 Multi-head attention and parallel retrieval

A key architectural issue concerns the number of retrieval operations. Classical cue-based models typically assume a single retrieval per integration event, guided by multiple cues (Section 4.3.2). In contrast, transformers compute several attention distributions in parallel, through multi-head attention, and in succession, by employing stacked attention layers.

This can be interpreted as multiple simultaneous retrieval operations in distinct learned subspaces. As attention heads appear to be specialised for different aspects of processing, as discussed above, retrieval might be decomposed by feature type. The finding that attention entropy aggregated over subsets of heads remains predictive (Ryu and Lewis, 2025) supports this interpretation.

At the same time, it remains an open question whether the aggregate behaviour of multi-head attention corresponds to a cognitively plausible retrieval process or whether individual heads perform heterogeneous computations not directly analogous to memory retrieval. Interestingly, Timkey and Linzen (2023) show that strongly constrained models with a single attention head can capture interference phenomena, suggesting that multiple heads are not strictly necessary to capute memory-effects.

5.4.2.5 Tokenisation

Due to tokenisation, it is non-trivial to align attention-based measures such as entropy with human reading times, which are typically provided for white-space separated regions in psycholinguistic datasets. In contrast to surprisal (see Section 4.2.1), where the chain rule of surprisal yields a well-defined expression for the surprisal of the concatenated region, the aggregation of attention-based measures over subword tokens constitutes an explicit modelling choice that introduces additional assumptions.

One option is to sum attention-based quantities over all subword tokens corresponding to a word. Under this approach, each subword contributes independently to the total processing cost, implying that words consisting of more tokens systematically incur higher costs. This is the strategy adopted by Oh and Schuler (2022). It effectively treats each subword as inducing a separate retrieval or integration operation. However, this might overestimate processing difficulty for morphologically complex or rare words that are split into many tokens.

Another option is to take the mean over subword-level values. This removes the direct dependence on the number of tokens but it assumes that on average, subwords of a multi-token word trigger retrieval operations that are structurally similar to retrieval for the whole word. Assuming that certain parts of words are more important in determining their relation to contexts than others, this can potentially underestimate actual costs, reducing spikes in entropy for individual subwords (Ryu and Lewis, 2025).

A third option is to take the maximum over subword-level values. While this avoids length effects, it presupposes that processing is governed by a single peak of difficulty rather than by the joint contribution of multiple items. This strategy, introduced by Ryu and Lewis (2025), is to be seen rather as a heuristic middle way between summation and averaging strategy. There is currently no consensus on which assumption is most appropriate.

A further complication arises from the fact that the memory representation itself is token-based. Transformers do not construct higher-level symbolic constituents over words, as is assumed in

many parsing-based models, but instead maintain a sequence of token-level representations. Under a psycholinguistic interpretation, this implies that subword units may be retrieved independently and that multiple subwords belonging to the same word can compete during retrieval. Consequently, splitting a word into multiple tokens increases the number of potential competitors and may inflate measures such as attention entropy.

This observation highlights that tokenisation is not merely a preprocessing step but directly affects the structure of the model's memory and the resulting complexity measures. In particular, more fine-grained tokenisation schemes increase the effective context length and thereby also attention entropy, unless explicitly normalised for context size.

To the best of my knowledge, there has been no systematic investigation of the influence of vocabulary size and tokenisation strategy on attention-based complexity measures such as entropy. Given that many psycholinguistic theories assume word-based memory representations, a natural first step to control for these effects would be to evaluate models operating on word-level tokenisation. Such models would allow disentangling effects of attention structure from artefacts introduced by subword segmentation.

5.4.2.6 Distance-based measures

Beyond entropy-based measures of attentional diffuseness, recent work has introduced predictors that capture the dynamics of attention across time, i.e. how attention distributions change from one word to the next (Oh and Schuler, 2022). These measures are motivated by the idea that processing difficulty may arise not only from uncertainty about retrieval targets, but also from the need to reallocate attention as new input is processed. This reintroduces performance constraints which were abstracted away by treating memory as a constant-time content-addressable system (see Section 4.3.2).

The following predictors are aggregated by Oh and Schuler (2022) across the attention heads in the topmost transformer layer in the same fashion as defined for attention entropy in Definition 5.38.

Change in attention diffuseness First, Oh and Schuler (2022) propose the absolute change in entropy across timesteps as a contributor to reading times.[28]

Definition 5.40 (Delta entropy)**.** Given a transformer with attention head $\langle\, l, h \,\rangle$ and an input sequence $w_1 \ldots w_n$, for any position $i \in \{\, 2, \ldots, n \,\}$ the function $\Delta_{\widetilde{\overleftarrow{\text{entr}}}^{\,\pi}_{l,h}}$ yields the change in scaled strict left-context entropy between positions $i-1$ and i, defined as

$$\Delta_{\widetilde{\overleftarrow{\text{entr}}}^{\,\pi}_{l,h}}[\theta](\, w_i, w_{<i} \,) \coloneqq \left| \widetilde{\overleftarrow{\text{entr}}}^{\,\pi}_{l,h}[\theta](\, w_i, w_{<i} \,) - \widetilde{\overleftarrow{\text{entr}}}^{\,\pi}_{l,h}[\theta](\, w_i, w_{<i-1} \,) \right|.$$

for $\pi \in \{\, \epsilon, \text{N}, \text{RL-N} \,\}$.

This measure is related to the *Entropy Reduction* hypothesis (Section 4.2.2) but differs in two important respects. First, it is defined over attention distributions rather than over distributions of

[28] The authors do not make explicit how they quantify delta entropy for the second token in the sequence. A justifiable option would be to define entropy as 0 for the empty left context, thus making entropy of step 2 also the delta entropy for step 2.

syntactic analyses, similar to the more recent proposals of entropy reduction over lexical distributions (e.g. Frank, 2013). Secondly, in contrast to classical entropy reduction (Hale, 2006), it assigns processing cost to both increases and decreases in entropy, reflecting the intuition that any substantial reorganisation of the retrieval distribution may incur cost.

Importantly, $\Delta_{\underset{\sim}{\text{entr}}}$ is insensitive to how attention mass is redistributed among individual tokens and only captures changes in overall diffuseness. As such, it groups together qualitatively distinct reallocation patterns that preserve entropy.

Empirically, Oh and Schuler (2022) find that this measure provides inconsistent improvements across datasets, suggesting that changes in uncertainty alone may not fully capture the relevant dynamics of processing difficulty.

Manhattan distance (MD) A more fine-grained measure is given by the *Manhattan distance* (MD) between consecutive attention distributions (Oh and Schuler, 2022):

Definition 5.41 (Manhattan distance)**.** Given a transformer with attention head $\langle\, l, h \,\rangle$ and an input sequence $w_1 \ldots w_n$, for any position $i \in \{\, 2, \ldots, n \,\}$, the function $\text{MD}^{\pi}_{l,h}$ yields the L^1-norm of the point-wise difference between the attention distributions for positions i and $i-1$:

$$\text{MD}^{\pi}_{l,h}[\theta](\, w_i, w_{<i} \,) \coloneqq \left\| \text{Attn}^{\pi}_{l,h}[\theta](\, w_i, w_{<i} \,) - \left(\text{Attn}^{\pi}_{l,h}[\theta](\, w_i, w_{<i-1} \,) \circ \begin{pmatrix} 0 \end{pmatrix}\right) \right\|_1.$$

The concatenation of a single 0 to the attention scores of the preceding processing step ensures that both vectors lie in $\mathbb{R}^i$ and can be compared (Oh and Schuler, 2022). Note that in contrast to attention entropy, this measure does not factor out attention to the current token.

MD measures the total magnitude of change in attention weights across positions. For distributions, it is equal to twice the amount of mass that must be redistributed. Take for instance $\langle\, 1, 0, 0 \,\rangle$ and $\langle\, 0, 1, 0 \,\rangle$. MD would amount to 2, as both positions 1 and 2 experience a change of of 1.

The measure is purely reflective of the amount of change between distributions and not of its structure. It is symmetric and insensitive to the linear distance between tokens: shifting attention from position j to k incurs the same cost regardless of their distance $|j-k|$. This is inconsistent with a DLT account of memory distance.

Empirically, Oh and Schuler (2022) report Manhattan distance to emerge as the strongest and most robust distance-based predictor. MD significantly improves model fit over surprisal-based baselines across both self-paced reading and eye-tracking data.

Earth mover's distance The third distance-based measure proposed by Oh and Schuler (2022) computes the *Earth mover's distance* (EMD) between consecutive attention distributions. This metric quantifies the minimal cost required to transform one distribution into another by movings mass across positions, determined by both the amount of weight shifted and the distance this weight traverses. The following characterisation of EMD is based on Rubner et al. (2000).

Definition 5.42 (Work)**.** The function Work is defined over all pairs of matrices of equal shape $\mathbf{D}, \mathbf{F}$ as follows

$$\text{Work}(\mathbf{D}, \mathbf{F}) \coloneqq \mathbf{1}^{\mathsf{T}}(\mathbf{D} \odot \mathbf{F})\mathbf{1},$$

where $\mathbf{1}$ is a vector where each entry is 1. $\mathbf{D}$ is interpreted as a matrix of distances between pairs and $\mathbf{F}$ as a matrix of weight flows.

Definition 5.43 (Earth mover's distance)**.** The function EMD is defined over all vectors $\mathbf{x}, \mathbf{y}$ and matrices $matrD$ with shape $|\mathbf{x}| \times |\mathbf{y}|$ as

$$\mathrm{EMD}(\mathbf{x}, \mathbf{y}, \mathbf{D}) \coloneqq \frac{\mathbf{1}^\mathsf{T}(\mathbf{D} \odot \mathbf{F}^*)\mathbf{1}}{\mathbf{1}^\mathsf{T}\mathbf{F}^*\mathbf{1}},$$

where $\mathbf{F}^*$ is an *optimal flow plan* minimising costs determined by

$$\mathbf{F}^* \coloneqq \underset{\mathbf{F} \in \mathbb{R}^{|\mathbf{x}| \times |\mathbf{y}|}}{\arg\min}\ \mathrm{Work}(\mathbf{D}, \mathbf{F}).$$

subject to constraints

$$\begin{aligned}
\mathbf{F}_{(i,j)} &\geq 0 && \forall i \in [|\mathbf{x}|], j \in [|\mathbf{y}|], \\
(\mathbf{F}\mathbf{1})_{(i)} &\leq \mathbf{x}_{(i)} && \forall i \in [|\mathbf{x}|], \\
(\mathbf{1}^\mathsf{T}\mathbf{F})_{(j)} &\leq \mathbf{y}_{(j)} && \forall j \in [|\mathbf{y}|], \\
\mathbf{1}^\mathsf{T}\mathbf{F}\mathbf{1} &= \min(\{\, \mathbf{1}^\mathsf{T}\mathbf{x}, \mathbf{1}^\mathsf{T}\mathbf{y} \,\}).
\end{aligned}$$

Following Oh and Schuler (2022), the solution $\mathbf{F}^*$ to the linear optimisation problem can be efficiently found using the transportation simplex method (Dantzig, 1951).

Definition 5.44 (Earth mover's attention distance)**.** Given a transformer with attention head $\langle\, l, h \,\rangle$ and an input sequence $w_1 \ldots w_n$, for any position $i \in \{\, 2, \ldots, n \,\}$, the function $\widetilde{\mathrm{EMD}}^{\pi}_{l,h}$ is defined as

$$\widetilde{\mathrm{EMD}}^{\pi}_{l,h}[\theta](\, w_i, w_{<i} \,) \coloneqq \mathrm{EMD}(\mathrm{Attn}^{\pi}_{l,h}[\theta](\, w_i \,), w_{<i}), \mathrm{Attn}^{\pi}_{l,h}[\theta](\, w_i, w_{<i-1} \,), \mathbf{D}),$$

where the distance matrix $\mathbf{D}$ is determined by the surface distance between position pairs scaled by context length (Oh and Schuler, 2022),

$$\mathbf{D}_{(j,k)} \coloneqq \frac{|j - k|}{i - 1}.$$

Similar to scaled attention entropy, the scaling factor $i - 1$ is determined by the maximum EMD value possible for a context size of $i - 1$, achieveable if the leftmost element in context is attended to with weight 1 in the preceding step and the rightmost one in the current, as the optimal flow plan transfers a weight of 1 across $i - 1$ positions.

Intuitively, EMD measures not only how much attention mass is moved, but also how far it is moved. Reallocating attention to distant tokens incurs higher cost than reallocating it locally. In this sense, EMD is closely related to locality-based theories such as Dependency Locality Theory (Section 4.3.1), although it differs in that distance is defined relative to the previous state focus in attention rather than the current input position. Thus, EMD might be low in the case of two successive long-range integrative processes if the integration sites are close to each other. A potential interpretation would be that of a smooth pointer that is moved across memory during sentence processing.

Empirically, however, Oh and Schuler (2022) find EMD to perform less consistently than MD and to fluctuate significantly with respect to the choice of attention distribution function ($\underset{\sim}{\widetilde{\mathrm{entr}}}, \underset{\sim}{\widetilde{\mathrm{entr}}}^{\mathrm{N}}, \underset{\sim}{\widetilde{\mathrm{entr}}}^{\mathrm{RL\text{-}N}}$), This suggests that linear token distance may not be the most appropriate metric for capturing attention reallocation.

Overall, Oh and Schuler (2022) made an important contribution towards deriving additional predictors from attention weights compatible with a retrieval framework. Further refinements and replications are necessary to assert the plausibility of effects of this kind.

6 Modelling reading times with linear mixed-effects models

A central goal of psycholinguistics is to identify the cognitive mechanisms that underlie human sentence processing. Since these mechanisms are not directly observable, they are typically studied via behavioural measures such as reading times. These measures are assumed to reflect processing difficulty at each word, albeit in a noisy and indirect manner.

The present work adopts the view that computational predictors derived from language models, such as surprisal or attention entropy, can serve as explicit hypotheses about the sources of processing difficulty. Fitting these predictors to reading times therefore serves a dual purpose. First, it evaluates their predictive adequacy: if a predictor explains systematic variance in reading times beyond standard controls, then the corresponding computational quantity captures an empirical regularity that is relevant for human reading behaviour. Second, it allows theoretically motivated comparisons between candidate sources of processing difficulty. For example, an effect of surprisal supports the relevance of expectation-based difficulty, whereas an additional effect of attention entropy would suggest that uncertainty or competition in retrieval-like operations captures variance not reducible to lexical predictability alone.

The empirical modelling approach pursued here is therefore not merely descriptive. Rather, it tests whether specific quantities extracted from neural language models can function as cognitively plausible linking hypotheses between linguistic input and observed reading behaviour. Positive results do not imply that the human parser implements the same algorithm as the language model. However, they constrain psycholinguistic theory by identifying dimensions of processing difficulty that require explanation and by suggesting computational mechanisms that may account for them.

6.1 Hypothesis testing, p-values and error rates

Before discussing how to model reading times, it is necessary to qualify a few preliminary concepts.

When modelling some proposed effect using measurements, e.g when estimating a regression coefficient, any observable relationship is not meaningful in isolation. Its interpretation depends on how much variation would be expected under a scenario in which no systematic relationship is present. Therefore, statistical modelling is usually performed using a *null-hypothesis*. It provides a formally specified baseline against which empirical evidence can be evaluated. Intuitively, it represents the assumption that a predictor has no effect on the dependent variable (Casella and Berger, 2002, Chapter 8).

Definition 6.1 (Null hypothesis)**.** Let Θ denote the parameter space of a statistical model. A *null hypothesis* H_0 is a subset of Θ that specifies a restricted class of parameter values, typically corresponding to the absence of an effect. The *alternative hypothesis* H_1 is the complement of H_0 with respect to the parameter space.

$$H_1 = \Theta \setminus H_0.$$

Statistical inference evaluates whether the observed data provide sufficient evidence to reject H_0 in favour of H_1 (Casella and Berger, 2002, Chapter 8).

In the following, I will assume (without loss of generality) a simple null-hypothesis containing only one parameter value $H_0 = \{ \theta_0 \}$.

Statistical testing follows an asymmetric logic. The null hypothesis represents a baseline or scientific status quo, and the burden of proof lies with the researcher to provide that data is sufficiently incompatible with it. This asymmetry is closely related to the notion of falsification in the philosophy of science (Popper, 2005).

A test is based on observed data and thus can only provide evidence against H_0 by rejecting it, or fail to reject it if the data is not sufficiently incompatible with the null hypothesis. The latter outcome does not imply that H_0 is true, but merely that the available data does not provide sufficient evidence against it (Wasserstein and Lazar, 2016).

Conversely, rejecting the null hypothesis proves neither that the scientific interpretation is true nor that the fitted model captures all relevant structure in the data. As Box (1976) emphasises, statistical models are idealisations rather than literal descriptions of reality. A rejection of H_0 therefore has to be interpreted relative to the statistical model under consideration: its parameter space, distributional assumptions and chosen set of independent variables. In particular, a different model specification or parameterisation might provide an equally plausible or more useful account of the observed data.

6.1.1 Statistical test, p-value

The idea of hypothesis testing is to partition the sample space into a region that allows to reject the null-hypothesis and a region that is consistent with the null-hypothesis (Casella and Berger, 2002, Chapter 8). This is specified using a *test statistic*.

Definition 6.2. Given a sample space $\mathbb{X}$, a *test statistic* $t_n \colon \mathbb{X} \to \mathbb{R}$ is a measurable function of the data $\mathbf{X} \subseteq \mathbb{X}$ for any sample size $n = |\mathbf{X}|$.

Let $P[\theta]$ for $\theta \in \Theta$ be a statistical model. The test statistic induces a sampling distribution via

$$T_n = t_n(X_1, \ldots, X_n), \qquad X_i \sim P[\theta]. \tag{120}$$

This allows the computation of the *p-value*,

$$p = P[\theta_0](\, T_n \geq t_n(\mathbf{X}) \,), \tag{121}$$

where $t_n(\mathbf{X})$ is the observed value of the test statistic and $n = |\mathbf{X}|$ (Wasserman, 2004). In practice, the sampling distribution of T_n is usually not available in closed form and must be approximated.

The p-value quantifies how likely the observed test statistic (or more extreme outcomes) are under the assumption that H_0 is true, i.e. no effect being present (Wasserman, 2004). The smaller the p-value, the less likely it is that the data originated from a generative process without the effect being tested. If the observed data would be highly unlikely under H_0, then H_0 is considered implausible.

To obtain a decision, a significance level $c \in (0,1)$ is chosen in advance. The null hypothesis is rejected if $p \leq c$. This rule defines when the evidence against H_0 is considered sufficiently strong to warrant rejection.

Definition 6.3 (Test function)**.** Given $\theta_0 \in \Theta$ one can define a *test function* on the sample space such that,

$$\iota_{c,\theta_0}(t_n(\mathbf{X})) \coloneqq \begin{cases} 1 & \text{if } p \leq c, \\ 0 & \text{else,} \end{cases}$$

indicating the rejection of the null-hypothesis.

The test function partitions the sample space into a rejection region and an acceptance region (Casella and Berger, 2002, Chapter 8). The outcome of the test depends on which region the observed data fall into.

Note that the p-value does not represent the probability that the null hypothesis is true (Wasserstein and Lazar, 2016). Rather, it quantifies how probable a result at least as extreme as the observed one would be if H_0 (and the statistical model) were correct. A small p-value therefore provides evidence against H_0, but not proof.

6.1.2 Type-I error, Type-II error and power

This decision procedure gives rise to two types of errors. A *Type-I error* occurs if H_0 is rejected although it is true, i.e. when the observed data $\mathbf{X}$ were generated according to $\mathbf{X} \sim P[\theta_0]$ (Wasserman, 2004).

Generally, for a simple null hypothesis $H_0 = \{\, \theta_0 \,\}$, the Type-I error rate is a property of the test procedure, defined as the probability of such an event under repeated sampling from the null model. This is given by

$$\alpha_n = P(\iota_{c,\theta_0}(X_1, \ldots, X_n) = 1), \qquad X_i \sim P[\theta_0]. \tag{122}$$

A *Type-II error* occurs if H_0 is not rejected although the alternative hypothesis is true. Since the alternative hypothesis is generally composite, containing more than one option, the Type-II error rate depends on the parameter value $\theta \in H_1$:

$$\beta_n(\theta) = P(\iota_{c,\theta_0}(X_1, \ldots, X_n) = 0), \qquad X_i \sim P[\theta], \qquad \theta \in H_1. \tag{123}$$

The quantity $1 - \beta_n(\theta)$ is called *statistical power*. It is the probability of correctly rejecting the null hypothesis when the alternative is true (Wasserman, 2004).

For confirmatory analyses, ideally one would like both α_n and $\beta_n(\theta)$ to be small. However, there is an inherent trade-off between both error types controlled by the significance level c. Model specification should control the Type-I error rate while retaining sufficient power (Barr et al., 2013; Matuschek et al., 2017). Increasing the sample size reduces the variability of the test statistic and generally leads to a decrease in β_n, thereby increasing statistical power. (Wasserman, 2004)

6.1.2.1 Control of Type-I error

A central property of the decision rule based on the p-value is that it controls the probability of false rejections. Under suitable regularity conditions, the test can be constructed such that the probability of a false rejection is smaller or equal to the desired significance threshold c (Vasishth et al., 2022),

$$\alpha_n = P(\iota_{c,\theta_0}(X_1, \ldots, X_n) = 1) = P(p \leq c) \leq c, \quad X_i \sim P[\theta_0], \quad \forall\, n. \tag{124}$$

This follows from the fact that p-values are uniformly distributed on $[0, 1]$ if the test statistic is continuous.[29] Consequently, if the decision rule $p \leq c$ is repeatedly applied to new samples

[29] Under H_0, it holds that $p = 1 - F(T)$ with F being the cumulative distribution function of the test statistic T. Now, if F is continuous, it is a standard result that $F(T)$ is uniformly distributed on $[0, 1]$ which implies that the same holds for p (Wasserman, 2004)

generated under H_0, a proportion of at most c of all tests will yield a false positive.

Thus, the significance level c serves as a user-specified upper bound on the Type-I error rate. However, the guarantee holds only if the assumed sampling distribution of the test statistic $T_n = t_n(X_1, \dots, X_n)$, with $X_i \sim P[\theta_0]$ is accurate (Casella and Berger, 2002). Usually, this sampling distribution is approximated – for instance via a standard normal distribution for the sample mean, justified by the Central Limit Theorem. The validity of the test therefore depends on the quality of this approximation, which may be affected by sample size and distributional assumptions. If the approximation is inaccurate, the actual Type-I error rate α_n may deviate from c, so that the guarantee $\alpha_n \leq c$ no longer holds.

Since the p-value depends on the sampling distribution of the test statistic under H_0, different choices of the test statistic induce different distributions and therefore require different assumptions for their valid approximation to ensure that this distribution is correctly specified or well approximated.

6.2 Linear regression

Linear regression is one of the most fundamental statistical models and enjoys an established role in every psycholinguists' toolset (Just et al., 1982). It models the relationship between a dependent variable and one or more predictors by assuming that the response (i.e. the observed values of the dependent variable) can be expressed as a linear combination of these predictors in addition to random noise.

Assume a series of n observations. For each observation $i \in [n]$, the model takes the form

$$y_i = \beta_1 \cdot x_{i,1} + \cdots + \beta_p \cdot x_{i,p} + \beta_{p+1} + \varepsilon_i, \tag{125}$$

where $x_{i,k}$ denotes the value of the k-th predictor for observation i, β_j with $j \in [p]$ are regression coefficients, β_{p+1} is the intercept and ε_i is a residual error term (Gelman and Hill, 2006, Chapter 1A).

This equation states that the response y_i is obtained by combining the predictor values $x_{i,j}$ with weights β_j, the intercept β_{p+1}, plus a noise term that captures variation not explained by the predictors.

Example 6.1. For example, a regression model for reading times may take the form

$$\mathrm{RT}_i = \beta_1 \cdot \mathrm{frequency}_i + \beta_2 \cdot \mathrm{length}_i + \beta_3 + \varepsilon_i. \tag{126}$$

Here, β_1 describes how reading time changes with word frequency, and β_2 describes the effect of word length. Each coefficient quantifies the expected change in reading time associated with a one-unit change in the corresponding predictor, holding the others constant. β_3 determines the base reading times for each word. In self-paced reading times, for instance, this might include some constant time it takes to press the button to continue on to the next word.

For notational convenience, the model can be written compactly in matrix form as

$$\mathbf{y} = \mathbf{X}_{/\mathbf{1}}\, \boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{127}$$

where $\mathbf{y} \in \mathbb{R}^n$ is the *response vector* collecting all observed values of the dependent variable, $\boldsymbol{\beta} \in \mathbb{R}^{p+1}$ is the vector of regression coefficients, including the intercept as its last element, and $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is the vector of residual errors. The so-called *design matrix* $\mathbf{X} \in \mathbb{R}^{n \times p}$ contains the predictor values (Everitt, 2002). Its i-th row corresponds to observation i and contains $x_{i,1}, \ldots, x_{i,p}$. With this notation, the matrix product $\hat{\mathbf{y}} = \mathbf{X}_{/\mathbf{1}}\boldsymbol{\beta}$ produces the linear combination of predictors for each observation.

Structurally, weights $\mathbf{W} \in \mathbb{R}^{p,m}$ in feed-forward networks, introduced in Section 5.3.2.7, are an extension of linear regression where a series of parameters $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_m$ are used to model m response vectors simultaneously.

6.2.1 Assumptions

To make statistical inference possible, the error terms are commonly assumed to follow a normal distribution,

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}), \tag{128}$$

where $\mathbf{I}$ is the identity matrix (a square matrix with ones on the main diagonal and zeros elsewhere) (Gelman and Hill, 2006, Chapter 1A). This implies that the error terms are independent and identically distributed with constant variance σ^2.

This further implies that the response vector $\mathbf{y}$ conditional on the design matrix $\mathbf{X}$ is normally distributed:

$$\mathbf{y} \mid \mathbf{X} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}). \tag{129}$$

At the level of individual observations, this means that each response is normally distributed around its expected value $\mathbf{x}_i^\top\boldsymbol{\beta}$ with constant variance σ^2.

In the following, I will shortly discuss the classical assumptions for statistical inference in linear modelling, including linearity of the conditional mean function as well as the distributional assumptions (Equation 128) and how to diagnose their violation. These strategies hold equally for linear mixed models, which will be introduced in Section 6.3.

6.2.1.1 Linearity and correct specification

Not mentioned yet, but perhaps the most important assumptions is that of a linear relationship between predictors and response. Violating this assumption can lead to biased estimates, poor predictions and misleading inferences (Poole and O'farrell, 1971).

To assess whether the conditional mean structure assumed by the model is appropriate, one can examine a plot of the observed *residuals* $\hat{\varepsilon}_i = y_i - \hat{y}_i$ against the fitted values $\hat{y}_i$ (*residual plot*). Ideally, the residuals should be randomly scattered around zero without any discernible pattern. Pattern such as curves suggest that the model is systematically underfitting or overfitting the response and that a model with a different order of growth is needed (Gelman and Hill, 2006, Chapter 1A). Similarly, plotting residuals against individual predictors can reveal whether the effect of a predictor is nonlinear.

If nonlinearity is detected, one can either transform predictors or response variables (e.g. log, square root) or use more flexible models like polynomial regression or splines (Gelman and Hill,

2006, Chapter 1A). For example, if residuals show a quadratic trend, it may indicate that a polynomial term (e.g. x^2) should be added to the model. Note that this might have the potential downside of decreasing the model's interpretability.

6.2.1.2 Homoscedasticity (constant variance)

Homoscedasticity assumes that the variance of the residuals is constant across all levels of the predicted values (Poole and O'farrell, 1971). To evaluate this, one can also use a residual plot and look for a "funnel" shape or any systematic change in spread (Vasishth et al., 2022). If the dispersal of residuals increases or decreases with the fitted values, *heteroscedasticity* is present since the variance of the errors depends on the magnitude of the response.

Remedies again include transforming variables (e.g. log transformation of the response for multiplicative heteroscedasticity) or using weighted least squares (Poole and O'farrell, 1971).

6.2.1.3 Normality of residuals

The assumption of normally distributed residuals is important since the procedure for significance testing assumes normal residuals (Vasishth et al., 2022). One can use a *quantile-quantile* (Q-Q) *plot* to check this, which compares the quantiles of the standardised residuals to the quantiles of a standard normal distribution. If the residuals are normally distributed, the points in the Q-Q plot should lie approximately along a straight 45-degree line. Systematic deviations from this line (e.g. S-shaped curves or heavy tails) indicate non-normality (Vasishth et al., 2022). For example, a heavy right tail suggests right-skewed residuals. If non-normality is detected, a transformation or alternative modelling approach may be considered (Vasishth et al., 2022).

6.2.1.4 Independence of errors

The assumption of *independence* posits that the error terms are not correlated. Violations can pose problems for statistical inference (Hughes and Fisher, 2025; Vasishth et al., 2022). Note that this condition does not concern the residuals which are generally correlated, as they depend on the some set of common estimated coefficients.

Importantly, independence is primarily justified by the data collection process (Hughes and Fisher, 2025). If observations correspond to distinct experimental units sampled independently, the assumption is generally plausible.

6.2.2 Estimation

The coefficients $\boldsymbol{\beta}$ are estimated to minimise prediction error for the observed data, commonly via *ordinary least squares* (OLS) (Gelman and Hill, 2006; Everitt, 2002). In the case that the errors are independent and identically distributed as $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, the OLS estimator coincides with the maximum likelihood estimator of $\boldsymbol{\beta}$ (Gelman and Hill, 2006, Chapter 1A).

Given that the errors ϵ_i are independent and identically distributed as $\epsilon \sim \mathcal{N}(0, \sigma^2)$, the response y_i is normally distributed with mean $\mu_i = \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p} + \beta_{p+1}$ and variance σ^2, i.e. $y_i \sim \mathcal{N}(\mu_i, \sigma^2)$. Thus, the likelihood function of the observed data is the product of the individual likelihoods for all observations, determined by the probability density function of the normal

distribution (Casella and Berger, 2002, Section 11.3):

$$\ell(\boldsymbol{\beta}, \sigma^2 \mid \mathbf{X}, \mathbf{y}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\mathbf{y}_{(i)} - \mu_i)^2}{2\sigma^2}\right). \tag{130}$$

Maximum likelihood estimation seeks the values of $\boldsymbol{\beta}$ and σ^2 that maximise this likelihood. For fixed σ^2, this is equivalent to minimising residual error.

This work will omit a more detailed exposition of the implementation of estimation techniques and refers readers to (Gelman and Hill, 2006). Potential practical issues in the optimisation process such as convergence or singularity problems will be discussed in Section 6.3.6 since solutions are partly specific to the linear mixed-effects architectures applied in this work.

6.2.3 Why ordinary linear regression is insufficient

Reading-time datasets are characterised by repeated measurements. A single participant contributes many observations, and the same linguistic items is observed across participants. As a consequence, observations are not independent in the sense required by ordinary linear regression and statistical inference might be compromised (see Section 6.2.1.4; Clark, 1973; Baayen et al., 2008).

Let's make this more precise: suppose that reading time is analysed using a simple linear regression. Then one assumes that the error terms ε_i are mutually independent and identically distributed. This independence assumption is implausible for reading-time data. If the same participant is generally slower than another participant, then all observations from that participant tend to be shifted upwards. Likewise, some words or sentences may be systematically harder than others for reasons not fully captured by the observed predictors. Ignoring such dependencies leads to an overestimation of the effective sample size. As a consequence, the variance of the parameter estimates is underestimated. This inflates test statistics and yields overly optimistic (anti-conservative) inference (Baayen et al., 2008; Barr et al., 2013).

6.3 From linear regression to linear mixed-effects models

In this section, I will introduce an extension to the framework of linear regression called *linear mixed-effects models* (LMEMs), which allows statistical inference using data of repeated measurements. First, it is necessary to introduce the concepts of *fixed* and *random effects*.

6.3.1 Fixed effects

Fixed effects are coefficients that are assumed to be constant across the population under study (Pinheiro and Bates, 2000). They represent systematic effects of predictors such as word frequency, word length, surprisal or an attention-based entropy on reading time.

The coefficients in standard linear regression (see Section 6.2) are fixed effects. If x is a predictor, then a fixed effect β_x means that the expected effect of x is described by one single coefficient shared across all participants and all observations, i.e. $\beta_x \cdot x_i$ for all responses y_i. The term *effect* is used because the model describes deviations from the population-level mean (Pinheiro and Bates,

2000). Assuming that the predictor was centered, a value of k for the predictor has the *effect* of shifting the dependent variable by $k \cdot \beta_x$.

6.3.2 Random effects

Random effects extend the linear model by allowing selected coefficients to vary across levels of a grouping factor such as participants or lab locations (Pinheiro and Bates, 2000; Baayen et al., 2008; Goldstein, 1995). The underlying idea is that the participants that took part in the study are sampled from larger populations, and the model should generalise beyond the specific sample observed. In contrast to the linear regression model, where each observation i is treated as independent, certain groups of observations are now treated as correlated. To make this explicit, let i index observations and let $f(i)$ map observations to the respective participant they were collected from. First, a *random-intercept* model with p participants as grouping factor can be written as

$$y_i = \beta_1 x_i + \beta_2 + b_{f(i)} + \varepsilon_i, \tag{131}$$

Here, β_2 is the population-level intercept and $b_{f(i)}$ is the participant-specific deviation from that intercept. Observations from the same participant are correlated because they share the same random intercept.

More generally, one may also allow slopes to vary per participant:

$$y_i = (\beta_1 + b_{1,f(i)})x_i + (\beta_2 + b_{2,f(i)}) + \varepsilon_i. \tag{132}$$

In this case, participant $f(i)$ has both an individual baseline reading speed and an individual sensitivity to predictor x (*random coefficient*).

These effects are called *random* due to the fact that different instantiations of the same experiment exhibit differing setups of subjects and thus, different participant-wise coefficients/intercepts (Barr et al., 2013). This stands in contrast to fixed effects which are assumed to be constant across experiments.

6.3.3 Definition

A *linear mixed-effects model* (LMEM) (Fisher, 1919) combines fixed and random effects in a single hierarchical specification. They are also called *hierarchical models* (Lindley and Smith, 1972) or *multilevel models* (Goldstein, 1995). The following exposition follows the introduction given by Baayen et al. (2008); readers are referred to it for more details.

Definition 6.4 (Linear mixed-effects model)**.** A linear mixed-effects model is specified by the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\varepsilon},$$

where $\mathbf{y} \in \mathbb{R}^n$ is the response vector, $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the fixed-effects design matrix, $\boldsymbol{\beta} \in \mathbb{R}^p$ is the vector of fixed-effect coefficients, $\mathbf{Z} \in \mathbb{R}^{n \times q}$ is the random-effects design matrix, $\mathbf{b} \in \mathbb{R}^q$ is the random-effects vector, and $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is the residual error term.

The notation used in the preceding section introduced random effects with two indices, where $b_{k,f(i)}$ denotes the deviation associated with predictor k for participant $f(i)$. In the general formulation of linear mixed-effects models, all such random effects are collected into a single vector $\mathbf{b}$.

Concretely, the vector $\mathbf{b}$ is obtained by stacking all participant-specific effects (e.g. all intercepts and slopes) into one long vector.

The design matrix $\mathbf{Z}$ specifies how observations are associated with random effects. Each row of $\mathbf{Z}$ corresponds to an observation and selects the relevant random-effect coefficients from $\mathbf{b}$ (Baayen et al., 2008). For example, in a random-intercept model, $\mathbf{Z}$ contains indicator variables that assign each observation to its corresponding participant. If random coefficients are included, the entries of $\mathbf{Z}$ additionally contain predictor values, duplicated from $\mathbf{X}$, so that each observation is associated with the appropriate participant-specific slope contained in $\mathbf{b}$.

In many applications, random effects arise from multiple *grouping factors*, such as participants and items. In this case, the random-effects vector $\mathbf{b}$ can be decomposed into subvectors corresponding to the different grouping factors. For instance, for two grouping factors one may write $\mathbf{b} = \mathbf{b}^{(1)} \circ \mathbf{b}^{(2)}$ where $\mathbf{b}^{(1)}$ and $\mathbf{b}^{(2)}$ collect the random effects associated with the first and second grouping factor, respectively. Correspondingly, the random-effects design matrix can be written as a horizontal concatenation $\mathbf{Z} = \mathbf{Z}^{(1)} \circ_2 \mathbf{Z}^{(2)}$ so that each submatrix links observations to the appropriate grouping structure.

Example 6.2. To make this more concrete, consider a simple dataset with three participants, each contributing several observations, and a single predictor x. Suppose that both participant-wise random intercepts and random coefficients are included. In the case of three participants, there are six entries in the random-effects vector to consider,

$$\mathbf{b} = (b_{1,1}, b_{2,1}, b_{1,2}, b_{2,2}, b_{1,3}, b_{2,3})^{\mathsf{T}}, \tag{133}$$

with the first index referring to the effect (coefficient and bias) and the second index corresponding to the participant.

Now consider four observations with the following participant assignments and predictor values:

observation i	participant j	x_i
1	1	x_1
2	1	x_2
3	2	x_3
4	3	x_4

Table 3: Example dataset containing a repeated observation for participant 1.

The corresponding random-effects design matrix $\mathbf{Z}$ is

$$\mathbf{Z} = \begin{pmatrix} x_1 & 1 & 0 & 0 & 0 & 0 \\ x_2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & x_3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_4 & 1 \end{pmatrix}. \tag{134}$$

Multiplying $\mathbf{Z}$ with $\mathbf{b}$ yields

$$\mathbf{Zb} = \begin{pmatrix} b_{1,1}x_1 + b_{2,1} \\ b_{1,1}x_2 + b_{2,1} \\ b_{1,2}x_3 + b_{2,2} \\ b_{1,3}x_4 + b_{2,3} \end{pmatrix}, \tag{135}$$

which corresponds exactly to the participant-specific intercepts and coefficients in the scalar formulation of the model.

6.3.4 Assumptions

As in the case of linear regression, the error terms are assumed to be drawn from a multivariate normal distribution with no covariance (see Section 6.2.1),

$$\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}). \tag{136}$$

The random effects are also assumed to be normally distributed,

$$\mathbf{b} \sim \mathcal{N}(\mathbf{0}, \mathbf{G}), \tag{137}$$

and independent of the residual errors (Gelman and Hill, 2006, Chapter 13).

Here, the matrix $\mathbf{G}$ denotes the covariance matrix of the random effects. For models with both random intercepts and random slopes, it has the form of a block-diagonal matrix, with each block corresponding to a group (e.g. a participant). All groups of a grouping factor share duplicates of the same block. A block captures the variance of the grouping factor random effects in the diagonal and the covariance between them in the off-diagonal (Gelman and Hill, 2006, Chapter 13). The consequence is that, for instance, the random intercepts of a specific grouping level are all sampled from the same distribution.

Example 6.3. For a model with a fixed intercept, one fixed effect and both per-participant random intercepts and random slopes, $\mathbf{G}$ is a block-diagonal matrix where each block corresponds to one of n participants and is a 2×2 matrix capturing the variance of the random intercepts ($\sigma^2_{u_0}$), the variance of the random slopes ($\sigma^2_{u_1}$), and their covariance (σ_{u_0,u_1}):

$$\mathbf{G} = \left.\begin{pmatrix} \mathbf{G}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{G}_1 \end{pmatrix}\right\} n \text{ times}, \tag{138}$$

where $\mathbf{G}_1$ is repeated n times and specified by:

$$\mathbf{G}_1 = \begin{pmatrix} \sigma_1^2 & \sigma_{1,2} \\ \sigma_{1,2} & \sigma_2^2 \end{pmatrix}. \tag{139}$$

Adding another grouping factor would introduce an additional block type $\mathbf{G}_2$ repeated m times where m is the number of groups in that factor.

Applied to reading times, $\mathbf{G}$ captures not only how much intercepts and slopes vary across participants, but also whether participants with higher baseline reading times tend to show stronger or weaker effects of a given predictor like word frequency or word length. Sometimes, it is justified to assume that there are no correlations between random coefficients/intercepts, fixing $\sigma_{1,2} = 0$.

Conditional on the random effects $\mathbf{b}$ and the design matrices, the observations are independent and normally distributed following to the variance of the error terms:

$$\mathbf{y} \mid \mathbf{b}, \mathbf{X}, \mathbf{Z} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b},\ \sigma^2\mathbf{I}). \tag{140}$$

However, since the random effects themselves are random variables, one can also consider the marginal distribution obtained by integrating out the random effects (Verbeke and Molenberghs, 2009, Chapter 3.3).

Corollary 6.1. Given a mixed effects model as defined above, the marginal distribution is given by

$$\mathbf{y} \mid \mathbf{X}, \mathbf{Z} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta},\, \mathbf{Z}\mathbf{G}\mathbf{Z}^\top + \sigma^2\mathbf{I}). \tag{141}$$

Proof. The marginal distribution of $\mathbf{y}$ is obtained by integrating out the random effects. By the law of total expectation,

$$\begin{aligned} \mathbb{E}[\mathbf{y} \mid \mathbf{X}, \mathbf{Z}] &= \mathbb{E}[\mathbb{E}[\mathbf{y} \mid \mathbf{b}, \mathbf{X}, \mathbf{Z}] \mid \mathbf{X}, \mathbf{Z}] \\ &= \mathbb{E}[\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} \mid \mathbf{X}, \mathbf{Z}] \\ &= \mathbf{X}\boldsymbol{\beta}, \end{aligned} \tag{142}$$

since $\mathbb{E}[\mathbf{b}] = \mathbf{0}$.

The covariance matrix follows from the law of total variance:

$$\begin{aligned} \mathrm{Var}[\mathbf{y} \mid \mathbf{X}, \mathbf{Z}] = \mathbb{E}[\mathrm{Var}[\mathbf{y} \mid \mathbf{b}, \mathbf{X}, \mathbf{Z}] \mid \mathbf{X}, \mathbf{Z}] \\ + \mathrm{Var}[\mathbb{E}[\mathbf{y} \mid \mathbf{b}, \mathbf{X}, \mathbf{Z}] \mid \mathbf{X}, \mathbf{Z}]\,. \end{aligned} \tag{143}$$

The first term is

$$\mathbb{E}[\mathrm{Var}[\mathbf{y} \mid \mathbf{b}, \mathbf{X}, \mathbf{Z}] \mid \mathbf{X}, \mathbf{Z}] = \sigma^2\mathbf{I}. \tag{144}$$

For the second term,

$$\begin{aligned} \mathrm{Var}[\mathbb{E}[\mathbf{y} \mid \mathbf{b}, \mathbf{X}, \mathbf{Z}] \mid \mathbf{X}, \mathbf{Z}] &= \mathrm{Var}[\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} \mid \mathbf{X}, \mathbf{Z}] \\ &= \mathrm{Var}[\mathbf{Z}\mathbf{b}] \\ &= \mathbf{Z}\,\mathrm{Var}[\mathbf{b}]\mathbf{Z}^\mathsf{T} \\ &= \mathbf{Z}\mathbf{G}\mathbf{Z}^\mathsf{T}. \end{aligned} \tag{145}$$

Combining both terms yields

$$\mathrm{Var}[\mathbf{y} \mid \mathbf{X}, \mathbf{Z}] = \mathbf{Z}\mathbf{G}\mathbf{Z}^\mathsf{T} + \sigma^2\mathbf{I}. \tag{146}$$

Since $\mathbf{y}$ is an affine transformation of jointly normal random variables, it follows that

$$\mathbf{y} \mid \mathbf{X}, \mathbf{Z} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \mathbf{Z}\mathbf{G}\mathbf{Z}^\mathsf{T} + \sigma^2\mathbf{I}). \tag{147}$$

□

The term $\sigma^2\mathbf{I}$ captures observation-level noise, while $\mathbf{Z}\mathbf{G}\mathbf{Z}^\mathsf{T}$ represents covariance induced by shared random effects. In particular, observations associated with the same grouping factor (e.g. participant or item) are correlated because they depend on the same underlying random variables.

This illustrates that when performing statistical inference, usually the assumptions for linear models (homoscedasticity, normality of residuals, independence of errors) also need to be checked, plus the normality of random effects. Diagnostics and potential remedies are discussed in Section 6.2.1.

6.3.5 Shrinkage of extreme estimates and working with imbalanced data

A defining feature of mixed-effects models – distinct from the fixed-effects-only models discussed in Section 6.2 – is their use of *shrinkage*. It arises as individual group estimates are not based solely on the data from that group but are influenced by the overall distribution of all groups in the population. As a result, extreme or unreliable group-specific estimates are pulled toward the population mean. This phenomenon stabilises estimates, which is particularly useful in cases of imbalanced data, i.e. when some groups have fewer observations than others (Pinheiro and Bates, 2000; Baayen et al., 2008). Furthermore, this improves the model's ability to generalise to new data and increases the reliability of hypothesis tests (Vasishth et al., 2022).

6.3.6 Estimation

Linear mixed-effects models are usually fitted by maximum likelihood or *restricted maximum likelihood estimation* (REML) (Pinheiro and Bates, 2000; Bates et al., 2015).

The ordinary ML criterion is based on the marginal density of the observed response vector. Let $\boldsymbol{\theta}_{\mathbf{G}}$ denote the parameters of the random effects covariance matrix $\mathbf{G}$. Then the marginal likelihood of the observed data is defined by

$$\ell(\,\boldsymbol{\beta}, \boldsymbol{\theta}_{\mathbf{G}}, \sigma^2 \,|\, \mathbf{X}, \mathbf{Z}, \mathbf{y}\,) = P[\boldsymbol{\beta}, \boldsymbol{\theta}_{\mathbf{G}}, \sigma^2](\,\mathbf{y} \,|\, \mathbf{X}, \mathbf{Z}\,), \tag{148}$$

where the density on the right-hand side is obtained by integrating out the random effects:

$$P[\boldsymbol{\beta}, \boldsymbol{\theta}_{\mathbf{G}}, \sigma^2](\,\mathbf{y} \,|\, \mathbf{X}, \mathbf{Z}\,) = \int P[\boldsymbol{\beta}, \sigma^2](\,\mathbf{y} \,|\, \mathbf{X}, \mathbf{Z}, \mathbf{b}\,)\, P[\boldsymbol{\theta}_{\mathbf{G}}](\,\mathbf{b}\,)\, d\,\mathbf{b}. \tag{149}$$

Corollary 6.1 shows that this amounts to the normal distribution $\mathcal{N}(\mathbf{X}\boldsymbol{\beta},\ \mathbf{Z}\mathbf{G}\mathbf{Z}^\top + \sigma^2\mathbf{I})$.

Maximum likelihood estimation selects parameter values that maximise this marginal likelihood. However, it is biased for the estimation of variance components. In contrast to that, restricted maximum likelihood, which takes the likelihood of linear combinations of the response, is less biased for variances and therefore preferred for small sample sizes when the estimation of variances is of interest (Patterson and Thompson, 1971; Harville, 1977). Because the REML likelihood depends on the fixed-effects design matrix, models fitted using REML can only be compared via likelihood ratio tests when they share the same fixed-effects structure (Pinheiro and Bates, 2000, Chapter 1.2). When comparing models with different fixed effects, maximum likelihood estimation must be used instead.

Estimation is performed numerically. The variance-covariance parameters $\boldsymbol{\theta}_{\mathbf{G}}$ are typically optimised using a Cholesky parametrisation to ensure positive semi-definiteness of $\mathbf{G}$, while the fixed effects $\boldsymbol{\beta}$ are obtained conditionally via generalised least squares at each iteration (Pinheiro and Bates, 2000). This process can face several practical challanges which might lead to unstable estimates or convergence failures. In the following, I will discuss these potential pitfalls and how they can be prevented.

Multicollinearity Multicollinearity refers to strong linear dependencies among the predictor variables (Vasishth et al., 2022). This issue is analogous to multicollinearity in ordinary linear regression and concerns near-linear dependencies among the columns of the fixed-effects design

matrix $\mathbf{X}$. Such dependencies make it difficult to attribute variation in the response to individual predictors, which leads to unstable coefficient estimates and inflated standard errors (Vasishth et al., 2022). Multicollinearity can also lead to *convergence errors* when the numerical optimisation fails to locate a stable optimum. In such situations, the optimisation algorithm may terminate prematurely or fail to converge altogether.

Collinearity between two predictors can be stimated using the Pearson correlation coeffcient. To assess multicollinearity, it is advised to compute the *variance inflation factor* (VIF) for each predictor (Vasishth et al., 2022). The VIF quantifies how much the variance of a coefficient is increased due to correlations with other predictors. Values close to 1 indicate little or no collinearity, whereas values above 5 or 10 are commonly treated as indicating potentially problematic collinearity (James et al., 2021).

If severe multicollinearity is detected, several remedies are possible. One may remove or combine correlated predictors or apply dimensionality reduction techniques such as principal component analysis (James et al., 2021).

Singular fits A singular fit occurs when the estimated random-effects covariance matrix is rank-deficient. Equivalently, at least one linear combination of the specified random effects has estimated variance zero (Bates et al., 2018). In simple cases, this may appear as one or more variance components being estimated as approximately zero or as correlations between random effects being estimated at the boundary values (-1 or 1) (Bates et al., 2018). This suggests that the data does not support the full complexity of the specified random-effects structure (Bates et al., 2018). In this sense, some components of the random-effects structure are effectively redundant, and the model is overparameterised.

A common remedy is to simplify the random-effects structure. This may involve removing unsupported random coefficients or reducing the number of correlated random effects (Bates et al., 2015; Matuschek et al., 2017).

6.3.7 Likelihood-ratio tests

In the context of linear mixed-effects models, null hypotheses are formulated in terms of regression coefficients. For a predictor x, the null hypothesis is $H_0 = \{ \boldsymbol{\theta} \in \Theta \mid \beta_x = 0 \}$ which states that x does not contribute to explaining variation in the dependent variable but it does not fix other coefficients specified for the parameter space Θ (Vasishth et al., 2022).

To test H_0 against H_1, one can use a *likelihood-ratio test* (LRT) (Casella and Berger, 2002, Section 8.2). It reduces the hypotheses H_0 and H_1 to a single representative LMEM each, amounting to the best possible selection of parameters. These are called the *null model* and the *alternative model*. One chooses to ignore all other parameter configurations in H_0 since any configuration with worse fit would yield a higher probability of rejecting the null hypothesis. If even this best-fitting null model fits much worse than the alternative, this is evidence against H_0.

Let $\hat{\boldsymbol{\theta}}_0$ and $\hat{\boldsymbol{\theta}}$ denote the maximum likelihood estimates under H_0 and Θ,[30] respectively. The LRT

[30] Usually, the parameter space of the second model includes the elements in H_0 instead of taking H_1, since it is simpler in practice (Wasserman, 2004, Chapter 10.6).

statistic is defined as[31]

$$d_n(\mathbf{X}) \coloneqq -2 \log \frac{P[\hat{\boldsymbol{\theta}}_0](\mathbf{y} \mid \mathbf{X})}{P[\hat{\boldsymbol{\theta}}](\mathbf{y} \mid \mathbf{X})}. \tag{150}$$

for any sample $\mathbf{X}$ and $n \in \mathbb{N}$ with $n = |X|$.

Equivalently, this can be written as the difference in log-likelihoods:

$$d_n(\mathbf{X}) = 2 \left(\log P[\hat{\boldsymbol{\theta}}](\mathbf{y} \mid \mathbf{X}) - \log P[\hat{\boldsymbol{\theta}}_0](\mathbf{y} \mid \mathbf{X}) \right). \tag{151}$$

Assuming that the null hypothesis H_0 is true, both $\hat{\boldsymbol{\theta}}_0$ and $\hat{\boldsymbol{\theta}}$ are estimating the same value. Furthermore, under H_0 and standard regularity conditions, *Wilks' theorem* states that the likelihood-ratio statistic converges in distribution to a chi-square random variable,

$$d_n(\mathbf{X}) \longrightarrow \chi^2_f, \tag{152}$$

as $n \to \infty$ with $f = \dim(\Theta) - \dim(H_0)$ degrees of freedom, i.e. the difference in the number of free parameters (see Lehmann and Romano (2022, Chapter 14.4) for a proof). In the case of testing a single predictor, this reduces to $f = 1$ (Barr et al., 2013).

Thus, given observed data $\mathbf{X}$, the p-value is defined as

$$p = P(\chi^2_f \geq d_n(\mathbf{X})). \tag{153}$$

It is important to note that the χ^2 approximation relies on asymptotic arguments and may be inaccurate in small samples or when testing variance components where parameters lie on the boundary of the parameter space.

The likelihood-ratio test in this form assumes that the compared models are *nested*. That is, the null model must be obtainable from the alternative model by imposing restrictions on its parameter space. Only then do the two models estimate the same point in parameter space if H_0 is true, making Wilk's theorem applicable. Furthermore, in practice, the random-effects structure should be kept constant across the models to ensure that differences in fit can be attributed solely to the predictor of interest (Barr et al., 2013; Bates et al., 2015).

6.3.8 Implementations

In the following, I will briefly introduce two libraries that implement linear mixed-effects models: the standard `lme4` package and the newer `C++`-based `gpboost` library.

6.3.8.1 R-package

Linear mixed-effects models are commonly implemented in the statistical programming language `R` (The R Core Team, 2025). The most commonly used package is `lme4` (Bates et al., 2015), which provides efficient routines for fitting mixed-effects models using maximum likelihood or restricted maximum likelihood.

[31] In the case of linear mixed-effects models, it will be implied that the random effects design matric $\mathbf{Z}$, consisting only of elements from $\mathbf{X}$ is also conditioned on.

The main function for fitting linear mixed-effects models is `lmer`, which takes a model formula and a data frame as input. The formula specifies both the fixed-effects structure and the random-effects structure of the model.

Formula notation In practice, both in documentation and papers, LMEMs are often specified using the formula notation of `lme4`. For example,

```
RT ~ 1 + frequency + length + (1 + frequency + length | subject)
     + (1 | item)
```
(154)

states that the fixed-effects part contains an intercept, frequency and length, that subjects receive a random intercept and random slopes for both predictors, and that items receive a random intercept.

The term `(1 | g)` denotes a random intercept by grouping level `g`. The term `(1 + x | g)` specifies a random intercept together with a random slope for predictor `x` at grouping level `g`. Caution is advised when including only predictor names in the formula, e.g. `(x + y | g)`, since `lme4` implictly includes an intercept. In order to exclude it, it is necessary to write `(0 + x + y | g)` or `(-1 + x + y | g)` (Bates et al., 2015). It is generally advised to denote intercepts explicitly for clarity, especially when reporting model specifications.

If several random effects are placed inside the same parentheses, their covariance is estimated as part of the model. By contrast,

```
(1 | g) + (0 + x | g) + (0 + y | g)
```
(155)

specifies the same random intercept and slopes but constrains their covariance to zero by placing them in separate terms. An equivalent and more compact notation is provided by the double pipe operator `||`. In this case

```
(1 + x + y || g)
```
(156)

is equivalent to Equation 155.

6.3.8.2 gpboost

An alternative implementation of mixed-effects models is provided by the `gpboost` framework (Sigrist, 2022; Sigrist, 2026), which combines tree-based boosting methods with Gaussian processes and mixed-effects models. For linear mixed-effects models, the tree-boosting component is inactive and both `lme4` and `gpboost` fit the same model using a marginal likelihood formulation. The main differences lie in the implementation and optimisation procedures. `lme4` uses the derivative-free `BOBYQA` optimiser (Powell, 2009) as its standard, whereas `gpboost` employs gradient-based optimisation (Sigrist, 2026).

The optimisation process is significantly faster with `gpboost`. Benchmarks demonstrate up to 100x speedup over `lme4` across sample sizes (Sigrist, 2021). This makes the tool especially valuable for model estimation on large amounts of data. However, `gpboost` does not support correlations between random effects of the same grouping level as opposed to `lme4` (Sigrist, 2026). Secondly, the package requires the explicit construction of the design matrices $\mathbf{X}$ and $\mathbf{Z}$ as opposed to the convenient syntax of `lme4`. Lastly, it has fewer utilities for statistical inference available.

6.4 Psycholinguistic application

Having introduced linear mixed-effects models and explained how to use them for statistical inference in the preceding section, there are still a few specificities missing that pertain to their usage in the study of repeated psycholinguistic measurements, specifically in human language processing. These will be discussed in the following.

6.4.1 Crossed random effects in psycholinguistics

Psycholinguistic data typically involves at least two grouping factors: participants and items such as regions of interest (Baayen et al., 2008). Participants differ in baseline reading speed and possibly also in the magnitude of predictor effects. Items differ in their intrinsic difficulty and (possibly other factors not accounted for). Since participants and items are usually not nested inside one another, the resulting mixed-effects structure is described as *crossed*, in contrast to *nested* designs (Baayen et al., 2008).

A typical model for word-level reading times therefore includes random effects for both participants and items. This structure directly addresses the problem identified by Clark (1973): inference should generalise beyond the specific sample of participants and beyond the specific sample of linguistic materials.

6.4.2 Choosing the random-effects structure

A central practical question concerns the complexity of the random-effects structure to choose for modelling. Barr et al. (2013) argue that, for confirmatory hypothesis testing, one should fit the maximal random-effects structure justified by the experimental design, because underspecified random-effects structures may inflate Type-I error as the ignore the requirement of conditional independence of the observations (see Section 6.2.1). They particularly observed researchers use models that only included random intercepts and no random coefficients, producing overly optimistic statistical tests.

As Barr et al. (2013) explain, variance that is unaccounted for can influence statistics in two ways: (i) unaccounted-for variation that is orthogonal to the fixed predictor to test – such as random intercepts – reduces the test's power; (ii) ignored variation confounded with the potential effect such as a random coefficient might result in deviations in fixed effect estimates which can increase Type I error, leading to the detection of effects where there are none.

Barr et al. (2013) note that failure to converge in fitting a LMEM using maximum likelihood estimation occurs more often for more complex effects structures. To counteract this, simplification of the random effects structure is warranted. However, instead of removing random slopes altogether, increasing the risk of detecting spurious effects, the researcher is advised to employ a cautious simplification strategy and should first consider removing the correlations between random effects from the model specification (Barr et al., 2013).

Barr et al. (2013)'s suggestions received wide adoption after their publication, as five years later Bates et al. (2018) address the issue of an increasing number of researches reporting results from maximal, overspecified LMEMs that did not converge. Bates et al. (2018) take the stance that non-convergence is not a sign of poor optimisation algorithms but of trying to fit a model that is too

complex for the data at hand. Additionally, rather than chasing for random coefficients with near-zero variance that usually have practically no consequence for tests of fixed effects to minimise Type I error, they argue for manual adjustment of the significance criterion c instead.

Matuschek et al. (2017) shows that while maximal models effectively keep down Type I error, they can decrease power if overparameterised for the data. They argue that one can maintain low Type I error while achieving higher power by using model selection criteria to find mixed effect structures actually supported by the dataset in a *forward-driven data-driven model selection* (compared to the *design-driven approach* of Barr et al. (2013)). Their rationale is that the estimation of mixed effect correlations is not expected to capture any variance if the random coefficient itself is not contributing significantly to model fit in the first place. In practice, their approach involves starting from a basic model, iteratively conducting likelihood-ratio tests for new random effect componence to add and including them if the desired significance threshold is met, contrasting the position of Barr et al. (2013) who had argued that superfluous random effects are not of concern for inference and disregarded data-driven approaches on the grounds of the risk of falsely excluding random effects.

A principled model simplification strategy similar to a *backward-driven* approach is outlined by Bates et al. (2018). The procedure starts from a maximal random-effects structure and iteratively reduces model complexity based on the empirical support provided by the data:

A first step consists in assessing the effective dimensionality of the random-effects structure, for instance via a principal component analysis of the estimated covariance matrices. This allows one to determine how many variance components are supported by the data and to identify redundant parameters.

If the maximal model exhibits convergence problems or yields a singular fit, a substantial simplification can be performed by removing correlation parameters, for example by fitting a zero-correlation model (Bates et al., 2018). Subsequently, coeffecients can be eliminated in a stepwise fashion, following the hierarchical principle that higher-order terms (such as interactions between predictors) should be removed first.

At each stage, the reduced model is compared to the previous model using likelihood-ratio tests or information criteria in order to assess whether the components to remove are actually unsupported by the data. Bates et al. (2018). Random effects that do not contribute significantly to model fit can be excluded, yielding a more frugal specification. Finally, once a reduced model with supported variance components has been identified, Bates et al. (2018) propose reintroducing correlation parameters for these components if supported.

6.4.3 Spillover

As discussed in Section 6.4.2, processing slowdown is often delayed in reading time measurements, occurring not (only) at the word that it is associated with but at later regions (Smith and Levy, 2013; Xu et al., 2023). Therefore, when estimating the effect of a word-based predictor on reading times, it is advisable to include delayed versions of it – both to account for confounders and to identify potential lagged effects. This adds to the complication of mixed-effect structure selection discussed in Section 6.4.2 as lagged versions of predictors may in principle also vary across subjects or items, suggesting the inclusion of corresponding random slopes where supported by the data.

A version of predictor `x` lagged by k steps is commonly written as `x.k` when specifying LMEMs in formula notation. Including it in the regression model has the effect of associating the value of `x` at region i to the region at position $i + k$. See for instance Equation 157 in which the reading time of each word is predicted using its frequency and the frequency of the preceding word.

As a convention, I will write that a model has a *spillover window of* k if there is a predictor x for which lagged versions `x.i` are included for all $i \in [k]$.

```
RT ~ 1 + frequency + frequency.1
     + (1 + frequency + frequency.1 | subject)
```
(157)

A natural modelling question concerns the appropriate to consider. A maximal approach would include delayed versions of all preceding items into the linear-mixed effects model. However, this would severely shrink the effective sample size to a minimum: given a window of k, the first k observations in a sentence (or more broadly, a context) lack preceding items that could provide them with spilled-over values and therefore need to be omitted in the analysis.

The standard procedure for dealing with spillover is therefore data-driven (cf. Xu et al., 2023): Starting with $k = 0$, one iteratively tests whether adding predictors lagged by $k + 1$ positions improves model fit. Specifically, for each $k \geq 0$, a model with spillover $k + 1$ is compared to a model with spillover k using a likelihood-ratio test. If the additional lagged predictors significantly improve model fit, the procedure continues with $k + 1$; otherwise, the model with spillover k is selected. This is repeated until some pre-defined and justifiable maximal spillover window is reached or until increasing the window further does not significantly improve model fit.

Following the methodology outlined in Section 6.4.2, the lagged versions of the predictors are also included as random effects of both null and alternative models. However, in practice, the maximal random-effects structure may not be identifiable. In such cases, model simplification strategies as discussed in Section 6.4.2 need to be applied, for example by removing correlations or higher-order random slopes. If convergence issues persist, it may be necessary to restrict the spillover analysis to a subset of theoretically motivated predictors. Such simplifications should be guided by theoretical considerations and validated by model diagnostics.

6.4.4 Multiple testing

In many empirical settings, several statistical tests are performed on the same dataset. For example, multiple predictors may be evaluated via separate model comparisons against a common baseline. This practice is referred to as *multiple testing* or *multiple comparisons* (Wasserman, 2004, Chapter 10.7).

A central issue arising from multiple testing is the inflation of the Type I error rate. If multiple independent tests are conducted, the probability of making at least one Type I error increases with the number of tests (Wasserman, 2004, Chapter 10.7). To illustrate this, suppose that m independent hypotheses are tested, each at significance level c. The probability of not making a Type I error in any of the tests is $(1 - c)^m$. Consequently, the probability of making at least one Type I error is given by $1 - (1 - c)^m$, which exceeds c for $m > 1$ under independence.

In the general case, this quantity is known as the *family-wise error rate* (FWER), defined as the probability of making at least one Type I error among a family of hypotheses (Hochberg and

Tamhane, 1987, Appendix 1).

Definition 6.5 (Family-wise error rate)**.** Let p_i denote a random variable for the p-value of test $i \in [m]$ for a random dataset. Let $I_0 \subseteq [m]$ be the index set of true null hypotheses. Then the family-wise error rate is defined as

$$\text{FWER} \coloneqq P\left(\bigcup_{i \in I_0} (p_i \leq c)\right).$$

It is therefore important to interpret p-values with caution when multiple tests are conducted. Several methods are available for adjusting statistical results. In the following, two common methods will be introduced.

6.4.4.1 Bonferroni correction

A simple and widely used method is the *Bonferroni correction*, which replaces the significance level c by c/m for each individual test (Wasserman, 2004, Chapter 10.7). Alternatively, the p-value can be replaced with p/m. This ensures that the overall probability of making at least one Type I error does not exceed c.

The proof can be attained using Boole's inequality, assuming $P(p_i \leq c) \leq c$:

$$\text{FWER}_{\text{Bonf}} = P\left(\bigcup_{i \in I_0} \left(p_i \leq \frac{c}{m}\right)\right) \leq \sum_{i \in I_0} P\left(p_i \leq \frac{c}{m}\right) \leq |I_0| \frac{c}{m} \leq m \frac{c}{m} = c. \tag{158}$$

While Bonferroni correction does not come with any additional assumptions, it can be overly conservative when the tests are positively correlated, leading to smaller FWER due to more overlap in the false rejection events. Thus, the bound in Equation 158 may be loose, possibly reducing statistical power (Wasserman, 2004, Chapter 10.7).

6.4.4.2 Benjamini-Hochberg procedure

An alternative approach is to control the *false discovery rate* (FDR), a generalisation of the p-value to multiple tests, defined as the expected proportion of false rejections among all rejected hypotheses (Wasserman, 2004, Chapter 10.7).

Definition 6.6 (False discovery rate)**.** Let $\delta_i \in \{0, 1\}$ indicate whether hypothesis H_i is rejected, and let $I_0 \subseteq [m]$ denote the index set of true null hypotheses. Then the false discovery rate is defined as

$$\text{FDR} \coloneqq \mathbb{E}\left[\frac{\sum_{i \in I_0} \delta_i}{\sum_{i=1}^{m} \delta_i}\right].$$

A widely used method of this type is the *Benjamini-Hochberg procedure* (Benjamini and Hochberg, 1995). Suppose that m hypotheses are tested, yielding p-values $p_1, \ldots, p_m$. Let f assign each index $i \in [m]$ a unique rank in $[m]$, so that $p_{f^{-1}(1)} \leq \cdots \leq p_{f^{-1}(m)}$.

Definition 6.7 (Benjamini-Hochberg procedure)**.** Given a target level $c \in (0, 1)$, define

$$k = \max\left(\left\{ i \in [m] \;\middle|\; p_{f^{-1}(i)} \leq \frac{i}{m} c \right\}\right).$$

All hypotheses corresponding to $p_{f^{-1}(1)}, \ldots, p_{f^{-1}(k)}$ are rejected.

In order to express the rejection criterion relative to the original threshold c, one can adjust the p-value with rank $f^{-1}(i)$ using $\min\left(\{\, p_{f^{-1}(j)} m/j \mid j \geq i \,\}\right)$ (Yekutieli and Benjamini, 1999). It can be seen that this correction yields smaller p-values than Bonferroni correction ($p_{f^{-1}(i)}/m$), thus retaining more power.

The rationale behind the procedure can be understood as follows. Under a true null hypothesis, the corresponding p-value is uniformly distributed on $[0, 1]$. Therefore, for a threshold $c \in [0, 1]$, the expected number of false rejections is approximately $|I_0|c$ (under independence). If the threshold is chosen as $t = \frac{i}{m} c$ and i hypotheses are rejected, the expected proportion of false rejections among the rejected hypotheses is approximately

$$\frac{|I_0| t}{i} = \frac{|I_0|}{m} c. \tag{159}$$

Since $|I_0| \leq m$, this implies

$$\frac{|I_0|}{m} c \leq c. \tag{160}$$

Thus, the procedure adaptively selects the largest rejection set for which the expected proportion of false discoveries remains controlled. BH works because it scales the rejection threshold with the number of rejections, ensuring that the expected number of false positives grows at most proportionally to the number of discoveries.

Under suitable assumptions, in particular independence or certain forms of positive dependence among the tests, the Benjamini–Hochberg procedure controls the false discovery rate at level c (Benjamini and Hochberg, 1995). Compared to family-wise error rate control, FDR-based procedures are typically less conservative and therefore more powerful, especially in settings with many simultaneous tests (Benjamini and Hochberg, 1995).

7 Experiment 1

This section addresses the first research question:

Q1 How well do structural costs extracted from transformer models with a pure language modelling objective, in particular attention entropy and weighted attention distance, predict human reading times?

The aim of the experiment is to assess whether quantities derived from the internal computations of a neural language model account for variation in behavioural measures of human reading. The focus lies on structural cost measures computed from transformer attention distributions. These measures are interpreted as possible proxies for aspects of memory retrieval and information integration during incremental sentence processing.

The section is organised as follows. First, the methodological framework is introduced, including the training of the language model, the extraction of model-derived predictors and the statistical evaluation procedure. References to theoretical concepts and methods introduced in earlier sections are provided where necessary. The second part presents the experimental results. Finally, the findings are interpreted with respect to the research question and discussed in relation to related findings in the field.

7.1 Methodology

The evaluation proceeds in three main steps:

1. A transformer language model is trained on a large corpus using a pure language modelling objective.
2. For each word in a reading-time corpus, four model-derived predictors are computed (Section 7.1.2), including surprisal and structural cost measures based on transformer attention scores.
3. A likelihood-ratio test with respect to an effect of the cost-measures on reading times is conducted by fitting linear mixed-effects models, allowing for random effects on grouping levels such as participants and items (Section 6).

The predictive contribution of each structural cost measure beyond surprisal is assessed by comparing nested models with and without the corresponding predictor using likelihood-ratio tests. Rejection of the null hypothesis is interpreted as evidence that the respective structural cost accounts for reading-time variation not already captured by surprisal and the baseline predictors. Depending on the cost under consideration, such a result provides support for [H1a] or [H1b].

The analysis combines confirmatory and exploratory components. The two main predictors, attention entropy and weighted attention distance, are specified in advance on the basis of prior work and the theoretical motivation developed in the preceding sections. The corresponding tests are therefore treated as confirmatory hypothesis tests (Barr et al., 2013; Tukey, 1980). In addition, two structurally related attention-based predictors are evaluated exploratorily. These analyses serve to

assess whether alternative operationalisations of attention structure show comparable relationships with reading times, but they are interpreted as hypothesis-generating.

The operationalisation of an attention-based structural cost involves several design choices, including the choice of language model, training regime, tokenisation, attention aggregation method, and reading-time measure. Consequently, a non-significant result should not be interpreted as decisive evidence against the underlying psycholinguistic theory. It may instead indicate that the particular linking hypothesis is inappropriate, that the chosen transformer parameterisation is inadequate, or that the available data provide insufficient statistical power.

For this reason, the analysis is restricted to relatively simple and interpretable design choices. This makes the tested linking hypotheses transparent and reduces the risk that positive results arise only from highly specific modelling decisions. More complex variants can then be motivated by the limitations of these simpler formulations in future work. Possible confounders and limitations will be discussed throughout the analysis.

It is important to emphasise that the analysis is correlational in nature (see Section 3.1). The predictors are derived post hoc from the stimulus material and are not experimentally manipulated. Consequently, observed relationships between model-derived quantities and reading times do not by themselves establish causal mechanisms. Instead, the results are interpreted as evidence regarding the extent to which the internal representations of language models capture statistical regularities that align with human processing difficulty.

The following sections describe the three components of this experiment in detail, beginning with the language model training procedure.

7.1.1 Language model

All model-derived predictors are obtained from a transformer-based language model trained with a standard next-token prediction objective (see Section 5). Importantly, the training objective does not explicitly encode any notion of syntactic structure or processing cost. Instead, structural properties emerge implicitly from the optimisation of predictive performance. This experiment investigates whether such emergent representations are nevertheless informative with respect to human reading behaviour.

7.1.1.1 Architecture

A transformer model comparable in architecture to `GPT-2 medium` (see Section 5.3, 16 attention heads and 24 transformer layers) was chosen and re-implemented by extending the code-base of Mielczarek et al. (2025).

The choice of language model family is predetermined by the research question. Focusing on the `GPT-2 medium` model variant is motivated by the fact that prior research on attention patterns for RT prediction has predominantly focused on the `GPT-2 small` transformer (Ryu and Lewis, 2022; Oh and Schuler, 2022; Ryu and Lewis, 2025). It is unknown whether the effects found by researchers generalise to transformers with more attention heads and this study might contribute to answering this question. Furthermore, [Q1] is not formulated with respect to a pre-trained transformer model of a specific size – the aim is to show a generalisable result.

The latter also motivates training a new model instead of testing existing transformer parameter

Hyperparameter	Value
Architecture	transformer
Number layers	24
Number of attention heads per layer	16
Dimensionality	768

Table 4: Fixed architectural hyperparameters for language model training in experiment 1.

estimates. As neural network training is a stochastic process, evidence based on parameters gained from additional training trials contributes toward cumulative evidence.

The second rationale for training a new model is to take into account only intra-sentential contexts, as the language processing theories underlying the structural predictors to test in this work limit themselves to predictions about the nature of sentence processing. Thus, the methodology proposed here generates predictors based on the internals of transformer models applied to single sentences to prevent extra-sentential confounders. However, `GPT-2` was originally trained using a wide context window that crosses sentence boundaries (Section 5.3.3). Previous research indicates that `GPT-2` yields noticeably worse perplexity when revaluated on sentential contexts than models specifically trained on data with this structure (Mielczarek et al., 2025).[32] The training of a new model is a measure to rule out poor adaptation to short contexts as an inhibiting factor.

The same motivation underlies the use of word-level tokenisation instead of Byte Pair Encoding (Section 5.3.2.2) since psycholinguistic theory often employs the simplifying assumption that whole words constitute cognitive units (e.g. cue-based retrieval, Section 4.3.2). Furthermore, this reduces complications regarding the aggregation of attention-based measures discussed in Section 5.4.2.5.

The use of whitespace-based tokenisation generally has a side effects: a large vocabulary size and thus a significant increase in parameters due to the embedding matrix. In order to lower the number of token instances, punctuation is removed in preprocessing and all words are uncapitalised. These two simplifications are often applied in word-based implementations (Jurafsky and Martin, 2026, Chapter 2). I assume that they will not inhibit the model's performance since both punctuation and capital letters most often (but not exclusively) serve to signify the start and the end of a sentence which is not necessary in a sentence-only model.

Table 4 contains key architectural properties of the language model optimised for this experiment. Due to restricted availablility of computational resources, the embedding dimensionality had to be reduced from 1024 as in `GPT-2 medium` to 768 as in `GPT-2 small`. Despite this, due to the large word embedding matrix, the overall number of tunable parameters is still larger than that of `GPT-2 medium` (483 million vs 345 million, see Section 7.2.1).

7.1.1.2 Data

All training is conducted on the pre-processed WikiText 103 dataset (Merity et al., 2016) consisting of over 103 million tokens from Wikipedia grouped into paragraphs. Each paragraph can

[32] It has to be noted that the evaluation of `GPT-2` was performed by Mielczarek et al. (2025) out-of domain while the custom-trained models were evaluated on their training domain. However, as `GPT-2` was optimised using more data and the custom models are by a magnitude smaller, this result still suggests to be wary of assuming that `GPT-2` is suitable for the task at hand.

	Value
Dataset	wikitext-103-v1
Minimum sentence length	3 tokens
Maximum sentence length	60 tokens
Training sentences	3 574 204
Validation sentences	15 312
Vocabulary size	222 319 toke types

Table 5: Overview over the dataset used for LM training and validation, after pre-processing.

contain several sentences. Following the motivation outlined in the preceding section, I decompose the paragraphs into individual sentences and prepend a special beginning-of-sequence token to each sentence to allow computing from it the probility for the first word of a sequence.

Since transformer time complexity scales with n^2 (Vaswani et al., 2017), where n is the length of the input sequence,[33] a maximum sentence length of 60 is enforced for practical reasons by discarding all sentences that have more than 60 tokens. I do not believe this limit to be overly restrictive since (Sichel, 1974) report average sentence lengths of 24–26 words for English literary works, with a standard deviation of 11–14.

The standard procedure of differentiating between disjoint training, development and testing splits of the data is adopted. The partition is provided by the WikiText 103 dataset itself. Overfitting motivates the use of separate splits because a model's performance on the data it was trained on is not a reliable indicator of how well it generalises (Jurafsky and Martin, 2026, Chapter 4). When a model is fit on a training set, it may capture not only the underlying regularities of the data-generating distribution but also noise and unique patterns specific to that sample, leading to overly optimistic estimates of performance. To obtain an unbiased assessment, the data is therefore partitioned into disjoint subsets. The training set is used to estimate model parameters via gradient descent and the validation set is used for model selection (here: choosing hyperparameters and training checkpoints) as it gives an indication as to the model's prediction capabilities on data not seen during training. Lastly, the test set is usually reserved for a final evaluation, the results of which are reported (Jurafsky and Martin, 2026, Chapter 4). This is necessary since model selection processes based on validation performance can in turn overfit the validation set. However, the test split is not used here since language modelling is not in itself a goal of the study.

Preprocessing The Wikitext corpus used in this experiment was made available by Merity et al. (2016) in a slightly preprocessed version for word-level work.[34] The data curators separated all punctuation and words, as well as hyphens in hyphenated words and clitics such as the possessive marker *'s* and constructed a vocabulary of the 267 735 most-common resulting tokens (Merity et al., 2018). All other string were replaced with `<unk>`. Besides limiting the vocabulary size, this allows a language model to learn a generic representation for rare words not in the training split to be able to deal with any new data (Setction 5.3.2.2).

Additional preprocessing is performed in this work to reduce the vocabulary size further by turning all characters to lowercase and removing punctuation and by removing titles, which are marked

[33] This is rooted in the fact that an attention head computes scores for all pairs of items in the input sequence.

[34] The non-preprocessed version is available under the name wikitext-103-raw-v1.

using in = symbols in the beginning and end of a paragraph in WikiText. Additionally, short pargraphs with less than five words are cut from the data in order to exclude artifacts such as formatting errors. All paragraph are split into individual sentences using the `spacy` English transformer pipeline (`en_core_web_trf`, Honnibal et al. (2020)). Aside from removing sentences longer than 60 tokens, sentences shorter than 3 tokens are also removed to exclude ill-formed artifacts that might have resulted from inaccuracies in sentence splitting or dataset composition. Overall, this results in 3 574 204 training sentences. Further results of pre-processing are summarised in Table 5.

Lastly, a special *beginning of sentence token* is prepended to each sentence to enable the model to learn a distribution for the first word in a sentence. This is necessary since the transformer model can only condition on non-empty contexts (Section 5.3).

Data loading The data is made available to the training procedure in batches of 1920 sentences in random order. All items of a batch need to have the same sequence length to process their embeddings with tensor operations. For `GPT` models, this is usually not an issue since they sample a fixed context window from a running text. However, here the sampled items do have different lengths.

A common strategy is to append *padding tokens* to sentences to increase their length to that of the longest item in their batch (Zeng et al., 2022). These padding tokens are then masked in attention heads and ignored when computing the language modelling loss. However, this has the consequence that part of the computation performed by the model does not contribute towards the training goal. In other words: The ratio between items in the loss and computational steps is smaller than in the case of models with a fixed sampled context window, wasting computation. The larger the batch size, the longer the expected maximum length element in a batch is, thus exacerbating the problem.

Efficiency is improved in this work by performing so called *bucketing* (Sutskever et al., 2014; Kocmi and Bojar, 2017). This involves prior grouping of sentences into batches of uniform length, making padding obsolete. It does not change the overall sampling distribution, however, batches with uniform length might change optimisation dynamics due to their gradient being less representative of the full dataset. In practice, Kocmi and Bojar (2017) report no substantial performance difference between machine translation models trained in bucketed and padded fashion. To reduce possible ordering effects, I randomise both the order of buckets and the order of sentences within buckets before each epoch.

7.1.1.3 Hyperparameter search

While the number of transformer layers and attention heads is fixed to the architecture of `GPT-2 medium`, I refrain from simply adopting hyperparamters such as learning rate and regularisation settings from earlier transformer implementations since the differing training regime (choice of dataset, single sentences instead of multi-sentence contexts, ...) likely induce a different set of optimal hyperparamters. For this reason, a hyperparameter search is performed before conducting extended model training.

Let Λ be the space of hyperparameters that affect the architecture of the language model (e.g. dimensionality), H the space of hyperparamters that affect the optimisation process (e.g. learning

rate) and Θ_λ the space of tunable neural network parameters (weight matrices, bias vectors and layer normalisation parameters) given an architectural configuration $\lambda \in \Lambda$. Let w_{train} be the training data, w_{val} the validation data, $\mathcal{L}_\lambda[\theta](\, w_{\text{train}}\,)$ the training loss for a model with an architecture set by $\lambda \in \Lambda$ and parameterisation $\theta \in \Theta_\lambda$ and $\mathcal{L}_\lambda[\theta](\, w_{\text{val}}\,)$ the corresponding validation loss (more generally called the *objective value* for the hyperparameter optimisation). Hyperparameter optimisation is the problem of finding

$$\begin{aligned} \lambda^*, \eta^* &= \arg\min_{\lambda\in\Lambda,\eta\in\mathrm{H}} \mathcal{L}_\lambda[\theta^*(\lambda,\eta)](\, w_{\text{val}}\,), \\ \text{with} \qquad \theta^*(\lambda,\eta) &= \arg\min^{\eta}_{\theta\in\Theta_\lambda} \mathcal{L}_\lambda[\theta](\, w_{\text{train}}\,), \end{aligned} \tag{161}$$

where $\arg\min^\eta$ describes an optimisation procedure parameterised by η.

Here, the search will estimate suitable levels of dropout, feed-forward hidden dimensionality factor (f_{ff}), learning rate, whether to use sinusoidal or learned positional encodings and whether to include bias terms in the affine transformations of the transformer. The initial search space of the parameters is given in Table 6. The learning rate is sampled from a log-uniform distribution to sample values from all orders of magnitude, while the other parameters are sampled from uniform distributions.

Parameter	Range	Distribution
f_{FF}	$\{\,4,\ldots,8\,\}$	discrete uniform
Bias term	$\{\,\text{yes},\text{no}\,\}$	discrete uniform
Dropout		
– Embedding	[0,0.5]	uniform
– Feed-forward	[0,0.5]	uniform
– Residual	[0,0.5]	uniform
Positional encoding	$\{\,\text{sinusoidal},\text{embedding}\,\}$	discrete uniform
Learning rate	$[1\times10^{-4}, 1\times10^{-3}]$	log-uniform

Table 6: Hyperparameters to identify for the transformer language model in experiment 1, corresponding search spaces and the initial distribution from which they are sampled.

The search interval [0,0.5] includes the no-dropout setting, the common transformer value 0.1 (Vaswani et al., 2017), and stronger regularisation values approaching the classical dropout reference point of 0.5 (Srivastava et al., 2014). The feed-forward factor is included in the hyperparameter search to potentially allow the model to acquire increased expressive power without enlarging the number of attention heads.

If for some numerical parameter the best result of the search process originates from a value at the border of its search space, the window is modified and the search repeated. For instance, it is possible (but probably unlikely) that ideal feed-forward dropout lies at 0.9. The initial search space is limited to allow for a denser set of results in the region assumed to include the ideal value according to prior assumptions. But if hyperparamter search identified the best value at $\text{drop}_{\text{ff}} \approx 0.5$, the interval would subsequently be moved to the right, e.g. to [0.4,1.0], and the search repeated.

Search process While the inner training objective $\arg\min^\eta$ in Equation 161 can be optimised via gradient descent with the parameters being learnable model weights, this is not the case for the

hyperparameters in the outer objective since they determine factors such as model architecture for which we cannot compute gradients with respect to the training loss.

Therefore, the search is implemented by repeatedly sampling hyperparamters λ, η as per the distributions noted in Table 6 and then estimating $\theta^*(\lambda, \eta)$ via neural network training. Every such training run is called a *trial* (Franceschi et al., 2025) with the number of trials set to $n_{\text{trial}} = 100$ in this work. In each trial, a neural network is trained for a maximum of $n_{\text{epochs}} = 10$ epochs to keep the hyperparameter search computationally feasible. Since this means that the validation loss $\mathcal{L}_\lambda[\theta^*(\lambda, \eta)](\, w_{\text{val}}\,)$ is not guaranteed to have reached its theoretical minimum, extended training will be performed after finding the best set of hyperparamters. However, note that due to different training dynamics, it is certainly possible that other configurations that were discarded due to reporting a higher best loss within the available epochs $n_{\mathbf{epochs}}$ would yield even better scores in extended training.

Training is performed following Section 5.3.3 with a batched loss iterating through the dataset. After processing a batch, gradient descent is performed. Every $b_{\text{interval}} = 500$ batches (i.e. every 960 000 sentences), the model is evaluated on the validation split of the WikiText corpus, reporting the loss to the hyperparameter optimiser. b_{interval} will be called the *validation interval*. An early stopping criterion is used to prevent wasting resources on models that have begun to overfit: if validation loss does not improve over the lowest validation loss found so far for $b_{\text{early}} = 3$ consecutive validation intervals, the trial is terminated. In each trial, a different *random seed* is used, to ensure different sampling behaviour from the dataset.

The hyperparameter search procedure can be implemented via naive sampling and by running all trials for the set number of maximal epochs n_{epochs} (Franceschi et al., 2025). Here, a more efficient implementation for the outer $\arg\min$ optimisation in Equation 161 is chosen by using both a *sampler* and a *pruner*.

Sampler The sampler determines how new hyperparameter configurations are proposed on the basis of the trials completed so far. In this work, the *Tree-structured Parzen Estimator* (TPE) sampler (Bergstra et al., 2011; Bergstra et al., 2013; Watanabe, 2023; Yamazaki, 2024), implemented in the `optuna` hyperoptimisation package (Akiba et al., 2019), is chosen.

TPE is a form of Bayesian optimisation. It introduces two surrogate models to estimate densities over configurations that lead to good/bad results, conditional on observed objective values (Bergstra et al., 2011). More precisely, let $\lambda_i \in \Lambda$ and $\eta_i \in \mathrm{H}$ denote the hyperparameter configuration for trial i and let $y(\lambda_i, \eta_i) = \mathcal{L}_{\lambda_i}[\theta^*(\lambda_i, \eta_i)](\, w_{\text{val}}\,)$ be its observed objective value. Let n denote the number of completed trials and let $y(\lambda_{k_1}, \lambda_{k_1}), \ldots, y(\lambda_{k_n}, \lambda_{k_n})$ be an ascending ordering of the statistics. Furthermore, let n_i^{good} denote the number of trials that are to be considered promising and define the threshold

$$y^* \coloneqq y(\lambda_{k_{n_i^{\text{good}}}}, \eta_{k_{n_{\text{good}}}}). \tag{162}$$

n_i^{good} is determined by a function of the number of completed trials, which by default yields 10% of trials in early stages, while capping the number of promising configurations at 25 (Akiba et al., 2019).

TPE then defines two Gaussian Mixture Models,

$$l(\lambda, \eta) \coloneqq P(\, \lambda, \eta \,|\, y(\lambda, \eta) \leq y^*\,), \qquad g(\lambda, \eta) \coloneqq P(\, \lambda, \eta \,|\, y(\lambda, \eta) > y^*\,), \tag{163}$$

which are updated after completing a trial.

Before each trial, a predefined number of candidate configurations is proposed from the density associated with promising regions l, for each candidate the ratio

$$\frac{l(\lambda, \eta)}{g(\lambda, \eta)}. \tag{164}$$

is computed and then the configuration with the highest such ratio is chosen. Under the assumptions of Bergstra et al. (2011), maximising the ratio is equivalent up to proportionality to maximising expected improvement over the best observations so far.

Hence, the sampler starts with a number of startup trials where the configuration is sampled according to the pre-specified distribution, necessary to acquire a sufficiently large set of observations for stable estimate (Bergstra et al., 2011; Akiba et al., 2019). After that, it gradually reshapes the sample distribution to prefer configurations that are likely under the density of good trials and unlikely under the density of bad trials, as it learns how model performance is influenced by different selections of hyperparameters. By doing that, TPE improves over random search in reaching empirical convergence to low objective values faster (Bergstra et al., 2011).

Pruner A pruner is used in hyperparameter optimisation to distribute computational resources across trials in an efficient manner to speed-up training. I will use the *hyperband* pruner (Li et al., 2018) available in `optuna` since empirical results suggest that combining Hyperband with TPE sampling yields strong performance in practice (Ohta, 2020).

Hyperband is a resource allocation strategy based on *successive halving* (Jamieson and Talwalkar, 2015) which evaluates a set of configurations with a small initial resource budget and eliminates poorly performing configurations in multiple rounds, while allocating increasing resources to the remaining ones.

Let n denote the initial number of configurations, r the initial resource allocated to each configuration, and $\eta > 1$ a reduction factor. In each round i, all remaining configurations are evaluated with resource r_i, and only the best fraction $1/\eta$ is retained. The number of configurations and the resource allocation evolve as

$$n_i = \left\lfloor \frac{n}{\eta^i} \right\rfloor, \qquad r_i = r \cdot \eta^i. \tag{165}$$

Thus, successive halving progressively reduces the number of configurations while increasing the computational effort available to each per configuration (Jamieson and Talwalkar, 2015).

Hyperband extends this idea by considering multiple choices of the initial number of configurations n and the initial resource r under a fixed maximum resource budget (Li et al., 2018). This avoids having to choose a single trade-off between exploration and exploitation in advance which might be suboptimal.

Let $r_{\max}$ denote the maximum resource that can be allocated to a single configuration, let $\eta > 1$ be the reduction factor and define

$$s_{\max} = \lfloor \log_\eta r_{\max} \rfloor. \tag{166}$$

For each $s \in \{0, \dots, s_{\max}\}$, Hyperband defines a bracket, corresponding to one run of successive halving with

$$n(s) = \left\lceil \frac{s_{\max} + 1}{s + 1} \eta^s \right\rceil, \qquad r(s) = r_{\max}\, \eta^{-s} \tag{167}$$

being the initial number of configurations and the initial resource allocation for bracket s.

Each bracket therefore represents a different trade-off between the number of configurations and the amount of resources allocated to each configuration (Li et al., 2018). Large values of s correspond to many configurations evaluated with very small initial budgets (*exploration*), whereas small values of s correspond to fewer configurations evaluated with larger budgets (*exploitation*) (Li et al., 2018). In the extreme case $s = 0$, all configurations are evaluated with the full resource $r_{\max}$, which reduces to standard random search. The procedure starts with successive halving for the maximal number of configurations $s = s_{\max}$ and decrements s successively.

An overview over the hyperparameter optimisation, sampler and pruner settings is given in Table 7. All other settings are set to the default values defined in *Pytorch 2.6.0* and *optuna 4.6.0*.

Setting	Value
Trials	100
Maximum epochs per trial	10
Batch size	1920 sentences
Validation interval	500 batches
Early stopping	>3 validations without improvement
Sampler startup trials	10
Sampler candidates	24
Pruner maximal resource	"auto"
Pruner reduction factor	3

Table 7: Settings for hyperparameter search procedure, sampler and pruner. All other settings are set to the default defined in *Pytorch 2.6.0* and *optuna 4.6.0*.

7.1.1.4 Training

A final model will be trained using the estimated optimal hyperparameters (see the results in Section 7.2). This is done for an prolonged amount of epochs (100), or until an extended early stopping criterion (10) is met.

In order to increase the efficiency of the neural network estimation process, both during hyperparameter search and in the final training, the `torch.amp` package is used to activate half-precision floating point tensor datatypes and operations (float16 instead of float32) (Ahmed et al., 2022). This is a common strategy to reduce GPU memory load in training, allowing for the specification of larger batch sizes, thus enabling faster training. Empirical practice shows that model accuracy is generally not reduced by this approach (Micikevicius et al., 2018). Additionally, the `torch.compile` package (You et al., 2024; Wen, 2023) is used to JIT-compile both model code and loss computation, further increasing training speed.

7.1.2 Proposed predictors

Beyond standard predictors such as surprisal (Section 4.2.1), this work focuses on structural cost measures derived from the transformer model's attention scores. Based on the analogy between

memory retrieval and transformer attention (Section 5.4.2.1), a family of psycholinguistically motivated structural cost measures is defined:

- **attention entropy**, quantifying the dispersion of attention over the context,
- **attention distance**, a weighted distance with attention to less recent items leading to higher costs,
- **attention activation**, a time-based decay for context items that resets when they are attented and
- **attention difference**, quantifying the positional distance of attention over the context between two time steps.

In the following, these measures are referred to as *attention costs*. They will be formally defined and related to their theoretical motivations outlined in this work.

Intuitively, the first two attention costs aim to operationalise hypotheses from memory-based theories of sentence processing, in particular the idea that processing difficulty increases with retrieval competition or dependency distance. Attention activation and attention difference are more loosely inspired by prior work in the field and their inclusion is explorative. All four measures are formalised to be differentiable in order to develop loss terms based on them in experiment 2 (Section 8).

A transformer model can consist of a large number of individual attention heads spread over layers and multi-head attention modules. It is non-trivial to decide which attention heads to extract the measures from and how to combine them. This experiment aims to explore whether transformer language model attention generally exhibits patterns that predict human reading times, as opposed to identifying some specific head that correlates with RTs. Therefore, two methods for aggregating attention costs across all heads of a transformer model are presented in detail in Section 7.1.2.7 and tested subsequently. Secondly, the attention costs can be expected to increase with the size of the left context. Thus, normalised measures that control for trivial sentence-length effects are proposed in Section 7.1.2.6. Overall, 14 potential attenion-based reading time predictors are tested.

7.1.2.1 Attention distribution

Before specifying the four investigated methods for extracting memory-based costs, it is necessary to decide on an appropriate extraction procedure for the attention distributions. As explained in Section 5.4.2, previous research has proposed various modifications to the raw score matrix generated by dot-product attention, such as ignoring attention to the current word or correcting for the magnitude of value and output transformations.

In this study, norm-aware attention $\mathrm{Attn}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i} \,)$ (Definition 5.33) is computed for all attention heads $\langle\, l, h \,\rangle$ in the model as it showed to be the basis for the strongest one among the entropy-based predictors investigated by Oh and Schuler (2022). Entropy is given a preference in this choice compared to Oh and Schuler (2022)'s explorative distance measures due to multiple studies suggesting the reliability of a diffuseness effect (Ryu and Lewis, 2021; Ryu and Lewis, 2022; Oh and Schuler, 2022; Ryu and Lewis, 2025).

Note that, as an attention head $\langle l, h \rangle$ with $l > 1$ attends to representations that are themselves informed by context-weighted representations generated by the preceding attention layers $< l$, the resulting attention probability simplex does not quantify retrieval of lexical items but of intermediate representations. The extent to which these represent the respective context token, some higher-level construction or an aggregation of information necessary for next-token prediction likely needs further consideration in this line of work. Therefore, caution is advised when interpreting the distribution provided by $\mathrm{Attn}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i}\,)$ as a measurement of the contribution of the preceding tokens towards the head's output at the current processing step i or even simply as a measure of retrieval of the preceding tokens. Rather than that, every layer provides a sequence of opaque memory representations which are incrementally accessed by the subsequent layer and each token w_i is associated with the memory accesses occurring at processing step i.

Research suggests that retrieval-like operations might be occur in multiple specialised attention heads in transformer LMs, as discussed in Section 5.4.2.2. In Section 5.4.2.4 I outlined that traditional psycholinguistic theory is not compatible with the notion of multiple parallel memory retrieval operations, thus raising the question whether transformer attention behaves like a single retrieval operation fractured across heads or rather as multiple individual memory accesses. I aim to approach this issue (i) by extracting attention costs individually from each head and averaging the results and (ii) by first aggregating a single vector of attention weights over all heads in the transformer and then extracting a single cost on these weights.

The global attention aggregation across multiple heads is defined as follows:

Definition 7.1 (Aggregated attention distribution)**.** For a set S of pairs of layer and attention head indices, the *aggregated attention distribution over* S is defined as

$$\mathrm{Attn}^{\mathrm{N}}_{\mathrm{glob}:S}[\theta](\, w_i, w_{<i}\,) \coloneqq \frac{1}{|S|} \sum_{\langle l,h \rangle \in S} \mathrm{Attn}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i}\,).$$

This form of aggregation is inspired by Attanasio et al. (2022). However, in contrast to their proposal, a simple averaging is applied instead of the $\mathrm{softmax}$ function. As is trivial to see, $\mathrm{Attn}^{\mathrm{N}}_{\mathrm{glob}:S}[\theta](\, w_i, w_{<i}\,)$ sums to 1 and contains only positive elements, thus preserving the probability simplex.

7.1.2.2 Attention entropy

Cue-based retrieval theories predict similarity-based interference effects: when several memory representations partially match the retrieval cues, retrieval becomes noisy and slower (Van Dyke and Lewis, 2003; Lewis et al., 2006) (Section 4.3.2). Similarity-based interference is modelled as the entropy of the attention distribution (see Section 5.4.2.3). I adopt this measure in a slightly modified form:

Definition 7.2 (Attention entropy)**.** Given a transformer with attention head $\langle l, h \rangle$ and an input sequence $w_1 \ldots w_n$, for any position $i \in [n]$, the function $\mathrm{entr}^{\mathrm{N}}_{l,h}$ is defined as

$$\mathrm{entr}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i}\,) \coloneqq H_b(\mathrm{Attn}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i}\,)).$$

In contrast to Oh and Schuler (2022), I do not factor out the current item i since that would lead to competition of different magnitude producing similar entropy estimates. Assume that an item did

not match any of the context elements well but did attend strongly to its own representation, e.g. $\mathrm{Attn}^{\mathrm{N}}_{l,h}[\theta](\, w_5, w_{<5}\,) = \langle\, 0.01, 0.01, 0.01, 0.01, 0.96\,\rangle$. Removing the rightmost item and renormalising the remaining scores such as proposed by Oh and Schuler (2022) (Equation 5.36) would result in the simplex $\langle\, 0.25, 0.25, 0.25, 0.25\,\rangle$ with entropy amounting to the maximum possible for a four-element vector, suggesting strong competition among the representations in memory. Retaining attention to the current item could account for such variations in matching strength. The restricted entropy of Ryu and Lewis (2021) and Ryu and Lewis (2025) (Equation 5.31) is not chosen since it is unclear how it should be normalised for length in the fashion of Oh and Schuler (2022).

7.1.2.3 Attention distance

Dependency Locality Theory (Gibson, 2000) (DLT, Section 4.3.1) proposes that integration costs increase with the distance between dependent elements and the current item measured by the number of intervening referents (usually nouns and verbs). Establishing longer dependencies requires maintaining representations across more intervening material, leading to greater processing cost.

Since attention assigns continuous weights to prior positions, it does not provide a discrete set of retrieved items as implied by DLT. Furthermore, transformer LMs do not overtly distinguish between referents and non-referents. One option would be to implement an artificial threshold on the attention scores to identify retrieved tokens and to manually classify referents and non-referents to generate a DLT-style integration cost from transformers. However, choosing a suitable threshold is non-trivial and likely to interplay with context length. Additionally, making the metric computation dependent on the arg-maxed results of a referent classifier would introduce a non-derivable component, counteracting the second experiment.

For these reasons, a continuous proxy for integration distance is proposed via the expected backward distance under the attention distribution.[35]

Definition 7.3 (Attention distance)**.** Given a transformer with attention head $\langle\, l, h\,\rangle$ and an input sequence $w_1 \ldots w_n$, for any position $i \in [n]$, attention distance over head $\langle\, l, h\,\rangle$ is defined as

$$\mathrm{dist}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i}\,) \coloneqq \sum_{j=1}^{i} \mathrm{Attn}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i}\,)_{(j)}(i-j).$$

Larger values indicate longer-distance retrieval and are predicted to correlate with increased processing cost under DLT. Since the $\mathrm{softmax}$ function used in dot-product attention can be understood as a differentiable version of $\arg\max$ (Goodfellow et al., 2016, Chapter 6.2), $\mathrm{dist}^{\mathrm{N}}$ can be interpreted as approximating the distance to the item with the largest attention for each head.

As such, attention distance does not vary by the number of dependents attended to. Compare that to DLT where integration costs can be made up of more than one dependency to past items being established when encountering a new word. In transformer models, assuming that they access the same context items for next token prediction as warranted by DLT, attention could be either dispersed over the relevant dependents in one head or be distributed across multiple attention heads focusing on one dependent each.

[35] Again, there is no proof that transformers perform sampling from attention distributions. The term *expectation* is used for reasons of convenience here, as in slight notational abuse $\mathrm{dist}^{\mathrm{N}}_{l,h}(w_i, w_{<i}) = \mathbb{E}_{j \sim \mathrm{Attn}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i}\,)}[i-j]$.

7.1.2.4 Attention activation

Generally, psycholinguistic theory assumes that the activation of cognitive units decreases as a function of time since last retrieval, unless they are reactivated (Section 4.1.2). This is operationalised by models of sentence processing based on the cognitive architecture ACT-R (Lewis and Vasishth, 2005; Vasishth and Lewis, 2004) (Section 4.3.2.2) where the retrieval of items with low activation raises processing costs.

I propose an attention-based implementation of this principle: each token maintains a *degradation score* that increases the more time passes without attending to it. Time is measured in processing steps. Degradation can be understood to track an inverse of memory activation, with high degradation making retrieval harder. As retrieved and non-retrieved items cannot be delineated with soft attention scores, again a smooth and differentiable approximation is proposed.

First, a recency trace over attention matrices is defined:

Definition 7.4 (Degradation score)**.** For a vector $\mathbf{v}$, the degradation function degr is defined recursively as:

$$\begin{aligned} \mathrm{degr}(\mathbf{v})_{(1)} &:= 0 \\ \mathrm{degr}(\mathbf{v})_{(i)} &:= (1 - \mathbf{v}_{(i-1)})\, \mathrm{degr}(\mathbf{v})_{(i-1)} + (1 - \mathbf{v}_{(i-1)}). \end{aligned} \tag{168}$$

Intuitively, if position j is not attended to at $i-1$, its recency trace increases at i; otherwise, the trace is partially reset. The activation-based cost at position i is then defined as the expected recency under the current attention distribution.

Definition 7.5 (Attention activation)**.** Given a transformer with attention head $\langle\, l, h \,\rangle$ and an input sequence $w_1 \ldots w_n$, for any position $i \in [n]$, the function $\mathrm{act}^{\mathrm{N}}_{l,h}$ yields the *attention activation over head* $\langle\, l, h \,\rangle$ defined as

$$\mathrm{act}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i} \,) := \sum_{j=1}^{i} \mathrm{Attn}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i} \,)_{(j)}\, \mathrm{degr}\left(\begin{pmatrix} \mathrm{Attn}^{\mathrm{N}}_{l,h}[\theta](\, w_j, w_{<j} \,)_{(j)} \\ \vdots \\ \mathrm{Attn}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i} \,)_{(j)} \end{pmatrix} \right)$$

This predictor is large when attention is allocated to items that have not been recently attended, corresponding to retrieval of low-activation representations. Importantly, it captures temporal recency rather than purely structural distance: a representation may be linearly close yet inactive, or distant yet frequently refreshed.

7.1.2.5 Attention difference

If contextual representations are stored sequentially in memory, reallocating attention across distant positions may incur additional processing cost. In attention-based models, each timestep redistributes retrieval weight over prior tokens, effectively shifting a soft pointer along a linear memory (Oh and Schuler, 2022). I quantify this shift as the expected displacement between consecutive attention distributions:[36]

[36] Again, in abuse of notation $\mathrm{diff}^{\mathrm{N}}_{l,h}(w_i, w_{<i}) = \mathbb{E}_{j \sim \mathrm{Attn}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i} \,), k \sim \mathrm{Attn}^{\mathrm{N}}_{l,h}[\theta](\, w_{i-1}, w_{<i-1} \,)}[|j-k|]$.

Definition 7.6 (Attention difference)**.** Given a transformer with attention head $\langle\, l, h \,\rangle$ and input sequence $w_1 \ldots w_n$, for any position $i \in [n]$, the function $\mathrm{diff}^{\mathrm{N}}_{l,h}$ yields *attention difference over head* $\langle\, l, h \,\rangle$ and is defined as

$$\mathrm{diff}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i} \,) \coloneqq \begin{cases} \sum_{j=1}^{i} \sum_{k=1}^{i-1} \mathrm{Attn}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i} \,)_{(j)} \cdot \mathrm{Attn}^{\mathrm{N}}_{l,h}[\theta](\, w_{i-1}, w_{<i-1} \,)_{(k)} \cdot |j-k|, & i > 1 \\ 0, & i = 1. \end{cases}$$

This cost is high at step i if the change in location of attention in the memory (including the current item) from step $i-1$ to step i is large (i.e. large surface distance). It is lowest, if w_i attends to the same item as w_{i-1}. Movements to the left and to the right have the same influence on the cost if their step size is equal.

Intuitively, this quantity is small when attention remains concentrated on similar memory positions across time-steps, and increases as attention shifts toward more distant tokens. Unlike earth mover's distance, explored by Oh and Schuler, 2022 (Definition 5.43), this formulation captures overall dispersion in retrieval focus rather than the minimal work required to transform one distribution into the other. In particular, it assigns non-zero cost to dispersed but identical distributions, penalising diffuse attention.

7.1.2.6 Normalisation

Since all proposed metrics increase with larger left contexts under certain conditions (cf. Equations 117 et seqq. for entropy; this holds analogously for the other costs), motivated by Oh and Schuler (2022), I define length-normalised variants that divide each measure by its maximal attainable value for the respective context size. This maps each cost to the interval $[0, 1]$.

For attention entropy, the method of Oh and Schuler (2022) is adopted: the maximum occurs under a uniform distribution over $[i]$ and is therefore $\log i$.

Definition 7.7 (Normalised attention entropy)**.** Given a transformer with attention head $\langle\, l, h \,\rangle$ and an input sequence $w_1 \ldots w_n$, for any position $i \in [n]$, the function $\widetilde{\mathrm{entr}}^{\mathrm{N}}_{l,h}$ is defined as

$$\widetilde{\mathrm{entr}}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i} \,) \coloneqq \frac{1}{\log i} \mathrm{entr}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i} \,).$$

For attention distance at position i, the maximum occurs when all attention is on the leftmost position, yielding a cost of $i-1$. For activation-based cost, the degradation trace of a position that has never been attended grows linearly with its distance from the current position. The maximal activation cost is thus likewise obtained by attending fully to the oldest, maximally degraded position, yielding $i-1$. Finally, for attention difference, the maximum occurs when attention shifts between opposite extremes of memory in consecutive steps. In this case, the absolute displacement equals $i-1$. Thus, in all three cases, the normalised metric is gained by dividing the measure by $i-1$, which is analogous to Oh and Schuler, 2022's normalisation of EMD (Definition 5.43).

Definition 7.8 (Normalised attention distance/activation/difference)**.** Given a transformer with attention head $\langle\, l, h \,\rangle$ and an input sequence $w_1 \ldots w_n$, for any position $i \in [n]$, the function $\widetilde{\kappa}^{\mathrm{N}}_{l,h}$ is defined as

$$\widetilde{\kappa}^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i} \,) \coloneqq \frac{1}{i-1} \kappa^{\mathrm{N}}_{l,h}[\theta](\, w_i, w_{<i} \,)$$

for $\kappa \in \{ \mathrm{dist}, \mathrm{act}, \mathrm{diff} \}$.

7.1.2.7 Aggregation

Two modes of aggregation over attention heads are distinguished. First, analogous to Definition 5.39, *local* attention costs for a selection of attention heads S are defined as the average attention cost over S. By extension, a local attention cost over a transformer is defined as the local attention cost over all of its heads.

Definition 7.9 (Local transformer attention cost)**.** Given a transformer, a set of layer-head pairs S, and an input sequence $w_1 \ldots w_n$, for any position $i \in [n]$, the function $\kappa^{\mathrm{N}}_{\mathrm{loc}:S}$ yields the *local attention entropy/distance/activation/difference over* S defined as

$$\kappa^{\mathrm{N}}_{\mathrm{loc}:S}[\theta](\, w_i, w_{<i} \,) \coloneqq \frac{1}{|S|} \sum_{s \in S} \kappa^{\mathrm{N}}_{s}[\theta](\, w_i, w_{<i} \,),$$

for $\kappa \in \{ \mathrm{entr}, \mathrm{dist}, \mathrm{act}, \mathrm{diff}, \widetilde{\mathrm{entr}}, \widetilde{\mathrm{dist}}, \widetilde{\mathrm{act}}, \widetilde{\mathrm{diff}} \}$.

For a transformer consisting of k layers with m attention heads each, *local attention entropy/distance/activation/difference over the transformer* is defined as

$$\kappa^{\mathrm{N}}_{\mathrm{loc}}[\theta](\, w_i, w_{<i} \,) \coloneqq \kappa^{\mathrm{N}}_{\mathrm{loc}:\{ \langle l,h \rangle | l \in [k], h \in [m] \}}[\theta](\, w_i, w_{<i} \,),$$

for $\kappa \in \{ \mathrm{entr}, \mathrm{dist}, \mathrm{act}, \mathrm{diff}, \widetilde{\mathrm{entr}}, \widetilde{\mathrm{dist}}, \widetilde{\mathrm{act}}, \widetilde{\mathrm{diff}} \}$.

In *local* attention costs, the respective measure is computed separately for each head in the model and then averaged over all heads. I hypothesise that this appraoch preserves functional specialisation across heads.

For the alternative *global* costs, attention is first aggregated across all heads, quantifying the overall weights which the transformer assigns to representations in the context when computing the prediction of the next item. Finally, the attention cost is computed just once on this aggregated distribution.

Definition 7.10 (Global transformer attention cost)**.** Given a transformer, a set of layer-head pairs S, and an input sequence $w_1 \ldots w_n$, for any position $i \in [n]$, the function $\kappa^{\mathrm{N}}_{\mathrm{loc}:S}$ yields the *global attention entropy/activation/difference over* S defined as

$$\kappa^{\mathrm{N}}_{\mathrm{glob}:S}[\theta](\, w_i, w_{<i} \,) \coloneqq \kappa^{\mathrm{N}}_{\alpha,\beta}[\theta](\, w_i, w_{<i} \,),$$

for $\kappa \in \{ \mathrm{entr}, \mathrm{act}, \mathrm{diff}, \widetilde{\mathrm{entr}}, \widetilde{\mathrm{act}}, \widetilde{\mathrm{diff}} \}$, where $\kappa^{\mathrm{N}}_{\alpha,\beta}[\theta](\, w_i, w_{<i} \,)$ is the attention cost applied to $\mathrm{Attn}^{\mathrm{N}}_{\alpha,\beta} \coloneqq \mathrm{Attn}^{\mathrm{N}}_{\mathrm{glob}:S}$.[37]

For a transformer with k layers with m attention heads each, *local attention cost over the transformer* is defined as

$$\kappa^{\mathrm{N}}_{\mathrm{glob}}[\theta](\, w_i, w_{<i} \,) \coloneqq \kappa^{\mathrm{N}}_{\mathrm{glob}:\{ \langle l,h \rangle | l \in [k], h \in [m] \}}[\theta](\, w_i, w_{<i} \,),$$

for $\kappa \in \{ \mathrm{entr}, \mathrm{act}, \mathrm{diff}, \widetilde{\mathrm{entr}}, \widetilde{\mathrm{act}}, \widetilde{\mathrm{diff}} \}$.

[37] Note that the head $\langle \alpha, \beta \rangle$ is just a temporary assignment that allows us to reuse the established notation for cost functions.

Note that there is no separate global version of (normalised) attention distance as difference computed on an aggregated attention distribution $\mathrm{Attn}^{\mathrm{N}}_{\mathrm{glob}:S}[\theta](\, w_i, w_{<i}\,)$ is equal to average of attention distances over the individual heads, as shown in Equation 169. This holds equivalently for the scaled variant.

$$\begin{aligned}\mathrm{dist}^{\mathrm{N}}_{\mathrm{loc}:S}[\theta](\, w_i, w_{<i}\,) &= \frac{1}{|S|}\sum_{s\in S}\mathrm{dist}^{\mathrm{N}}[\theta](\, w_i, w_{<i}\,) \\ &= \frac{1}{|S|}\sum_{s\in S}\sum_{j=1}^{i}\mathrm{Attn}^{\mathrm{N}}_s[\theta](\, w_{<i}\,)_{(j)}(i-j) = \sum_{j=1}^{i}\frac{1}{|S|}\sum_{s\in S}\mathrm{Attn}^{\mathrm{N}}_s[\theta](\, w_{<i}\,)_{(j)}(i-j) \\ &= \sum_{j=1}^{i}\mathrm{Attn}^{\mathrm{N}}_{\mathrm{glob}:S}[\theta](\, w_i, w_{<i}\,)(i-j). \quad (169)\end{aligned}$$

Altogether, the set of potential reading time predictors to test includes 14 functions: four local unnormalised, four local normalised, three global unnormalised and three global normalised costs.

7.1.2.8 Untokenisation

While the language model treats white-space separated regions mostly as single token, there are cases of finer tokenisation, such as the possessive marker *'s* being treated as its own token (Section 7.1.1.2). For these multi-token words, reading time predictors will be generated by combining the measures of their components.

For surprisal, this is done by summing the surprisal of the subwords (see Section 4.2.1). For the attention costs, the mean over subword token costs is taken. A second review of the relevant literature after the experiments revealed that taking the maximum among subwords might have been a better compromise between summing and averaging (see Section 5.4.2.5). However, since the tokenisation strategy in this work leads to only a small number of multi-token words, this is issue likely does not diminish the validity of subsequent results.

7.1.3 Psycholinguistic evaluation

The attention-based predictors defined in the preceding section will be individually tested for their effect on human reading times using likelihood ratio tests (Section 6.3.7) in a standard linear mixed model paradigm (Section 6).

The contribution of a predictor will be evaluated in terms of its ability to explain variability in reading times beyond baseline effects. In each case, the null-hypothesis is that reading time is not linearly dependent on the candidate measure. In order to check this hypothesis, (i) a null-hypothesis model without the candidate as a fixed effect (but possibly with random slopes for the candidate to account for repeated measurements) and (ii) an alternative model with the same design but including the candidate as a fixed effect are fitted. Then, both models are compared via a likelihood ratio test. This allows to determine whether its effect significantly exceeds random group-level variability. The null-hypothesis is rejected if the p-value is below a significance threshold of $c = 0.05$, with 0.05 being chosen out of convention, since I do not perform a power analysis beforehand that would allow to select a more appropriate threshold. The rejection of the

null hypothesis will be interpreted as indicating an effect of the candidate measure on reading times.

All model designs used for statistical inference are specified to allow for reproduction and to verify the suitability of the LMEM design. Additionally, an `R` notebook of the model selection process can be found at https://github.com/filemon11/MITransformer.

7.1.3.1 Data

Two main types of data are commonly used for exploring human sentence processing while reading: self-paced reading times (Section 3.3) and eye-tracking measurements (Section 3.2). Due to the two paradigms capturing different aspects of processing (see Section 3.3.7), this study initially aimed to evaluate the experiments on both kinds of corpora. Furthermore, in order to increase the power of the analysis, I also planned to perform evaluations on multiple datasets of the same paradigm. In order to minimise variability among subjects and confounding factors, the datasets considered were narrowed down to corpora that aim to measure "normal" reading (see Section 3.1.2, e.g. excluding sentiment analysis such as ZuCo 1.0 task 1) and are collected mainly from skilled English L1 speakers. The resulting selection is presented in Sections 3.3 and 3.2.

Unfortunately, performing the tests on multiple datasets turned out to be infeasible in the available time frame. Thus, I limited the analysis to the Provo corpus (Section 3.2.7.4). The reason for choosing an eye-tracking corpus over a self-paced reading time dataset is two-fold:

Firstly, the only publicly available SPR corpora matching the requirements outlined above are the Natural Stories Corpus and UCL (Sections 3.3.6.4 and 3.3.6.3). Natural Stories is omitted since it is specifically designed to consist of many low-frequency syntactic structures such as non-local constructions and prepositional phrase attachment ambiguity (Futrell et al., 2018). As such, its makeup is not representative of naturally occuring text and therefore not a suitable choice to test for a general effect of a potential predictor on RTs as is intended here. However, the dataset could be valueable to conduct comparisons between promising but similar predictors in future work, whose contributions on regular data cannot be teased apart.

The SPR portion of the UCL corpus is also discarded. While it has a large number of participants (117), the SPR portion of the corpus exhibited weaker effects of syntactic integration than the eye-tracking part in a previous study (Mielczarek et al., 2025).

Among the eye-tracking corpora readily available, the Provo corpus sets itself apart due to its large number of participants for a dataset of this kind (84). As eye-movements-while-reading exhibit a large space of possible reading strategies (see Section 3.2), the variation in this type of data is typically strong. The higher the number of participants, the greater the probability that population-level effects are not due to an unfortunate sampling of participants with a certain reading strategy.

Additionally, in contrast to corpora such as GECO (Section 3.2.7.3) in which participants were tasked to read a whole novel, Provo consists of small paragraphs of two to three sentences. Furthermore, the paragraphs were shown in random order to the participants, decreasing the risk of participants reconnecting paragraphs into a larger contexts. Both aspects can be expected to reduce the effects of extra-sentential context which could confound statistical inference. Therefore, the analyses in this thesis will be conducted using the Provo corpus.

Eye-tracking makes available several aggregate measures – early features such as first-fixation duration and composite features such as go-past time (Section 3.2.4). Previous work has sug-

gested that go-past time, which includes the gaze duration on the current word but also the time spent on regressions to the past is most indicative of high-level integrative processing (Radach and Kennedy, 2013). Studies on attention entropy appear to confirm this, finding a stronger effect on go-past times and only a delayed effect on first-fixation duration (Ryu and Lewis, 2025). Therefore, it is fair to assume that attention costs aimed at quantifying memory processes such as those investigated here, have a higher chance of affecting go-past time. This motivates its use as the response variable for the LMEM analyses.

Out-of-domain evaluation As is standard, transformer LM training and the pycholinguistic evaluation of transformer-derived measures are conducted on different datasets. This is motivated by practical and theoretical considerations.

Firstly, the psycholinguistic corpus is significantly smaller than WikiText 103 (2689 tokens vs 103 million tokens), to the extent that it is impossible to train a transformer model of the intended size without leading to heavy overfitting. Secondly, if one were to assume that humans have available a distribution over next tokens in some form or another, influencing speed while reading a text (see Section 4.2.1), it would not be derived from that very text. While avoiding any assumptions about the precise nature of language acquisition processes, it is likely that this hypothetical human LM was shaped by prior interaction or consumption of language, justifying a similar procedure for the neural LM.

Therefore, the issue of out-of-domain evaluation reduces down to the question of whether the language model training data used is representative of human language. This is not to be confused with the data being developmentally plausible (see for instance the BabyLM challenge (Warstadt et al., 2023)). It can be stated without doubt that WikiText 103 is not representative of the material available to humans in their process of language acquisition – missing for instance spoken language and easy texts. However, this is not the concern here as transformer models are not a model of human language acquisition. The actual concern is whether WikiText 103 is broadly indicative of the already established mental model a skilled reader participating in the Provo eye-tracking study might maintain.

This is a very broad requirement and hard to validify. As the WikiText corpus is large and encompasses texts on various subjects, it is unlikely to be biased topic-wise. Additionally, Wikipedia is directed towards a general audience. Nonetheless, by nature of its purpose, it consists of only one genre of text and contains many technical terms (precisely those that an encyclopedia aims to explain) that taken together surpass any single human's knowledge. Thus, disparity between the theoretical human model of language and an artificial neural network trained on WikiText is to be expected.

This divergence is likely to slightly bias surprisal and attention costs generated for the Provo dataset using the trained LM, possibly leading to underprediction of surprisal for named entities (see Section 5.4.1.1). The effect on attention costs is unclear but might be similar. Possibly key and query representations are overly adjusted to terms and patterns in the dataset, reducing competition effects. However, this is rather speculative.

Overall, the approach adopted here is methodologically justified due to the selection of a large and thematically diverse dataset for language model training. However, interpretations of the results must take into account the possible biases it introduced on the predictors.

Preprocessing Before estimating linear-mixed effects models, the Provo dataset undergoes data-cleaning and preprocessing. First, the last word of each sentencs is removed to filter out wrap-up effects (Section 3.2.6). Then, observations of skipped words are excluded from the analysis instead of assigning a go-past time of 0 to them. The underlying assumption is that skipped words do not trigger integrative processing whose nature this study aims to investigate.

Extreme values are removed to exclude observations that do not reflect lexical processing, e.g. arising due to measurement errors or distracted participants. This is done using the standard method of first applying strict cut-offs and then conducting statistical outlier removal (Eskenazi, 2024).

A lower cut-off of 80 ms is applied according to standard procedure, as it is believed that this is the minimal time needed to register a word (Eskenazi, 2024). Conveniently, Provo already does not provide GPT values smaller than 81 ms (Luke and Christianson, 2018). An upper cut-off of 3000 ms is chosen based on the modal cap observed by Eskenazi (2024) in previous research (1000 ms), multiplied by 3 to account for the fact that they included studies working with shorter measures like first-fixation duration. 3000 ms is also the upper cut-off used by Ryu and Lewis (2025) for GPT. The Provo curators removed fixations larger than 800 ms, meaning that 3000 ms GPT would comprise of almost four maximal fixations (or multiple shorter ones) which can be safely regarded as outliers (Luke and Christianson, 2018).

Participant-wise outlier removal is conducted by calculating the standard deviation of GPT for each group of measures associated with the same participant and then removing all entries lower or larger than the mean plus/minus 2.5 times the participant's standard deviation. 2.5 is the mean factor observed by (Marsden et al., 2018) in previous work.

Sentences that contain words unknown to the language model are removed from the analysis instead of adopting the standard practice of removing only the respective word. This is due to an unknown token possibly having effects on the attention patterns of the succeeding tokens.

Lastly, observations with a word logarithmic word frequency below 2 (words occurring less than 100 times per 1 billion words) are removed since manual inspection showed that such low probabilities likely arise from formatting errors and special symbols which teh the word frequency library cannot properly deal with (see Section 7.2.3.1).

Overall, the steps outlined above lead to the following reductions, starting with a total number of 212 436 observations in Provo:

1. removal of skipped-over words: 72 968 measurements,
2. upper cut-off: 411 measurements,
3. frequency cut-off: 588 measurements,
4. outlier removal: 3949 measurements,
5. removal of sentences that include words unknown to the LM tokeniser: 7896 measurements (19 words in 18 sentences)

This results in an overall number of 122 256 observations after data cleaning.

Finally, all predictors are standardised (z-transformed, i.e. their mean is subtracted and they are divided by their standard deviation) to aid the LMEM optimisation process.

7.1.3.2 Model selection

In order to reduce Type I error rates, model selection aims to include the maximal random effects structure supported by the data and justified by the study design (see Section 6.4.2 for a discussion on this topic). The procedure adopted here is follows: first, a set of baseline predictors is selected. Then, multiple hierarchies of random effect groupings are identified and a maximal model, including random intercepts for all identified levels and random coefficients for all predictors that vary within the respective groupings, is specified. The model is fit and simplified iteratively as long as convergence or singularity issues occur.

Simplification proceeds gradually following Bates et al. (2018), first by removing correlation parameters and only then, if necessary, by removing variance components that are not supported by the data. In each step, a principle component analysis on the random-effects variance-covariance estimates is performed to identify the number of variance components supported (Bates et al., 2018). In cases of convergence or singularity issues, the number of components is appropriately reduced.

The design of the final models used for inference will be given in the results section (Section 7.2) along with information regarding the simplifications that were necessary to obtain them, following the advice of Barr et al. (2013). Providing this information is important to allow to judge the generalisabiltiy of the statistical tests.

Check of assumptions Initially, collinearity and multicollinearity between potential predictors will be checked by computing pairwise Pearson correlations and computing the variance inflation factor. This is important to prevent issues in the estimation process (Section 6.3.6).

Five assumptions are checked for the LMEMs used for statistical inference (Sections 6.2.1 and 6.3.4). In the following, the assumptions are listed alongside the tools that are used to diagnose them visually:

1. homogeneity of variances: plotting predictors against residuals,
2. normality of residuals: Q-Q plot of the residuals,
3. independence of errors: this is asserted based on the study design. Repeated measurements violate this assumption. A maximally supported random effects structure is assumed to account for this assumption,
4. normality of random effects: Q-Q plots of the random effects.

Additionally, linearity is diagnosed by plotting the predicted reading times against the residuals to ensure that a suitable model class is used (Section 6.2.1).

Selecting a baseline Before fitting attention costs to reading times, a strong baseline model accounting for established RT predictors and grouped effects by participants and stimuli is selected. This is done in three steps: (i) first, a set of baseline predictors is chosen, (ii) then, random effects are introduced and finally (iii) lagged (spillover) variables are added. In each of these steps, the LMEM structure can reach overspecification, in which case it is simplified following the strategy outlined above.

Word frequency and word length (established contributors are computed for the regions of interest in the Provo corpus and included as baseline predictors due to their known strong contribution to reading times (Section 3.2.6). Word frequency is calculated using the `wordfreq` package (Speer, 2022), producing a logarithmic frequency. The library splits some words such as compound modifiers (e.g. "ill-defined") into multiple tokens and combines their frequencies using the half-harmonic-mean function. Word length will be extracted by counting the number of characters of a region of interest.

The inclusion of surprisal (Section 4.2.1) might make it difficult to identify the nature of the processes that lead to predictability effects. Concretely: if surprisal is a reflection of the memory effects investigated here, for instance with larger attention entropy leading to (or at least correlating with) higher surprisal and lower entropy in cases of low surprisal, then it is not only hard to identify attention entropy effects beyond surprisal, but already ill-motivated to do so and to employ the LMEM design proposed here (Slaats and Martin, 2025).

However, recent studies have found an effect of attention entropy beyond surprisal (Oh and Schuler, 2022; Ryu and Lewis, 2022; Ryu and Lewis, 2025) and propose entropy as reflecting a process distinct from prediction (Ryu and Lewis, 2022). In particular Ryu and Lewis (2025) suggest surprisal to reflect an effect of probabilistic expectations in lexical identification and attention entropy to indicate later stages of sentence processing. This hypothesis is extended here to the other three potential predictors, distance, activation and difference, on the grounds of all of them relying on the analogy between memory retrieval and attention scores. This motivates testing them in the presence of surprisal.

In addition to word length, frequency and surprisal, I check whether to include features tracking temporal reading progression: word position inside a sentence, sentence position inside a paragraph and paragraph position inside the corpus.These can be interpreted as tracking exhaustion or the decrease in attention paid by the participants. Additionally, the position of a word in its sentence also tracks the size of the available left context – i.e. the number of elements that a subject may have saved in memory or which they can regress to visually.

Grouping levels For the grouping variables, besides per-participant effects, two nested hierarchies are considered:

- `lemma` → `word` and
- `paragraph` → `sentence` → `abs.pos`,

where `abs.pos` corresponds to one specific token occurrence (token instance) and `word` refers to a word form (token type). The second hierarchy is considered since the temporal features presented above lack significance as fixed effects and lead to convergence issues. They might still explain reading time variability, but possibly a non-linear fashion.

Reading times might vary for words due to factors other than frequency, length or surprisal – for instance due to their meaning, motivating the inclusion of token type as a grouping level. However, the number of unique surface forms in the corpora is very large (1052), in part due to punctuation being treated as part of the tokens and due to capitalisation. For this reason, grouping word forms according to their lemmata is explored. These are generated in sentential contexts for the word-level tokenised sentences (see Section 7.1.1.2) using the `nltk.stem.WordNetLemmatizer`

(Bird et al., 2009) which queries the WordNet database (WordNet, 2010). For regions of interest that were separated into multiple tokens, a dependency parse with `spacy` (Honnibal et al., 2020) is used to determined the lemma of the region's head, or if not applicable, the lemma of its first component ignoring punctuation.

Spillover After finding a maximal model for the baseline predictors and the given grouping variables, I check for possible *spillover*-effects (Section 6.4.3). Processing slowdown is often delayed in reading time data (Ehrlich and Rayner, 1983). This needs to be accounted for in order to ensure that the analyses are not compromised by spurious correlations.

Therefore, I consider shifted versions of the predictors following the procedure outlined in Section 6.4.3. Spillover is applied paragraph-wise, i.e. for each sentence in a paragraph of Provo, the spilled-over value assigned to the first word originates from the last word of the preceding sentence. In the first sentence of each paragraph, the first k observations are removed for a spillover window of k. A pre-predetermined maximal threshold of 3 is applied in the estimation procedure since a larger spillover window could not be attested in literature and could potentially remove too much relevant data by cutting off the first observations in a paragraph.

Fitting predictor models For each of the 14 attention costs to test, a null-hypothesis model and a alternative model are devised. The alternative model is determined by adding the attention cost as a fixed effect and as a random effect in all grouping levels justified by the design. In cases of convergence or singularity issues, the model structure is simplified adopting the same strategy sketched above. The null-hypothesis model corresponds to the alternative but with the fixed-effect for the attention cost removed. Random effects for the predictor to test are retained to account for random group-based variability.

7.1.3.3 Statistical inference

For each attention cost variant, the null-hypothesis is that go-past time is not linearly dependent on the candidate measure. In order to check this hypothesis, the null-model and the alternative model are both fitted to reading times and then compared via a likelihood ratio test (Section 6.3.7). The null-hypothesis is rejected if the p-value is below the significance threshold $c = 0.05$.

Multiple testing correction Since the experiment consists of 14 likelihood ratio tests against the same baseline model (minus candidate-based random coefficients), correction for multiple comparisons is conducted using the BH procedure (Section 6.4.4). It is chosen instead of the Bonferroni correction due to its greater sensitivity. The requirement of positive correlations between test statistics is assumed to be given here since the predictors exhibit strong correlations (Table 17) and since the hypotheses are not assumed to be mutually exclusive.[38]

Additional reported quantities In addition to the p-value of the likelihood ratio test and the BH-corrected p-value, for each predictor the change in *Akaike Information Criterion* (AIC) and *Bayesian Information Criterion* (BIC) between null model and alternative model are reported.

[38] For instance, entropy effects could be paired with activation decay as in ACT-R.

These are scoring methods commonly used for model selection (Wasserman, 2004, Section 13.6) and will be denoted as Δ_{AIC} and Δ_{BIC} respectively. A negative value of $\Delta_{\text{AIC}}/\Delta_{\text{BIC}}$ signifies that the alternative model is preferred. Furthermore, the standard error, the χ^2 statistic and the estimated effect sizes are provided.

7.1.3.4 Controlling for position effects

When selecting the baseline LMEM, sentence position was non-significant as a fixed effectHowever, manual inspection after performing the statistical tests for the candidate attention predictors showed that they tend to increase with sentence position. Therefore, a second set of analyses for the four strongest attention cost variants is conducted, controlling for sentence position effects.

This is done by re-including position as a fixed effect. Two variants are separately tested: inclusion of position as a linear effect and as a logarithmic effect due to attention entropy showing a tendency to increase logarithmically with sentence position and the other costs linearly. See Section 7.2 for a in-depth analysis. The testing procedure is the same as in the main test (Section 7.1.3.3). BH correction is applied separately under the new stricter specification for 8 LRTs.

7.2 Results

This section presents the results of the first experiment. First, I outline the findings of the language model hyperparameter search and the final training. Then, the LMEM selection process is summarised. Finally, the results of the main statistical tests are given. The full training scripts that allow for reproduction (including specification of seeds for random operations) can be found in the repository at `https://github.com/filemon11/MITransformer`.

7.2.1 Hyperparameter search

The initial sample space of hyperparameters was given in in Table 6. As outlined in Section 7.1, the initial model architecture follows that of `GPT-2 medium`, with 24 attention layers, 16 parallel heads and dimensionality 1024.

After 40 search trials, the estimation script unfortunately ran into a storage issue which resulted in its abortion. Before initiating the execution of the remaining 60 trials, I adjusted the search spaces slightly to account for maxima occurring at borders. Specifically, the lowest validation losses accumulated at the maximum allowed factor $f_{FF} = 8$, which is why the right boundary was increased to 12 for the second run. A similar phenomenon could be observed for feed-forward dropout where I increased the right boundary from 0.5 to 0.75. On the other hand, trials with learned positional embeddings vastly outperformed sinusoidal encodings. Due to the conclusiveness of the result, I excluded sinusoidal encodings from the second run. Similarly, the upper boundary for residual dropout was lowered from 0.5 to 0.4 due to strongly increasing losses above the 0.3-mark.

Additionally, the dimensionality of the model was reduced from 1024 to 768, comparable to `GPT-2 small`. This was necessary to adjust the optimisation process to changes in the availability of computational resources for this project. The number of heads and layers was not changed. The second search run was conducted for 48 hours, the maximum time limit for a single job on the HHU Hilbert cluster, resulting in the overall number of 109 trials. The updated search space for

Parameter	Run 1	Run 2
Trials	40	69
Dimensionality	1024	768
Batch size	1920 (640 · 3 GA*)	1920 (320 · 6 GA*)
Resources	Google Cloud	HHU HPC Hilbert
Device	8 Nvidia A100 GPUs	
Threads (dataloader workers)	8 per process	
Processor	unknown	AMD EPYC 75F3, 2.95 GHz
Peak RAM usage	unknown	117.89 Gb DDR4
Duration	32.03 h	47.31 h

Table 8: Computational resources used for the hyperparameter search. The processor and RAM of the Google Cloud machine could not be reported due to deletion of the virtual machine. Despite using the same GPU architecture, batch size had to be reduced in run 2 since it otherwise led to memory errors. *gradient accumulation, i.e. combining gradients of several batches and running optimisation steps in intervals.

the second run is specified in Table 4. Table 8 gives an overview of the computational resources used.

In Figures 6–9 the sampled hyperparameters are plotted against the respective best reported loss on the validation split. A trial is marked as finished if it reached the maximum number of epochs or if it was stopped due to having reached a plateau. The remaining trials were stopped by the hyperband pruner. The trials are differentiated by training run and the best trial is marked with a diamond symbol.

The learning rate (Figure 5) is reported using an x-axis in log-scale to reflect the sampling space. It can be seen that learning rates between 10^{-4} and $10^{-3.5} \approx 3.2 \cdot 10^{-4}$ yield the best results, with the lowest loss achieved on a learning rate of $1.92 \cdot 10^{-4}$. Runs with lower learning rates are all pruned. There are multiple finished runs for higher rates but these tend to produce higher losses.

For dropout (Figures 6–8), no clear tendency can be observed and the loss can reach values close to the observed minimum for a wide range of dropout parameters. However, it is possible to identify an upper boundary for each dropout parameter above which regularisation was too strong and training was inhibited: approximately 0.3 for residual dropout, 0.6 for dropout in the feed-forward layer and 0.4 for the embedding layer.

It is possible that training was not long enough for the beneficial effects of regularisation to be fully noticeable in this hyperparameter search. Regularisation serves to reduce overfitting, meaning that convergence on the evaluation split is reached at a lower loss but also later during training. Therefore, I refrain from concluding from the lack of patterns that the inclusion of dropout has no positive influence on the loss and that lack leaving dropout out entirely would produce equally performant models. Based on this selection process and on previous work supporting the inclusion of dropout (e.g. Vaswani et al., 2017; Mielczarek et al., 2025), I choose the moderate dropout values reported as optimal by the hyperparameter search: 0.15 for embeddings, 0.32 for feed-forward layers and 0.17 for residual dropout.

Finally, there is a clear picture with respect to bias terms. Figure 9 shows that the finished trials with bias terms score consistently below a loss of 3.5 while leaving out the bias yields results in

Parameter	Range
f_{FF}	$\{6, \ldots, 12\}$
Dropout	
– Embedding	[0,0.5]
– Feed-forward	[0,0.75]
– Residual	[0,0.4]
Learning rate	$[6 \times 10^{-5}, 6 \times 10^{-4}]$

Figure 4: Updated search space for hyperparameter optimisation run 2.

Property	Value
Trial	84
Validation loss	3.42
Validation perplexity	30.67
Early stop after	7 epochs
Parameter count	482 590 464
f_{FF}	9
Dropout	
– Embedding	0.14735
– Feed-forward	0.32006
– Residual	0.16950
Positional encoding	embedding
Learning rate	1.9223×10^{-4}

Table 9: Best configuration found during hyperparameter search in experiment 1.

the interval between 3.5 and 3.7 – plus two outliers at approximately 5.7 and 7. The optimal model includes the bias term which is consistent with this trend.

Overall, the hyperparameter search found its optimal model at the 84th trial with a validation loss of 3.42 and a perplexity of 30.67. The corresponding hyperparameters are given in Table 9. Interestingly, the identified feed-forward factor f_{FF} (9) is more than twice as large as the factor of 4 employed by `GPT-2`. The model seems to profit from a greater capacity for position-wise transformation applied to the output of transformer layers. Note that the model has a larger overall dimensionality than `GPT-2 medium` (483 million vs 355 million parameters) due to the large embedding matrix for word-based tokens used in this work.

7.2.1.1 Check for convergence

In order to investigate the possibility that results are suboptimal due to the neural networks not reaching convergence, a manual investigation of training dynamics with respect to training and validation loss was conducted. While most trials that finished (i.e. those that reached the maximum number of epochs or were stopped early due to three consecutive evaluations without improvement of the best reported score) stagnated for several evaluation steps before stopping, trials 41, 42, 43 and 45 which all ran until the maximum number of epochs, still exhibited slow progress at the latest stages of training. Therefore, while it is unlikely that the general results of the hyperparameter search would change drastically if longer training was allowed, this motivates conducting an extended training run using the optimal found hyperparameter configuration to allow the network to tune its parameters further. The results are described in the next section.

7.2.2 Language model training

The final model was trained without an early stopping criterion based on the optimal hyperparameters identified in Section 7.2.1. Training modalities can be found in Table 11; the model was optimised for a total of 38 epochs but reached minimum validation loss at epoch 8.8. A plot of

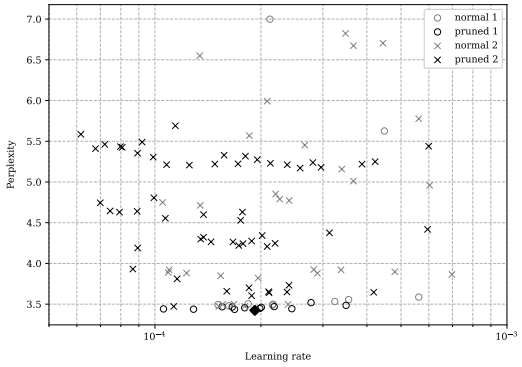


Figure 5: Learning rates sampled during hyperparameter search plotted against the minimal validation loss reported for the search trials; the x-axis is in log-scale.

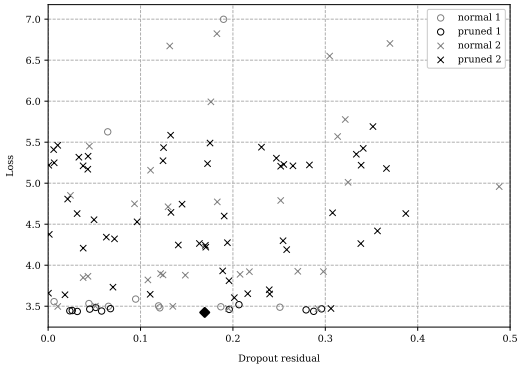


Figure 6: Residual dropout sampled during hyperparameter search plotted against the minimal validation loss reported for the respective search trials.

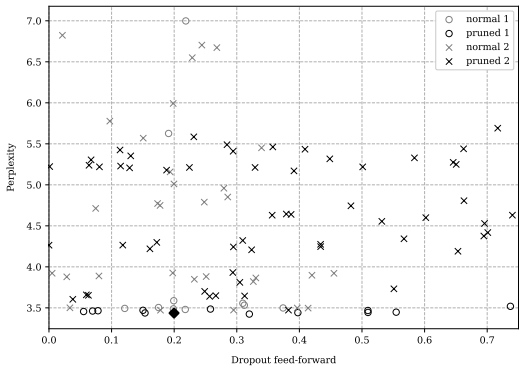


Figure 7: Feed-forward dropout sampled during hyperparameter search plotted against the minimal validation loss reported for the respective search trials.

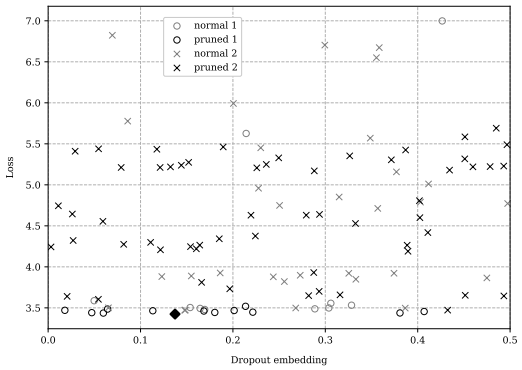


Figure 8: Embedding dropout sampled during hyperparameter search plotted against the minimal validation loss reported for the respective search trials.

the reported losses on the train and validation splits of the WikiText dataset during training can be found in Figure 10.

Both loss and perplexity on the training and validation splits of the WikiText 103 dataset were computed for the trained model (Table 10). Unfortunately, the full training run resulted in slightly higher validation perplexity than the optimal value found during hyperparamter search (30.67 vs 31.45). Reasons could be different random seeds and a rounding of the identified hyperparameter values. Testing is done on a Lenovo ThinkPad P14s Gen 4 with 16 AMD Ryzen 7 PRO 7840U processors, 64 GiB of DDR5 RAM and using an integrated AMD Radeon 780M GPU.[39]

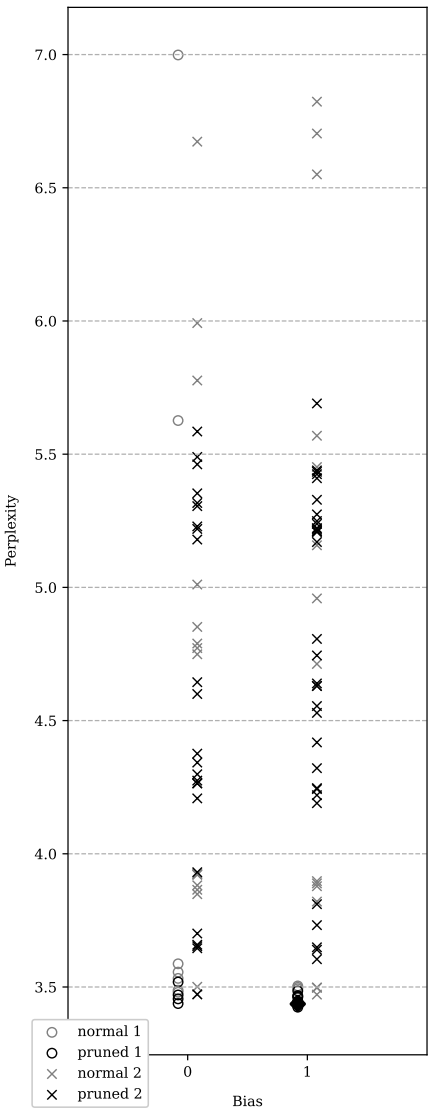


Figure 9: Minimal reported validation loss plotted separately for the trials that include a bias term and for those that omit it.

Additionally, sentence-based perplexity on the dataset was computed for the `GPT-2 medium` model and can also be found in Table 10. Due to the lack of a beginning-of-sentence token, the first word of a sentence is ignored in the measure. Validation split perplexity for `GPT-2 medium` is more than twice as high as for the custom LM considered here (72.96 vs 31.45), appearing to confirm the earlier hypothesis that `GPT-2` is not well adapted to single-sentence contexts (see Section 7.1.1.1). However, note that the perplexity values are not directly comparable since the two models have different vocabulary sizes and that the comparison is not fair since `GPT-2` is evaluated out-of-domain here. Additionally, Radford et al. (2019) found that Wikitext has a training/test set overlap of 9.09 percent in terms of 8-grams due to a duplicated article and reused paragraphs across related entries. They do not include an evaluation with respect to the validation set but it is plausible to assume that there is also some degree of overlap which influences the validation loss of the custom LM positively.

On the other hand, `GPT-2` has seen more training data than the LM trained from scratch (40 GB of text (Radford et al., 2019), i.e. around 7-8 billion words assuming 5-6 bytes/word vs 103 million for WikiText), and it has a significantly smaller vocabulary (50 257 vs 222 319) which typically reduces perplexity. Radford et al. (2019) get a lower perplexity of 26.37 on the WikiText 103 test set with `GPT2-medium`. Note however, that they use a much larger context size of 1024 tokens instead of considering sentences of up to 60 tokens which is known to reduce perplexity strongly. This lets me conclude that the model

[39] This required the use of pytorch for ROCm and passing the environmental variable `HSA_OVERRIDE_GFX_VERSION="11.0.2"` since AMD Radeon 780M cards were not officially supported by ROCm at the time of conducting the evaluation.

Split	Train	Validation
Loss	2.93	3.45
Perplexity	18.70 (70.39)	31.45 (72.96)

Table 10: Benchmarks on the final language model. In parentheses, the corresponding perplexities of `GPT-2 medium` on the sentence-level are provided.

Hyperparameter	Value
Maximum epochs	39*
Batch size	1920 ($160 \cdot 12$ GA†)
Evaluation interval	every 500 batches
Resources	HHU HPC Hilbert
Device	4 Nvidia A100 GPUs
Threads (dataloader workers)	8 per process
Processor	AMD EPYC 75F3, 2.95 GHz
Peak RAM usage	89.98 Gb DDR4
Duration until minimum	10.53 h
Epochs until minimum	8.8

Table 11: Settings for training the final language model; *early termination due to clear overfitting, see Figure 10; †gradient accumulation.

trained here delivers reasonably strong language modelling performance on the validation split. Therefore, it is used for the subsequent calculation of surprisal and attention costs.

7.2.2.1 Psycholinguistic evaluation while training

In addition to monitoring the training process with respect to the loss function on the training batches and the validation split, I also tracked the predictiveness of surprisal for human reading times based on model snapshots. This serves as a proof of concept for experiment 2 which requires quick on-the-fly tests as part of the model training loop.

Fitting linear mixed-effects models via the `lme4 R` package (Bates et al. (2015); Section 6.3.8) was not an option for performance reasons: firstly, it requries writing the measures extracted from the neural language model to the disk and then loading them into `R`. This is computationally costly. Secondly, the `lme4::lmer` function is comparably slow for large datasets and sophisticated random effects structures. Thus, using `lme4` once in each evaluation step, would severely slow down training.

Therefore, I implement a heuristic solution using the faster `gpboost Python` package (Section 6.3.8). At each validation interval, a new LMEM with a pre-determined structure is fitted and its likelihood is recorded. Statistical tests for the effect of surprisal are not conducted in this case since the random effect structure cannot be adjusted appropriately to ensure valid inference during training. Furthermore, the LMEM design chosen is not meant to account for all levels of variance since it is only an intermediate step towards a sufficient modelling approach for reading time prediction as a hyperparameter validation metric.

The model is given in Equation 170. `len`, `freq` and `surp` stand for length, frequency and sur-

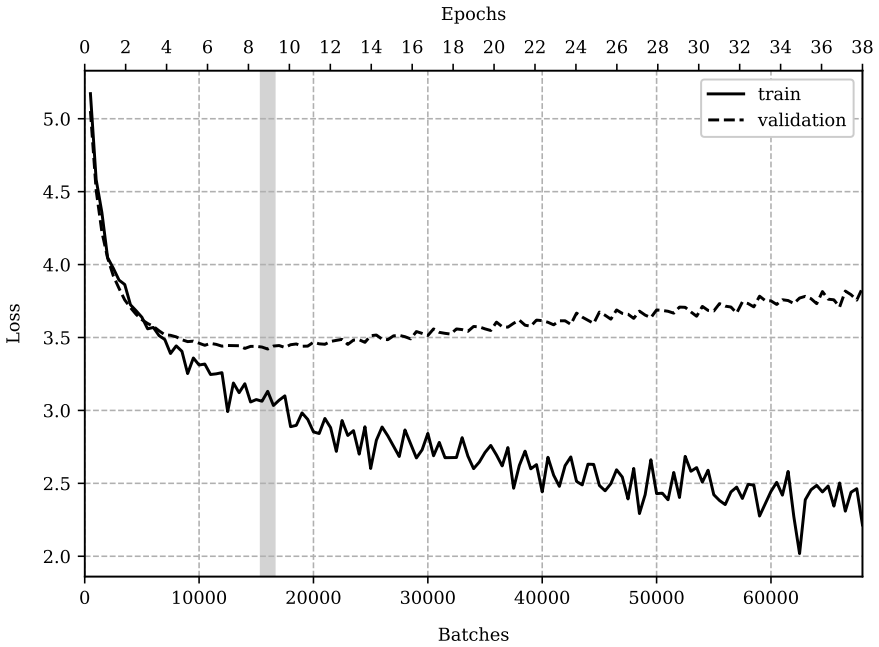


Figure 10: Development of loss during training. The top x-axis marks the number of epochs while the bottom x-axis gives the corresponding number of batches (1920 sentences each) which loss was calculated on. The gray vertical bar marks the best validation score (3.45) reached after training for 8.8 epochs. This determines the model checkpoint used for psycholinguistic evaluation.

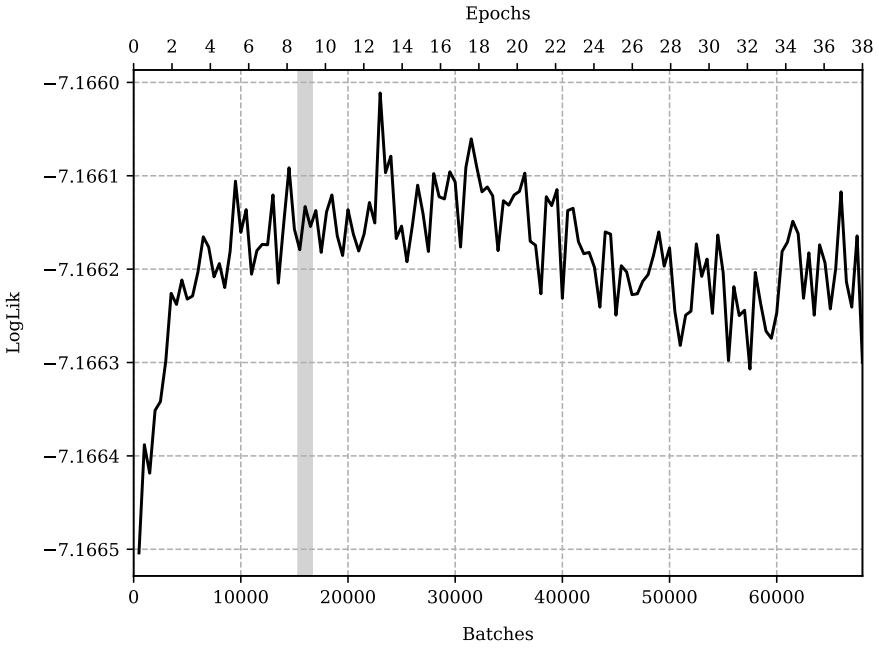


Figure 11: Development of LMEM log likelihood resulting from fitting surprisal generated by model checkpoints to go-past times in Provo. The top x-axis marks the number of epochs while the bottom x-axis gives the corresponding number of batches (1920 sentences each) which loss was calculated on.

prisal respectively. `.1` denotes a variable lagged by one position (spillover). Note that correlations between random coefficients are not supported by `gpboost`.

```
GPT ~ 1 + len + len.1 + freq + freq.1
      + surp + surp.1 + (1 + len + len.1                    (170)
      + freq + freq.1 + surp + surp.1 || subject)
```

Recording the likelihood of LMEMs including surprisal from LM snapshots allows to track the change in goodness of fit caused by different versions of the surprisal predictor. This allows to select a model snapshot with an optimal psychometric predictive power, which will be needed in experiment 2 (Section 8). However, this of course does not entail that the surprisal values from the optimal model present a significant effect or that they express predictive power beyond their alternative forms at different validation steps.

Figure 11, shows the development of LMEM log-likelihood resulting from fitting surprisal generated by model checkpoints to go-past times in Provo. Higher log-likelihoods signify that the model describes greater variability. The line largely appears to mirror the progression of loss on the WikiText validation split during training (Figure 10): log-likelihood is lowest in the beginning, quickly increases and then stagnates after batch 10 000, before starting to decrease again. In contrast to validation loss, stagnation is marked by jagged movements. Furthermore, the optimum is reached a bit later at around epoch 13 instead of 8.8 where the validation loss has already started to increase again. This insinuates that slight overfitting for language modelling might improve psychometric predictive power. However, this is to be interpreted with caution due to the non-maximal inadequate design of the LMEM.

Overall, this test showed that the model selected for evaluating attention patterns does not suffer from any training-based impediments of the psychometric predictive power of its next-token distributions, making it an adequate choice of model for generating the surprisal baseline in the next section.

7.2.3 Model identification

In order to perform statistical inference with respect to the candidate attention measures, appropriate linear mixed-effects model designs needed to be identified. In the following, I will report on key results in the identification process for the null-model and on adjustments to the baseline that had to be made for the individual likelihood-ratio tests.

7.2.3.1 Baseline LMEM

This section describes relevant design decisions during the baseline model selection following the process outlined in Section 7.1.3.2. First, multicollinearity for frequency, length, surprisal and the temporal predictors is determined via VIF, finding a maximum value of 2.82 for frequency, which is below what practitioners consider a problematic level of multicollinearity (Section 6.3.6). Therefore, multicollinearity is not assumed to be an issue. Then, a linear model including frequency, length and surprisal as predictors is used to conduct separate LRTs with respect to the temporal predictors (word position inside a sentence, sentence position inside a paragraph and paragraph position inside the corpus), finding that the null-hypothesis cannot be rejected for either of them.

In the next step, random effects are introduced by adding random intercepts and coefficients for the three baseline predictors in the model. First, subject-based random effects are added. In this context, word position inside a sentence is again tested for inclusion but provides only an insignificant contribution and is therefore left out.Then, the two grouping hierarchies are considered, starting at the lowest level and extending iteratively to coarser groupings. Simplifications are necessary due to a maximal structure being unsupported by the data, resulting in the LMEM design given in Equation 171.

```
GPT ~ 1 + len + freq + surp
      + (1 + len + freq + surp || subject)
      + (1 + surp || word)
      + (0 + freq + surp || paragraph)
      + (1 || sentence)
      + (1 + len + freq || abs.pos)
```
(171)

This structure teaches us that the magnitude of the effect of length, frequency and surprisal appears to vary between subjects. Additionally, the effect of surprisal varies by word type, i.e. some words seem to be more susceptible to predictability-based slow-down than others. Lastly, the effect of word length only varies by participant and not on a sentence or paragraph level, while frequency and surprisal effects both exhibit contextual variability. The `lemma` grouping variable was discarded due to singularity issues when including random coefficients and non-significance of its random intercept.

Note that no random intercept is assumed for the paragraph-level. The model estimated a very small bias parameter, suggesting that difference in mean RTs between paragraphs is explainable largely through sentence-wise intercepts. Nonetheless, the intercept is added back to the model after adding spillover effects due to a significant LRT. Secondly, two random coefficients (`len`, `freq`) for `abs.pos` were introduced by mistake and are not justified by the study design: neither length nor frequency vary for repeated observations of a single word instance. The study is not repeated without the coefficients since the models still reach converging estimates and the coefficients can be expected to behave like random intercepts in this group, having no negative effect on the statistical results.

Before exploring the inclusion of spillover effects, a first test of the assumptions which likelihood-ratio testing relies on, is conducted on the model specified in Equation 171. A check for linearity of the predictors reveals a diagonal threshold in the lower left of the residual plot in Figure 12. This is an expected pattern due to reading times being non-negative, possibly exacerbated due to the lower cut-off of 80 ms in the Provo corpus. Furthermore, residuals disperse more to the top than to the bottom, indicating that among high reading times underpredictions of GPT are more common than overpredictions.

These observations suggest that another kind of modelling that does not predict values smaller than the lower cut-off point (or even negative values) might be more appropriate. One way to achieve this would be to log-transform the dependent variable. However, this would change the interpretation of the model predictors, making them multiplicative. Additionally, it contradicts prior findings about the linear effect of word length, log-frequency and surprisal. In order to retain interpretability, the dependent is thus not scaled.

The normality of residuals seems to be suboptimal: the Q-Q plot in Figure 13 shows a dent, i.e. residuals are right-skewed, also suggesting log-transformation. Nevertheless, the large number of

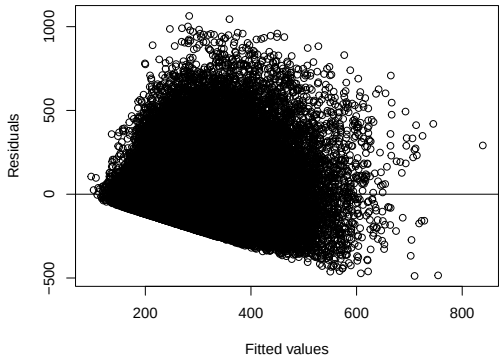


Figure 12: Residual plot: predicted go-past times plotted against residual errors for the Provo corpus for the model specified in Equation 171.

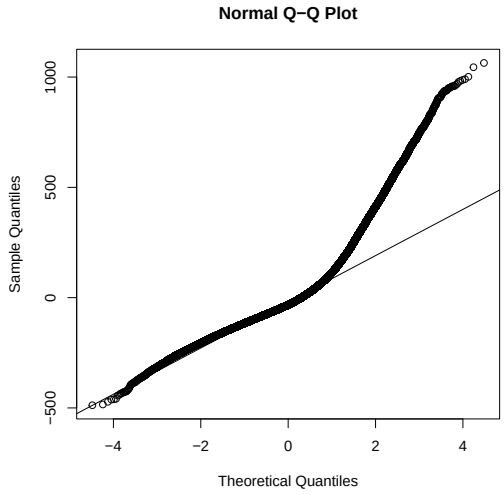


Figure 13: Q-Q plot for the model specified in Equation 171.

observations suggests that this should not be an issue for the validity of inference.

The check for homoscedasticity mostly did not give reasons for concern. At a first glance, variance appears to be lower for high surprisal/low frequency/high length words, however, the data is also more sparse in these areas, possibly explaining the occurrence of fewer outliers. A peculiar observation concerns word frequency (Figure 14): Two sets of extremely low frequencies (words occurring less than 100 times per 1 billion words) are positioned far from the rest of the values. Manual inspection shows that this pertains to 550 observations, based on words partly including special characters. Possibly `wordfreq` was not able to estimate a sensible frequency in these cases. Therefore, I remove these instances from the analysis.

Finally, a check for the normality of the random effects reveals only slight violations of normality, less severe than the right-skew found for the normality of residuals.

The spillover inclusion procedure found an ideal window of 2. However, not all predictors exhibited significant spillover effects: lagged frequency did not improve model fit significantly while length only did so for a window of one. For surprisal, a lag of two is supported. The final baseline model is given in Equation 172.

```
GPT ~ 1 + len + len.1 + freq + supr + surp.1 + surp.2
      + (1 + len + len.1 + freq + surp | subject)
      + (1 | word)
      + (1 + surp + surp.1 || paragraph)
      + (1 | sentence)
      + (1 + len.1 + freq || abs.pos)
```
(172)

The spillover inclusion procedure required the removal of certain random effects due to convergence issues, e.g. frequency for the paragraph grouping. Furthermore, the inclusion of random coefficients for spillover predictors was supported by the data only to a limited extent, namely for length on the subject-level and surprisal within paragraphs.

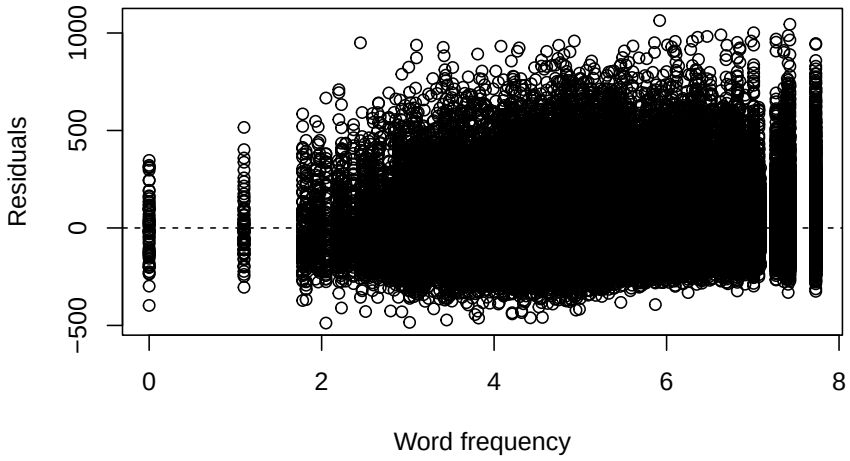


Figure 14: Q-Q plot for the model specified in Equation 171.

After finding the optimal spillover level, the reintroduction of the paragraph-wise intercept and correlations between random effects were successfully tested. The resulting model was fit on 9240 fewer observations than the the model without lagged predictors due to the removal of the first two words of each paragraph for which no lagged independent variables are available.

An additional check of diagnostics was performed, finding no substantive changes.Table 12 shows the effect estimates of the baseline model and Table 13 gives the random effects variances. There are 2413 groups for abs.pos, 1093 word groups, 133 sentence groups, 84 groups for `subject` and 55 distinct paragraphs. Correlations for the fixed effects are given in Table 17.

7.2.3.2 Hypothesis models

For each of the 14 attention costs (*candidates*) to test, a null hypothesis model and an alternative model are fitted based on the baseline identified in the preceding step. This involves adding random coefficients for the respective candidate to all groupings included in the model design for both models and adding a fixed coefficient for the candidate in the alternative model. Since doing this again leads to issues in model estimation, additional simplifications are performed. This results in differing pairs of models for the 14 predictors given in Table 14.

For each predictor, multicollinearity with the baseline predictors was checked, raising no issues. Additionally, manual inspection of the predictors via histograms revealed an accumulation of extreme values for the first word in each sentence. For instance, there is a spike at zero attention entropy (see Figure 15) since a context size of 1 (only the current word) produces an entropy of 0. However, due to the risk of the modelling decision for this special case influencing the predictiveness of the predictor as a whole, the first word in each sentence is subsequently omitted from the analysis.

Including costs measures required additional simplifications of the random effects structure. In

	Estimate	SE
intercept	282.01	5.36
frequency	-7.77	1.95
length	23.11	2.12
surprisal	8.07	1.21
length.1	-7.30	1.23
surprisal.1	4.65	1.22
surprisal.2	2.34	0.85

Table 12: Baseline LMEM effect estimates and standard errors.

Group	Predictor	Variance	Correlations		
abs.pos	intercept	788.21			
	frequency	257.46			
	length.1	66.32			
word	intercept	166.86			
sentence	intercept	61.13			
subject	intercept	2235.16			
	frequency	40.70	-0.45		
	length	188.30	0.63	-0.35	
	length.1	31.38	-0.31	0.12	-0.28
paragraph	intercept	29.95			
	surprisal	9.80			
	surprisal.1	23.39			
residual		21505.84			

Table 13: Baseline LMEM random effect variances and correlations.

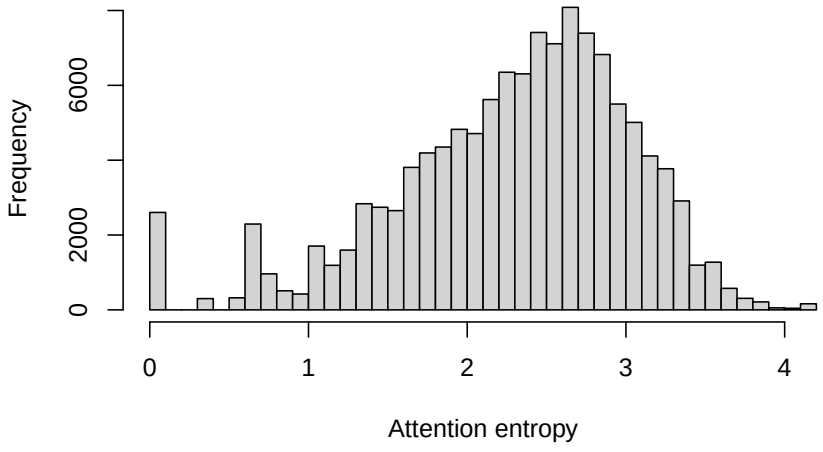


Figure 15: Histogram of local unnormalised attention entropy before removing the observations for the first word in each sentence.

Candidates	Formula
$\mathrm{entr}^{\mathrm{N}}_{\mathrm{loc}}, \mathrm{dist}^{\mathrm{N}}_{\mathrm{loc}}, \mathrm{act}^{\mathrm{N}}_{\mathrm{loc}}, \mathrm{diff}^{\mathrm{N}}_{\mathrm{loc}}, \mathrm{act}^{\mathrm{N}}_{\mathrm{glob}}$	`GPT ~ 1 + freq + len + surp + len.1 + surp.1 + surp.2 + **cand** + (1 + freq + len + len.1 + surp + **cand** || subject) + (1 + **cand** || word) + (1 + surp + surp.1 || paragraph) + (1 + **cand** || sentence) + (1 + freq + len.1 + **cand** || abs.pos)`
$\mathrm{entr}^{\mathrm{N}}_{\mathrm{glob}}, \widetilde{\mathrm{act}}^{\mathrm{N}}_{\mathrm{loc}}, \widetilde{\mathrm{act}}^{\mathrm{N}}_{\mathrm{glob}}$	`-(**cand** | abs.pos)`
$\widetilde{\mathrm{entr}}^{\mathrm{N}}_{\mathrm{loc}}$	`-(**cand** | word)`
$\widetilde{\mathrm{dist}}^{\mathrm{N}}_{\mathrm{loc}}, \widetilde{\mathrm{entr}}^{\mathrm{N}}_{\mathrm{glob}}$	`-(**cand** | word) - (**cand** | sentence)`
$\widetilde{\mathrm{diff}}^{\mathrm{N}}_{\mathrm{loc}}$	`-(**cand** | word) - (surp.1 | paragraph) - (**cand** | abs.pos)`
$\widetilde{\mathrm{diff}}^{\mathrm{N}}_{\mathrm{glob}}$	`-(**cand** | word) - (**cand** | abs.pos)`

Table 14: Identified LMEMs for testing the attention-based measures. The second column supplies the LME model descriptions used for individually testing the candidate metrics given in the first column. The term **cand** stands for the respective attention measure given in `lme4` syntax. Rows 2–6 contain the components removed from the maximal model fitted for LRT given in row 1. The notation is explicit – intercepts were not removed. `freq` = frequency, `len` = length, `surp` = surprisal. "`.1`" and "`.2`" signify lagged versions of predictors. "`||`" means that correlations between random effects are not accounted for.

all cases, random correlations were unsupported by the data. Furthermore, for some candidates, specific random coefficients had to be omitted: Most unscaled predictors supported the maximal random effects structure (except for $\mathrm{entr}^{\mathrm{N}}_{\mathrm{glob}}$) while all scaled measures needed simplification. Most often, this pertained to the word form grouping, i.e. where variations of the effect of attention costs depending on the word were unsupported, at least by the data at hand.On the other side of the spectrum, candidate coefficients were not removed in any case for the subject grouping. Under an interpretation of the attention measures as reflections of memory costs, this observation would be consistent with subjects showing different susceptibilities to phenomena such as long-range dependency integration, but the memory cost not interacting with the kind of lexical item stored or retrieved.

7.2.4 Statistical testing

Table 15 summarises fixed-effect estimates (β; per one standard deviation (SD) increase due to z-scaling), χ^2 statistics, raw p-values, BH-corrected p-values and information-criterion differences (Δ_{AIC}, Δ_{BIC}) for attention-based predictors of go-past time relative to their respective baseline models.

Across specifications, three patterns emerge: (i) head-averaged (local), unnormalised metrics consistently improve fit beyond surprisal and lexical controls, (ii) globally aggregated metrics show weaker and less reliable improvements and (iii) context-length scaling largely removes (and sometimes reverses) effects. These results will be explained in more detail below.

7.2.4.1 Head-averaged (local) metrics

All four unnormalised head-averaged metrics – entropy, distance, activation and difference – significantly improved fit relative to the baseline (corrected $p < 0.05$). Fixed effects were positive and similar in magnitude across predictors ($\beta \approx 3\,\mathrm{ms}$ per SD), indicating that higher attention-derived memory cost is associated with longer go-past times.

Improvements in Δ_{AIC} were modest but consistent across predictors, suggesting small but systematic gains in explanatory power beyond surprisal, word length, frequency and their spillover terms. The estimated effects align in direction with theoretical predictions, meaning positive relationships between each local metric and GPT. Figure 16 shows positive relationships between each local metric and GPT with binned measurements resembling mostly linear growth. Effect sizes are less than half the estimated effect of surprisal (see Table 12). Notably, this contrasts with findings on self-paced reading where attention entropy can exceed surprisal in predictive strength (Oh and Schuler, 2022).

Table 17 provides correlation coefficients for the baseline predictors and the attention costs, including correlation between different attention costs within category (local/global, scaled/unscaled). The attention measures – not included together in a joint LMEM – are strongly correlated, with the highest correlations of 0.99 for the local unscaled costs occurring between attention distance and activation and between difference and activation. This suggests that the measures capture closely related aspects of attention structure, making them difficult to tease apart.

7.2.4.2 Globally aggregated metrics

When attention measures were computed from the globally averaged attention distribution, results were less consistent. Activation and attention difference remained significant after correction (corrected $p < 0.05$), with positive coefficients comparable to the local variants, whereas global entropy did not reach significance and showed a smaller effect size than local entropy (2.29 vs 2.97).

Inspection of the distributions suggests a mechanistic reason for the weaker global results: after averaging heads, the attention distribution becomes highly diffuse, pushing entropy toward its maximum, thereby reducing variability. Figure 17 illustrates this pattern for a long sentence: local metrics vary more substantially, while global entropy closely tracks its maximum.

These observations indicate that averaging attention before metric computation can wash out head-specific structure that is informative for GPT. If attention is interpreted as a retrieval-like mechanism, the contrast between local and global variants suggests that a single pooled "retrieval process" is an oversimplification of how transformers allocate attention. Additionally, the global attention distribution $\mathrm{Attn}^{\mathrm{N}}_{\mathrm{glob}}$ may be influenced by noisy uninformative attention heads whose contribution to the model is actually low following their output weight $\mathbf{W}^{\mathrm{O}}$. However, the global measure takes all attention heads into account equally, without weighing them with respect to their overall magnitude. This might manifest in semi-static noise when computing attention measures on a per-head basis in the local setting (possibly being a contributor to the linear trend observed in Figure 17), but degrade the global distribution.

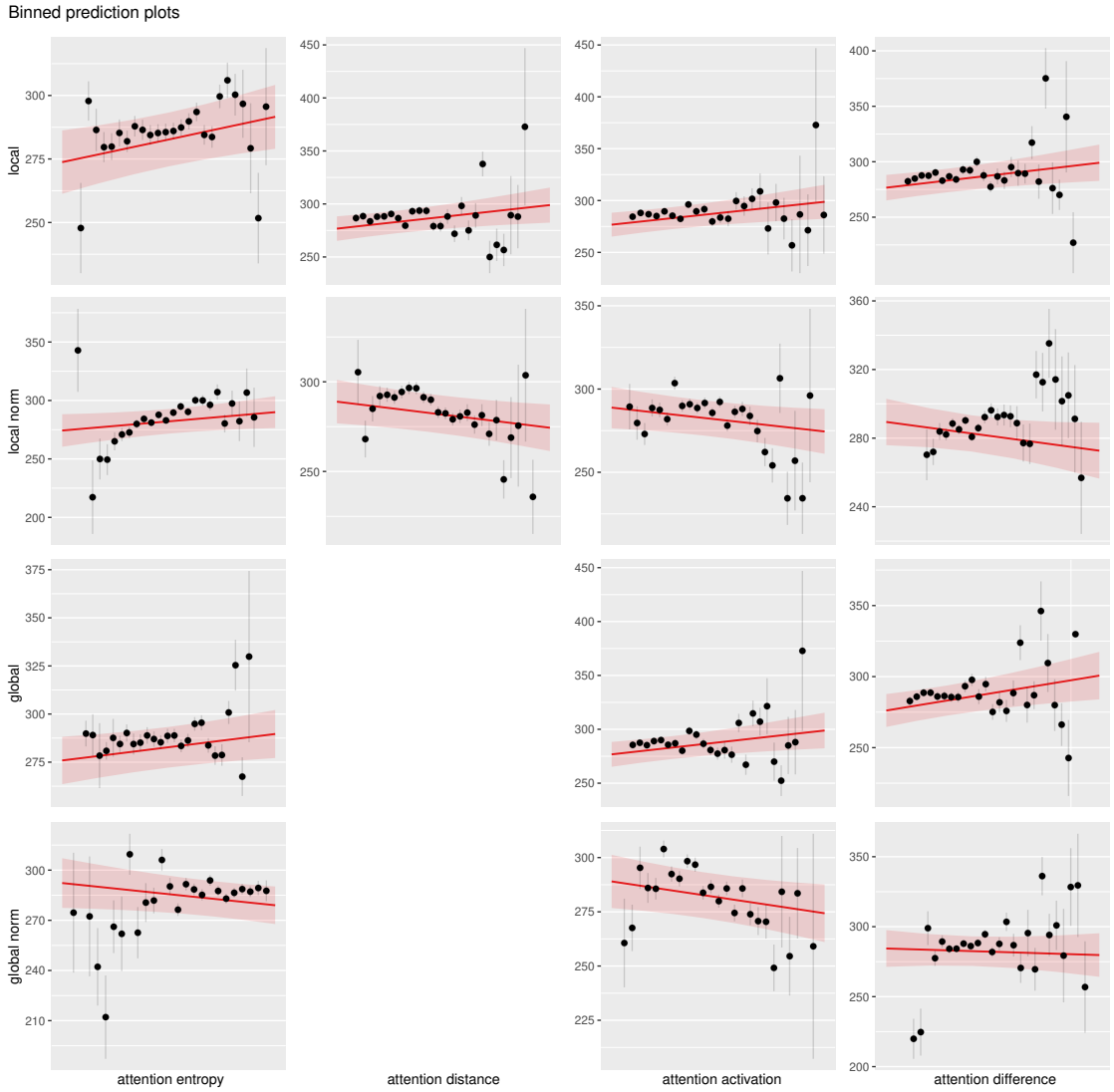


Figure 16: Population-level predicted go-past times as a function of the attention measures (x-axis), based on fixed effects from the mixed-effects models, with other predictors held constant at average values. Points show mean observed go-past times within evenly spaced bins of the predictor. Shaded areas and error bars indicate 95% confidence intervals.

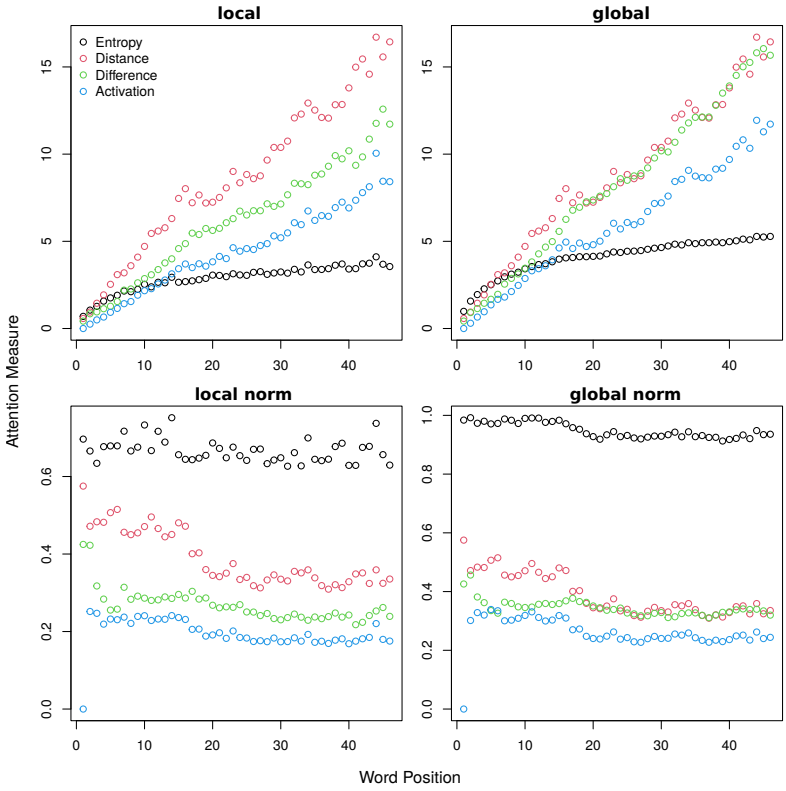


Figure 17: Attention measures extracted for the example sentence *"When a nucleus has all of its proton shells or neutron shells loaded to full capacity, the layers can align so well that each shell can spoon intimately with its attractive neighbors, forming a more compact sphere that fits well within the strong nuclear force's area of influence.", taken from Provo.*

Metric	Variant	Scaling	Effect	SE	χ^2	p	BHp	Δ_{AIC}	Δ_{BIC}
entropy	local	no	2.97	1.14	6.51	0.011	0.033	-5	5
		yes	1.96	1.13	2.97	0.085	0.099	-1	9
	global	no	2.29	1.13	4.18	0.041	0.072	-2	7
		yes	-1.90	1.04	3.35	0.073	0.093	-1	9
distance		no	3.17	1.26	6.07	0.014	0.033	-4	5
		yes	1.96	0.98	4.39	0.036	0.072	-2	8
activation	local	no	3.15	1.25	6.13	0.013	0.033	-5	5
		yes	-2.04	1.08	3.45	0.063	0.088	-2	9
	global	no	3.16	1.26	6.04	0.014	0.033	-4	6
		yes	-2.09	1.04	3.97	0.046	0.072	-2	8
difference	local	no	3.21	1.24	6.43	0.011	0.033	-4	5
		yes	-1.67	1.06	2.38	0.123	0.133	0	9
	global	no	3.51	1.28	7.20	0.007	0.033	-5	5
		yes	-0.47	0.98	0.00	0.935	0.935	2	11

Table 15: Effects of attention-based predictors on go-past times in LMEMs. For each metric, I report the fixed-effect estimate (β; per one standard deviation increase), standard error (SE), likelihood-ratio χ^2, raw p-value, BH-corrected p-value (BHp), and information-criterion differences relative to the corresponding baseline model (ΔAIC, ΔBIC). Highlighted BHp-values indicate effects surviving multiple-testing correction (BHp < 0.05).

7.2.4.3 Context-length scaling

In contrast to the unnormalised predictors, none of the context-length-scaled variants produced significant improvements beyond the baseline after correction. Moreover, several scaled predictors exhibited negative coefficients, indicating shorter predicted go-past times with increasing scaled memory cost. The bottom plots in Figure 17 hint towards a recency bias in attention with the measures declining throughout the sentence, i.e. sublinear growth with word position for distance, activation and difference. The absence of positive scaled effects suggests that go-past times align more with absolute growth in contextual demands than with proportional attention allocation within an expanding window as suggested by Oh and Schuler (2022).

7.2.4.4 Magnitude of improvement

Across significant models, Δ_{AIC} values (Table 15) indicate modest but consistent improvements over the baseline. This suggests that the additional attention-based predictors yield small likelihood gains relative to a model containing only expectation-based and lexical predictors. In contrast, all Δ_{BIC} values are positive, indicating that BIC favours the corresponding baseline models. This discrepancy is expected: whereas AIC penalises an additional parameter by a constant factor, BIC uses a penalty that increases with the number of observations (Wasserman, 2004, Section 13.6). Given the large number of observations in the eye-tracking data, the likelihood gains associated with the attention-based predictors are therefore not large enough to overcome the stronger BIC penalty.

Taken together, the model-comparison criteria support a cautious interpretation. The likelihood-ratio tests and Δ_{AIC} values provide evidence that several attention-based metrics explain addi-

tional variance in go-past times beyond surprisal and lexical predictors. However, the positive Δ_{BIC} values indicate that these gains are small relative to the increase in model complexity under a more conservative criterion. Thus, attention-based metrics appear to capture statistically detectable but weak additional variation in reading times, while surprisal remains the dominant predictor.

7.2.4.5 Example sentence analysis

I take a look at the attention measures generated by the models for a long example sentence in Figure 17. For both the local and global variant, the structural metrics generally increase with sentence position. For entropy, this increase resembles a logarithmic relationship while it appears linear for the other metrics. The transformer seems to spread its attention across the whole available context at each position, leading to increasing measures.

However, the relationship is not smooth: For instance, distance is decreasing locally between positions 15 – 20 and 30 – 40 which coincides with the beginnings of the main phrase and a relative phrase. Additionally, the increasing attention difference measure indicates that the heads do not simply specialise to attend to specific absolute word positions in the sentence, since maintaining attention at the same word would result in low costs. However, difference and activation are lower in the local setting, indicating some sort of specialisation.

Global entropy is an exception to this observation, in that it follows the maximum possible entropy for the respective word position closely, as is visible in the normalised global setting. This means that the global attention distribution is very diffuse and it explains why this measure provided only a poor fit in the LMEM: across all heads, the transformer model appears to attend to all previous context to roughly the same degree while entropy for individual heads is lower and varies more. It also suggests that it is the functional specification of individual heads which focus on specific item that is influenced by interference and has an effect on RTs.

For all normalised settings, division by $i - 1$ or $\log i$ effectively removes the dominant growth of the metric with sentence position. The negative effect that remains primarily reflects local recency biases in attention allocation. Thus, scaling appears to alter the interpretation of the metrics from cumulative memory load to relative allocation patterns within a growing context. The absence of positive effects in the scaled variants suggests that the predictive contribution of the raw metrics is closely tied to their sensitivity to expanding context.

This seems to contradict the positive observations of Oh and Schuler (2022) for strict left-context entropy. However, note that their report is only based on self-paced reading times due to convergence issues on the eye-tracking data.

7.2.5 Robustness to sentence position

Because the unnormalised attention metrics increase with sentence position, I tested whether their improvements reflect shared variance with word position rather than an independent contribution. The results are reported in Table 16.

Adding position and the logarithm of position individually to the maximal baseline model (first row in Table 10) yielded significant positive effects (linear effect: 3.35, $p = 0.011$; log effect: 2.51, $p = 0.015$), confirming a general increase in GPT across sentences not captured by surprisal,

Metric	Baseline	p	BHp
entropy	position	0.898	0.898
	log(position)	0.029	0.208
distance	position	0.424	0.484
	log(position)	0.086	0.208
activation	position	0.224	0.358
	log(position)	0.104	0.208
difference	position	0.332	0.443
	log(position)	0.086	0.208

Table 16: p-values and BH-corrected p-values (BHp) for the effect of local unnormalised predictors controlled for word position.

length or frequency. This differs from the negative effect of sentence position found by Demberg and Keller (2008), i.e. reading times getting faster as the sentence progresses. It is possible that this trend is rooted in word predictability effects, where a larger context increases certainty about the continuation. If that is the case, the neural language model in this work, likely delivering better performance than Demberg and Keller (2008)'s parser-based surprisal measure, might have absorbed the negative position effect

Now, controlling for position, the improvements of the attention-based metrics were substantially reduced: only entropy showed marginal significance in raw p-values for log(position) resulting in a flipped position effect of -6.85 and an entropy effect of 9.64. However, no effects survived multiple-testing correction.

This suggests that part of the predictive signal in the unnormalised metrics overlaps with generic context growth. Note that this does not imply that attention metrics are spurious. Increased memory costs caused by a larger context might be the cause of the position effect in the first place.

7.3 Discussion

In summary, attention-derived structural metrics over all transformer heads provide modest but systematic improvements in predicting reading times beyond surprisal, providing support to hypotheses [H1a] and [H1b]. However, their explanatory contribution depends critically on aggregation choices and overlaps substantially with sentence position. Additionally, this experiment has shown that the three proposed alternative attention-based costs (distance, activation and difference) provide similar contributions as attention entropy. This is likely connected to their strong correlations, with all four cost types being interpretable as explicit or implicit measures of locality. This makes them particularly hard to tease apart and through the means of this analysis, it cannot be determined which of the underlying mechanisms provides the most suitable explanation for human sentence processing.

These results support a cautious view: transformer attention patterns exhibit properties consistent with memory-related processing difficulty, but they should not be equated directly with human retrieval mechanisms. These findings do not warrant the strong claim that attention is cognitive retrieval. A more conservative linking hypothesis is supported: certain attention-derived structural properties correlate with go-past times in a direction consistent with memory-based accounts, but

frequency	-0.16									
length	0.18	-0.76								
surprisal	0.13	-0.61	0.49							
length.1	-0.03	0.07	-0.03	-0.07						
surprisal.1	-0.00	0.06	-0.07	0.00	0.55					
surprisal.2	0.01	0.04	-0.04	0.05	-0.07	-0.00				
$\text{entr}^{\text{N}}_{\text{loc}}$	0.02	-0.06	0.09	0.03	0.00	-0.07	-0.08			
$\text{dist}^{\text{N}}_{\text{loc}}$	0.00	0.03	0.02	-0.07	0.03	-0.05	-0.06	0.93		
$\text{act}^{\text{N}}_{\text{loc}}$	0.01	0.00	0.04	-0.03	0.04	-0.04	-0.06	0.94	0.99	
$\text{diff}^{\text{N}}_{\text{loc}}$	0.01	-0.02	0.06	-0.01	0.05	-0.02	-0.06	0.93	0.98	0.99
								$\text{entr}^{\text{N}}_{\text{loc}}$	$\text{dist}^{\text{N}}_{\text{loc}}$	$\text{act}^{\text{N}}_{\text{loc}}$
$\widetilde{\text{entr}}^{\text{N}}_{\text{loc}}$	0.05	-0.24	0.13	0.27	-0.16	-0.13	-0.07			
$\widetilde{\text{dist}}^{\text{N}}_{\text{loc}}$	-0.04	0.16	-0.15	-0.13	-0.08	-0.06	-0.01	0.31		
$\widetilde{\text{act}}^{\text{N}}_{\text{loc}}$	-0.02	0.04	-0.08	0.00	-0.02	-0.02	-0.00	0.51	0.86	
$\widetilde{\text{diff}}^{\text{N}}_{\text{loc}}$	0.02	-0.19	0.09	0.22	0.02	0.01	-0.03	0.39	0.51	0.68
								$\widetilde{\text{entr}}^{\text{N}}_{\text{loc}}$	$\widetilde{\text{dist}}^{\text{N}}_{\text{loc}}$	$\widetilde{\text{act}}^{\text{N}}_{\text{loc}}$
$\text{entr}^{\text{N}}_{\text{glob}}$	0.01	-0.01	0.06	-0.04	0.03	-0.05	-0.07			
$\text{dist}^{\text{N}}_{\text{loc}}$								0.95		
$\text{act}^{\text{N}}_{\text{glob}}$	0.00	0.03	0.03	-0.06	0.04	-0.04	-0.06	0.95	1.00	
$\text{diff}^{\text{N}}_{\text{glob}}$	0.01	-0.00	0.05	-0.03	0.04	-0.02	-0.05	0.93	0.99	0.99
								$\text{entr}^{\text{N}}_{\text{glob}}$	$\text{dist}^{\text{N}}_{\text{loc}}$	$\text{act}^{\text{N}}_{\text{glob}}$
$\widetilde{\text{entr}}^{\text{N}}_{\text{glob}}$	0.00	-0.08	0.03	0.02	-0.11	-0.12	-0.06			
$\widetilde{\text{dist}}^{\text{N}}_{\text{loc}}$								0.82		
$\widetilde{\text{act}}^{\text{N}}_{\text{glob}}$	-0.04	0.17	-0.16	-0.15	-0.06	-0.04	0.01	0.86	0.97	
$\widetilde{\text{diff}}^{\text{N}}_{\text{glob}}$	0.02	-0.13	0.06	0.16	0.00	0.01	-0.06	0.50	0.51	0.42
	GPT	frequency	length	surprisal	length.1	surprisal.1	surprisal.2	$\widetilde{\text{entr}}^{\text{N}}_{\text{glob}}$	$\widetilde{\text{dist}}^{\text{N}}_{\text{loc}}$	$\widetilde{\text{act}}^{\text{N}}_{\text{glob}}$

Table 17: Pearson correlations between baseline predictors and attention metrics computed for the observations included in the LMEM analysis. The attention distance-based predictors are listed twice since their global and local form are equal.

the behaviourally relevant signal appears concentrated in specific components as evidenced by the success of Ryu and Lewis (2021) and Ryu and Lewis (2022) in using only specific heads, rather than the fully pooled (global) distribution. Architecturally, this suggests that some retrieval-like operations contribute to internal computation without behavioural consequences. It remains a challenge to unify the cognitively plausible view of a single retrieval mechanism (cf. Timkey and Linzen, 2023) with the multiple parallel and hierarchical attention operations in transformers if psycholinguistic hypotheses are to be explored using these models. A unified theory of transformer-based sentence processing, as proposed by Ryu and Lewis (2025), needs to account for the cognitive feasibility of multiple parallel attention operations.

Moreover, additional analyses are necessary to disentangle the contribution of sentence position to reading times from potential memory effects. As the estimated attention effects are already very small and given that the proposed measures seem to naturally correlate strongly with sentence position, reliably detecting a memory effect beyond position might be difficult to achieve with corpus-based methods and require significantly larger amounts of data.

On the other hand, the choice of eye-tracking metric might have contributed to this correlation. GPT includes the duration of regressions to preceding words. If regressions are used to re-read words that have been imperfectly detected or forgotten (see Booth and Weger, 2013), a large context would imply the existence of more items of this kind and thus lead to more regressions, increasing the go-past time. Against this backdrop, reconducting the analysis for lower-level features such as first fixation duration and gaze duration might be advisable. Alternatively, moving-window self-paced reading times would provide a measure that is not confounded by regressive eye movements.

Secondly, the aggregation choice should be revisited: rather than uniformly averaging heads, weighting heads by their functional contribution (e.g. via output projections) may better preserve behaviourally relevant structure. Third, to reduce reliance on post-hoc linking assumptions and possibly to disentangle position effects and various candidates for memory costs, a possible path forward is to incorporate memory-relevant constraints into modelling itself – either by developing process-level models that predict RT as a function of model-internal computation, by developing dynamic models of memory access as discussed by Ryu and Lewis (2025) or by training objectives that penalise costly attention patterns, as explored in the second experiment (Section 8).

Finally, caution is advised when generalising the results to the broad class of transformer-based LMs. As dicussed in Section 7.1, previous work relating attention patterns with reading times has predominantly focused on the `GPT-2 small` variant. However, given that the predictive power of next-word surprisal decreases for larger transformers (Section 5.4.1.1), this raises the question of how model size and architecture influence the predictiveness of other metrics, such as attention entropy. Although I report modest effects beyond surprisal for a larger model than those used before (Ryu and Lewis, 2025; Oh and Schuler, 2022), the findings do not support strong generalisations. The differences in results might also be due to aggregation scheme or other architectural choices such as word-level tokenisation or a focus on intra-sentential contexts.

However, considering the overwhelming reliability of the effects reported in previous research, my findings are consistent with the possibility that the predictiveness of aggregate metrics decreases as the number of attention heads increases, potentially due to a growing proportion of heads whose computations do not resemble the retrieval operations posited by psycholinguistic theory. Whether these heads play other cognitively interpretable roles or capture linguistic regularities in alternative

ways remains unclear, as does whether increases in the number of heads arising from additional layers and those due to a greater number of parallel heads have comparable effects.

8 Experiment 2

This section describes the methodological framework used to address the second research question:

Q2 Does integrating the minimisation of attention-based costs, in particular entropy and weighted distance, into a transformer language model's loss function improve the predictive quality of the model for reading times?

This question was motivated by the hypothesis that representations of and interactions between cognitive units are influenced by pressure to reduce memory-based processing costs (Section 1.1)

The section is organised as follows. First, the methodological framework is introduced, including a custom transformer loss function, a novel approach to hyperparameter tuning and the statistical evaluation procedure. The second part presents the experimental results. Finally, the findings are interpreted with respect to the research question and discussed in relation to other findings in the field.

8.1 Methodology

For answering [Q2], I will introduce custom loss terms for the transformer language model architecture based on attention costs tested in experiment 1 (Section 7.1.1.1). The LM is not only trained to minimise next-word cross-entropy (equivalent to minimising surprisal; Section 5.3.3.1) but also structural costs derived from attention patterns. Models of this kind will be called *cost models* (CMs). Here, the focus lies on attention entropy and attention distance as proxies for interference and locality effects. This selection is based on the fact that most previous work has focused on these two dimensions (cf. Section 4), enabling an interpretation of the results in this context. The same cannot be said for attention activation and difference proposed in the previous section. Furthermore, experiment 1 showed that attention distance, activation and difference all tend to scale linearly with context size while the growth is logarithmic for entropy. Thus, I assume that choosing entropy and one of the three linear measures will provide the strongest contrast in the analysis.

The evaluation proceeds in three main steps:

1. For both predictor candidates, separate hyperparameter searches are performed to determine the most adequate modality for incorporating the respective attention cost loss term into the transformer model.

2. Using the hyperparameters found in step 1, for each predictor candidate a separate language model is trained for an extended duration of time.

3. Both models are used to generate the respective predictor on the Provo corpus. It is subsequently compared to its version extracted from a pure language model, with respect to

its psychometric predictive power and controlling for participant-based and stimulus-based variation.

The following sections describe the transformer model estimation process and the procedure for statistial inference.

8.1.1 Language model

The language models trained as part of this experiment follow the same overall design choices as the models use in experiment 1 (word-based tokenisation, sentence-contexts; see Section 7.1.1.1). The key changes pertain to the optimisation goal and the size of the model, which will be discussed below.

8.1.1.1 Loss term

The transformer loss function (Section 5.3.3) is altered to constitute a composition of language modelling loss and attention-based measures. This is structurally similar to forms of multi-task learning in which several loss functions are aggregated (Evaldo Leal et al., 2020). In such frameworks, it is often observed that the loss of one task dominates the training process, leading to detrimental performance at the other objectives (Standley et al., 2020). Therefore, losses are usually weighted and a mutually beneficial ratio between them is determined (Evaldo Leal et al., 2020). Since an external single method of evaluation is available in the work at hand, namely psychometric explanatory power, the scaling of the losses will be determined using a simple hyperparameter search, as opposed to sophisticated learning speed- or uncertainty-based methods discussed by Evaldo Leal et al. (2020).

In the following, surprisal and attention costs will be collectively called *(transformer) costs*. For the sake of notational convenience, I define $\mathrm{surprisal}$ for a transformer model over vocabulary V with parameters θ as a function over tokens $w_i \in V$ and token contexts $w_{<i} \in V^*$ as $\mathrm{surprisal}[\theta](\, w_i, w_{<i}\,) \coloneqq -\log P[\theta](\, w_i \mid w_{<i}\,)$. It is assumed that the transformer prepends a special <bos> (beginning of sentence) token to its input sequence, so that $P[\theta](\, w_1 \mid \varepsilon\,)$ is defined.

Definition 8.1 (Transformer cost operator)**.** Let $\boldsymbol{\kappa}$ be a tuple of parameterised cost functions on a transformer for vocabulary V. The *transformer cost operator for* $\boldsymbol{\kappa}$ is defined for all $n \in \mathbb{N}$ as

$$\mathfrak{O}_{\boldsymbol{\kappa}}[\theta] \colon V^n \to \mathbb{R}^{n \times |\boldsymbol{\kappa}|},$$

$$w_1 \ldots w_n \mapsto \begin{pmatrix} \boldsymbol{\kappa}_{(1)}[\theta](\, w_1, w_{<1}\,) & \cdots & \boldsymbol{\kappa}_{(|\boldsymbol{\kappa}|)}[\theta](\, w_1, w_{<1}\,) \\ \vdots & & \vdots \\ \boldsymbol{\kappa}_{(1)}[\theta](\, w_n, w_{<n}\,) & \cdots & \boldsymbol{\kappa}_{(|\boldsymbol{\kappa}|)}[\theta](\, w_n, w_{<n}\,) \end{pmatrix}$$

Given an input token sequence, the transformer cost operator for $\boldsymbol{\kappa}$ yields the values of all component losses specified by $\boldsymbol{\kappa}$, for each of the positions in the input. The combined loss term used in this experiment is defined as a weighted sum of costs. The following definition allows for a flexible selection of multiple composite costs.

Definition 8.2 (Weighted loss)**.** Let $\boldsymbol{\kappa}$ be a tuple of parameterised cost functions on a transformer for vocabulary V. The *weighted loss for* $\boldsymbol{\kappa}$ is defined as

$$\mathcal{L}_{\boldsymbol{\kappa}}[\theta, \mathbf{v}](\, w_1 \ldots w_n \,) \coloneqq \frac{1}{n} \mathbf{1}^{\mathsf{T}} \mathfrak{D}_{\boldsymbol{\kappa}}[\theta](\, w_1 \ldots w_n \,)\, \mathbf{v}.$$

with $\mathbf{v} \in \Delta^{|\boldsymbol{\kappa}|}$ being a probability simplex of weights.

In the present context, attention costs can be interpreted as a form of regularisation since they are not conditioned on the (self-supervised) prediction of training data. Here, combined losses are explored for $\langle\, \text{surprisal}, \text{entr}^{\text{N}}_{\text{loc}} \,\rangle$ and $\langle\, \text{surprisal}, \text{dist}^{\text{N}}_{\text{loc}} \,\rangle$. The focus lies on local unnormalised costs since they emerged as the most promising variants in experiment 1. Thus, the minimisation of attention entropy and attention distance is part of the neural network objective function and interacts with the language modelling task. The patterns that emerge through this process will be the object of psycholinguistic investigation. The search process can be described as sampling an LM cost weight $\lambda \sim \mathcal{U}(0, 1)$, with the complete weight vector being defined as $\mathbf{v} = \langle\, \lambda, 1 - \lambda \,\rangle$. λ will be called the *LM weight*. Transformer models with an LM weight smaller than one will be called *cost models* (CMs) while models with an LM weight of one will be canonically referred to as LMs.

The theoretical motivation for exploring the reduction of attention costs as part of transformer-based language modelling is the hypothesis that they might represent biases in human sentence processing subject to resource minimisation constraints (Section 1.1). Thus, the magnitude and location of their appearance could be influenced by a processor's ability to reduce them in some contexts and failure to do so in others. The consequences of introducing attention-based regularisation to transformer LMs warrants additional discussion: encouraging the reduction of attention entropy might lead to sparser and more focused attention distributions. This implies changes in query/key representations and therefore in the head-specific similarity space. Less competition may be achieved by dispersing some tokens further away from each other in representational space. Alternatively, experiment 1 showed that some attention heads may not partake in retrieval-like operations but introduce noise and diffuse attention. The entropy reduction objective could have a stronger influence on these heads, inducing more retrieval-like patterns.

Note that the reduction of attention entropy should not be confused with the entropy reduction hypothesis (Section 4.2.2) under which a decrease of uncertainty over syntactic representations of the input induces reading time slow-downs. The entropy reduction hypothesis quantifies changes in entropy during incremental processing while the reduction of entropy in this context pertains to the neural network parameter selection process.

The integration of attention distance as a regularisation term is similar to Clark et al. (2025)'s recency-biased attention. They show that applying a soft distance-graded mask to raw attention scores leads to surprisal estimates with stronger predictive power. The minimisation of attention to items further away in context tested here might produce a comparable effect. The transformer model could learn to focus more strongly on local information. It should be noted that Clark et al. (2025)'s method is inspired by lossy-context surprisal, under which it is assumed that preceding words can be forgotten and that a comprehender fills memory-holes with likely content (Section 4.2.1.8), but is not a literal implementation of it since past context is not forgotten, but less likely to be retrieved. While the approach at hand is likewise not a true model of lossy context, it is compatible with a more variable forgetting schema: the model is discouraged to focus on the retrieval of more distant parts of the context, but this is a soft constraint weighted against language

modelling performance. Thus, the model can still focus on some selection of distant promising items if this aids next-word prediction. It might learn to vary the signal strength of key representations depending on an ad-hoc estimation of the importance of items for further processing.

As attention serves an essential role for next-token prediction by performing contextualisation, it can be expected that the incorporation of attention cost optimisation will influence language modelling performance and the psychometric predictive power of surprisal. I hypothesise that assigning very high weights to attention costs and low weight to language modelling loss generally reduces language modelling performance by making attention less flexible and thus hindering the model to perform necessary contextualisation to produce a good next-token estimate. In particular, if attention heads that do not exhibit retrieval-like patterns in the base LM model do so due to having learned such a behaviour through optimisation for next-token prediction, they likely serve some role in this objective, making it probable that attention entropy reduction will inhibit language modelling capabilities.

The research question therefore can be made more precise by asking: does there exist an equilibrium at some moderate attention cost weight, right before language modelling performance begins to degrade, that allows for appropriate levels of flexibility to yield a performant (but not super-human, see Section 5.4.1) language model and a more predictive attention costs shaped by training dynamics?

8.1.1.2 Finding weights

It is not trivial to decide on the weight vectors $\mathbf{v} \in \Delta^2$ for the cost terms. While the present work explores at most two transformer costs in a joint model, the following notations will be formulated with respect to the general case of n costs.

A naive option would be to determine the loss weights as part of the neural network by including a vector of trainable parameters $\mathbf{p} \in \mathbb{R}^n$ and then to normalise the coefficients, for instance by applying the softmax function, such that $\mathrm{softmax}\,(\mathbf{p}) \in \Delta^n$. However, one would run the risk of collapsing the loss with such a scaling. If one task's loss is easier to minimise, the model might drive its weight to be near 1 and ignore the other costs. If some hard to optimise cost were to be a suitable proxy for a cognitive constraint or bias, this method would not be able to identify this fact.

A better option is to treat the weights as hyperparameters and to select the configuration that results in the transformer model that produces the costs that fit reading time data best. More concretely, a hyperparameter search trial will involve sampling cost weights from the uniform Dirichlet distribution $\mathrm{Dir}(n)$. Then, a transformer with a weighted loss parameterised by these weights is trained. Its validation score is computed by using the transformer to generate the costs on a reading time corpus and determining their psychometric predictive power.

As discussed in Section 7.2.2.1, conducting likelihood ratio tests during neural network training is neither feasible nor desirable. The change in the ability of surprisal and attention costs to linearly explain reading times can be measured by tracking the likelihood of LMEMs that include these measures as predictors alongside the standard baseline and grouped covariates. The weight configuration resulting in the highest LMEM likelihood then marks the best trial.

Let $\mathbf{X}_{\mathrm{base}} \in \mathbb{R}^{n \times p}$ be a LMEM design matrix for p baseline predictors and $\mathbf{y} \in \mathbb{R}^n$ the vector of dependent variables (here: go-past time), following Definition 6.4. The $\mathbf{Z}$ matrix is omitted here

due to it being a recompilation of the fixed effect design matrix. Let $\boldsymbol{\kappa}$ be the tuple of transformer costs and $w_1 \ldots w_n$ the training data. The complete optimisation task is described by the following equation:

$$\begin{aligned} & \arg\max_{\mathbf{v} \in \Delta^{|\boldsymbol{\kappa}|}} \ \mathrm{llh}\left(\mathfrak{D}_{\boldsymbol{\kappa}}[\theta^*_{\mathbf{v}}](\, w_1 \ldots w_n \,)\right), \\ \text{where} \qquad & \theta^*_{\mathbf{v}} = \arg\min_{\theta} \mathcal{L}_{\boldsymbol{\kappa}}[\theta, \mathbf{v}](\, w_1 \ldots w_n \,), \\ & \mathrm{llh}\left(\mathbf{X}_{\mathrm{cost}}\right) = \max_{\boldsymbol{\beta}, \boldsymbol{\theta}_{\mathbf{G}}} \ell(\, \boldsymbol{\beta}, \boldsymbol{\theta}_{\mathbf{G}}, \sigma^2 \mid \mathbf{X}_{\mathrm{base}} \circ_2 \mathbf{X}_{\mathrm{cost}}, \mathbf{y} \,). \end{aligned} \tag{173}$$

The hyperparameter search is again implemented using `optuna` and both the TPE sampler and the Hyperband pruner are used in the same fashion as in experiment 1 (Section 7, Table 7), with the difference that instead of language modelling loss, LMEM likelihood is reported in intervals of 500 training batches. Furthermore, the batch size was halved (960), as was the validation interval (250) and the maximal number of epochs (5), with the goal of pruning unpromising trials faster and speeding up the search process. Early stopping after more than three validations without improvements is also applied, but with respect to the combined loss on the WikiText validation split. This comes with the assumption that the configurations behave similarly with respect to gains in predictive power after starting to overfit on their training data.

The LMEM model design used for hyperparameter search is determined by the simplified baselines for the costs identified in experiment 1 (Table 14), but with unsupported absolute position-wise random coefficients omitted and without correlations between random effects, since they are not supported by `gpboost`. The full design is given in Equation 174 and provides a strongly controlled setting to reliably quantify changes in predictive power.

```
GPT ~ 1 + len + len.1 + freq + supr + surp.1 + surp.2 + cand
      + (1 + len + len.1 + freq + surp + cand || subject)
      + (1 + cand | word)
      + (1 + surp + surp.1 || paragraph)
      + (1 + cand | sentence)
      + (1 | abs.pos)
```

(174)

8.1.1.3 Data

Psycholinguistic validation during hyperparameter tuning and the final significance tests of attention costs will both be performed based on the Provo corpus. This should reduce potential confounders due to variations in domain and recording procedure of the reading time corpora. In order to mitigate the risk of overfitting through the hyperparamter search set by selecting non-generalising cost weights, I partition the Provo corpus into a validation split, used during hyperparameter optimisation, and a test split for LMEM evaluation. There are is a potential caveat with this method: psycholinguistic datasets are usually quite small, so that using the common 20/80 split might make test results unrepresentative. To address this, I randomly assign half of the paragraphs in Provo to either of the subsets. This reduces the number of observations available for the final inferential analysis, which may increase standard errors and reduce statistical power. Under sufficiently respected test assumptions, reducing the sample size does not increase the nominal Type-I error rate but it increases Type-II error (Section 6.1). I therefore treat the loss in statistical

efficiency as justifiable, because the main purpose of the split is to keep hyperparameter selection and final hypothesis testing statistically separated to reduce potential type I error.

For training the language models, I will again use the WikiText dataset (Merity et al., 2016). As the linguistic material in Provo is not included as a training signal, validation is performed out-of-domain. Again, training the language model on the Provo corpus would lead to strong overfitting due to its small size and the size of the WikiText dataset can be expected to allow for reasonable well generalisation to Provo.

Note that while hypothesis [H2] can be interpreted as postulating that attention measures as a minimisation objective can help a model acquire more human-like behaviour, neither the LM training dataset, nor the psycholinguistic evaluation dataset are plausible in terms of human language acquisition. They do not include spoken material and stimuli while hearing but only written text of higher degrees of formality. For this (and other conceptual reasons such as the nature of the training procedure), it is important not to understand this study as an exploration of the similarities and differences between human language acquisition and language model training. The purpose of the weight ratio is only to identify a possible fruitful combination between language modelling loss and attention costs during transformer optimisation, which results in strong RT predictors. Neither do I claim that this ratio has an equivalent in humans nor do I want to claim that this is equivalent to how humans acquire language.

8.1.1.4 Changes in architecture and hyperparameters

Initial trials of this experiment using the transformer architecture identified in experiment 1 revealed that changes in model size were necessary to make it feasible to perform a meaningful number of trials with justifiable computational resources. Therefore, the number of transformer layers is reduced from 24 to 12, the number of parallel heads is lowered from 16 to 12 and the feed-forward network factor f_{FF} is reduced to 4. Thus, the model now corresponds to the architecture of `GPT-2 small`. Additionally, the large word embedding matrix is shrunken by lowering the size of the vocabulary from 222 319 to 150 000. This decreases the learnable parameters of the embedding matrix significantly, from 170 740 992 to 115 200 000, i.e. by 33 %. Despite this strong reduction, the number words in the Provo corpus unknown to the tokeniser does not increase, justifying this design choice.

Additional changes on the implementation-level concern the use of 8 bit optimisation through `bitsandbytes` (Dettmers et al., 2021). The 8-bit Adam optimiser is used alongside `bitsandbytes.nn.StableEmbedding` in place of the regular `torch` Embedding module. Following the authors, 8 bit optimisation can deliver significant improvements in memory usage while retaining performance. Nevertheless, it is a possible limitation of the approach taken here that the task at hand might actually need greater precision to find an optimal solution.

Hyperparameters such as learning rate and regularisation are held at the same values as in experiment 1 (see Section 7). The assumption that these parameters will be optimal when integrating additional loss terms is merely a heuristic that keeps computational resource use at a responsible level for this explorative work.

8.1.1.5 Models for comparison

After finding the best weights for the two sets of costs, I will train three final models with different random seeds for each configuration for a longer duration (referred to as $\mathrm{CM}_i^{\mathrm{entr}}$ and $\mathrm{CM}_i^{\mathrm{dist}}$ for $i \in [3]$) as well as three baseline models involving only on language modelling loss (referred to as LM_i). Multiple models are trained to account for the stochastic nature of the optimisation procedure, thus increasing the reliability and stability of the reported results. Based on the training dynamics observed in experiment 1, a maximum number of 10 epochs is adopted.

8.1.2 Psycholinguistic evaluation

The final evaluation will be performed separately for attention entropy and for attention distance. For each predictor candidate, three paired comparisons between a pure language model and a cost model will be performed. Attention costs are tested for their predictive power with respect to go-past time. Individual result for the three trial-pairs are reported since averaging results could mask strong variations.

The baseline linear mixed-effects model identified in experiment 1 (Equation 172) is used as the basis. There, the surprisal control predictor was generated using the same model from which the attention predictor of interest was extracted. However, following the same strategy here would entail introducing two different baselines – one per transformer variant – making the contributions provided by the attention cost from the respective models incomparable. It is thus necessary to settle on one model for generating surprisal. Additionally, the requirement of nestedness imposed by the likelihood ratio test makes it difficult to compare several alternative predictors.

The solution adopted here is the following pipeline, applied to each pair of LM and CM:

1. Select the version of surprisal (from the LM or the CM) that allows to explain the reading time measurements better.
2. Test the two versions of the candidate predictor separately against the surprisal variable selected in the preceding step included in the null-hypothesis model.
3. Compare the predictive qualities of LM- and CM-extracted attention costs.

The first step does not involve statistical testing, as it is only intended to find the strongest among two baselines. It is performed by selecting the model that yields a higher likelihood. In the second step, for each setting (LM-based attention cost and CM-based attention cost) a likelihood-ratio test against the chosen surprisal baseline is performed using a significance threshold of 0.05. The difference in log-likelihood (Δ_{llh}), Δ_{BIC} and Δ_{AIC} are reported per setting and compared to gain an indication with respect to which variant yields greater improvements in explaining the eye-tracking data. This process is conducted separately for all three trials, possibly allowing to assert a general trend.

Note that this testing schema does not allow to derive statements about whether an attention measure from one model is significantly more predictive than the other above and beyond chance. The reliability of this analysis relies on evidence from multiple training runs.

An alternative approach would be to estimate the significance of the effect of the attention cost produced by one transformer variant against a baseline containing the other. However, manual in-

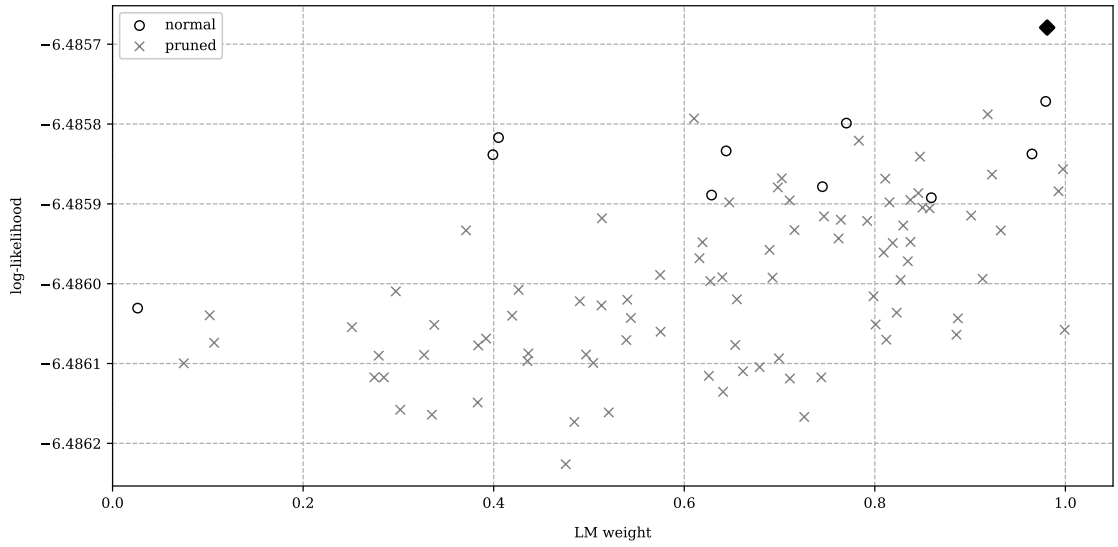


Figure 18: Weights for the language modelling objective sampled during hyperparameter search for $\mathrm{CM}^{\mathrm{entr}}$ plotted against the maximal log-likelihoods reported for the respective search trial. The trial with the highest log-likelihood is marked with a diamond symbol.

spection has shown that the cost variants are strongly correlated (around 0.99 Pearson correlation), possibly making it difficult to estimate a joint LMEM (Section 6.3.6).

8.2 Results

This section presents the results of the second experiment. First, the findings of the searches for suitable cost weights are discussed separately, then benchmarks for the transformer LMs and CMs are provided and the selection of LMEM model designs is reported. Finally, the results of the statistical tests provide insight into the effect of incorporating attention cost minimisation into transformer training.

8.2.1 Weight search

8.2.1.1 Attention entropy

Figure 18 depicts the log-likelihoods reported for the weights of the language modeling loss, subsequently called *LM weights*, sampled during hyperparameter search. It can be seen that the predictive quality of a transformer model, when considering both surprisal and attention entropy, increases with stronger focus on next-token prediction in its training objective. The highest log-likelihood is produced by a model in which attention entropy contributed only to a very small degree to the training loss (LM weight 0.981, attention entropy weight 0.019).

In Figures 19 and 20, the relationships between LM weight/LM loss and LM weight/attention entropy are shown. Changes in training focus influence attention entropy exponentially for LM weights in the interval $[0,0.8]$ and hyper-exponentially in $[0.8,1]$. For LM loss, several layers of

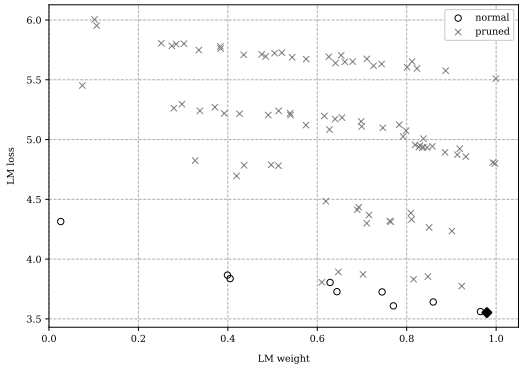


Figure 19: Weights for the language modelling objective sampled during hyperparameter search for $\mathrm{CM}^{\mathrm{entr}}$ plotted against the language modelling loss reported for the respective search trial. The trial with the highest log-likelihood is marked with a diamond symbol.

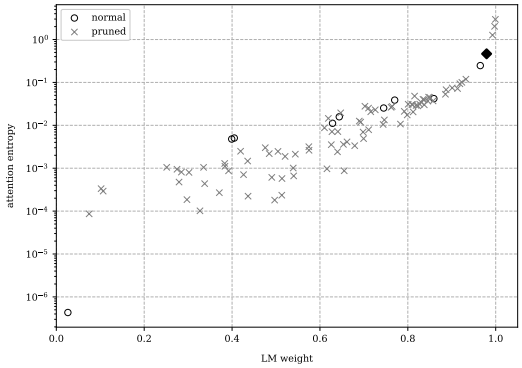


Figure 20: Weights for the language modelling objective sampled during hyperparameter search for $\mathrm{CM}^{\mathrm{entr}}$ plotted against the attention entropy per token reported for the respective search trial; the y-axis is given in log-scale. The trial with the highest log-likelihood is marked with a diamond symbol.

trials seem to manifest at different offsets from the x-axis. As all non-pruned trials are located in the lowest layer, the pattern likely reflects pruning at a few early validation intervals. Across the layers, LM loss decreases linearly with LM weight.

Interestingly, multiple trials with a stronger focus on attention entropy (e.g. two trials with LM weight 0.4) that finished or were aborted due to early stopping reached only slightly lower log-likelihoods than the optimal model while exhibiting noticeably lower attention entropy (< 0.02 vs 0.5). These models appear to have adapted their internals successfully to attention regularisation.

8.2.1.2 Attention distance

The LM weight search results for attention distance are shown in Figures 21–23. The maximum log-likelihood is produced by a model with a high LM weight, albeit lower than in the case of attention entropy (LM weight 0.940, attention distance weight 0.060).

Both LM loss (disregarding pruning-induced layering) and attention distance appear to be monotonic functions of LM weight whereas the log-likehood resulting from combining them in a linear-mixed effects model to predict reading times shows a non-monotonic behaviour (Figure 21). More precisely, while log-likelihood generally increases as a function of LM weight, in the weight interval $[0.4,0.65]$ the search results exhibit a plateau of elevated log-likelihoods. These configurations produce higher LM loss than models approaching the maximum in the interval $(0.65,1]$ (see Figure 22) and would therefore not be compatible with the general linear relationship between language modelling quality and psychometric predictive power of surprisal. This suggests that this pattern is not caused by the surprisal predictor but by the structure of attention distance. If this is indeed the case, there exist weight configurations under which attention distance predicts reading times better than its version emerging purely as a side-product of language modelling.

While the configurations producing the highest log-likelihood focus strongly on language mod-

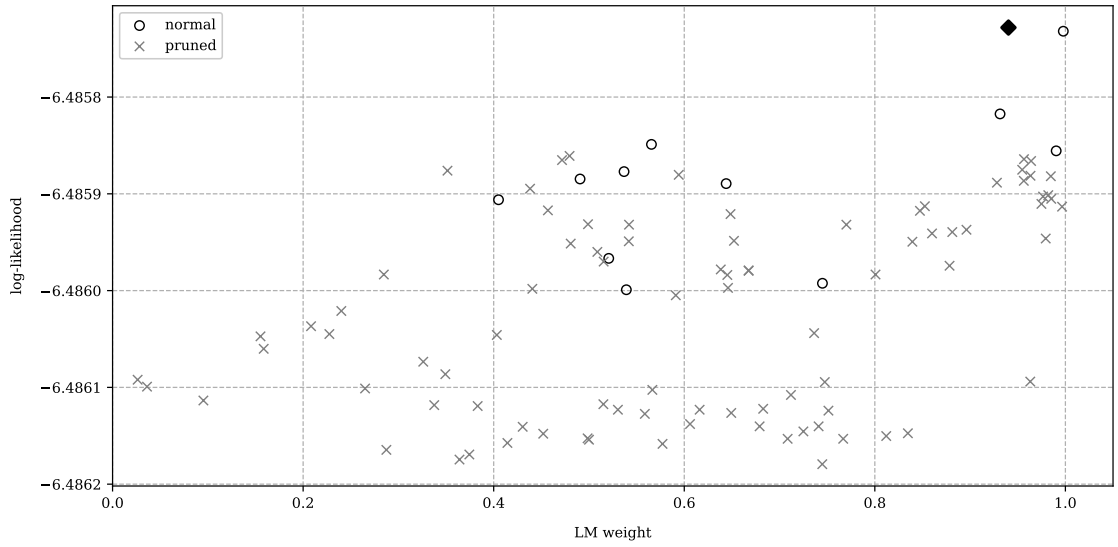


Figure 21: Weights for the language modelling objective sampled during hyperparameter search for $\mathrm{CM}^{\mathrm{dist}}$ plotted against the maximal log-likelihoods reported for the respective search trial. The trial with the highest log-likelihood is marked with a diamond symbol.

elling, it is possible that the ideal weights for attention costs are simply very small. The search space is non-conclusive on this matter: for attention entropy there were three trials with even higher LM weights but they were all pruned by the search algorithm and produced lower log-likelihoods. A more focused weight search for the interval $[0.9,1]$ could be approached in future work.

8.2.2 Model training

The hyperparameter searches for attention entropy and attention distance found that the configurations yielding the highest log-likelihood exhibit high LM weights. However, the searches did not include the case of attention weight being zero. Even small attention weights might have influenced the models' predictive power for reading times, motivating a principled comparison with pure LMs. For both costs, this involves training three CMs with the ideal configuration and three LMs, i.e. models with attention cost weights set to zero.

The models' language modelling capabilities are compared by evaluating them on WikiText (Table 18). On average, the $\mathrm{CM}^{\mathrm{entr}}$ models exhibit a validation perplexity higher by 0.94 than the LMs while the perplexity of the distance cost models $\mathrm{CM}^{\mathrm{dist}}$ deviates slightly stronger, being on average 1.92 lower than the LMs'. The overfitting behaviour is similar across cost specifications. Compared to the larger language model trained in experiment 1 (Table 10), the smaller LMs used here produce a higher perplexity on the training set (on average 21.48 vs 18.70 in experiment 1) but a comparable perplexity on the validation split (on average 31.73 vs 31.45 in experiment 1). Thus, the smaller transformers overfit the training data less while performing similarly, suggesting that the model used in experiment 1 was overparameterised for its optimisation task.[40]

[40]Note that the perplexities are not perfectly comparable due to the change in vocabulary size. A smaller vocabulary

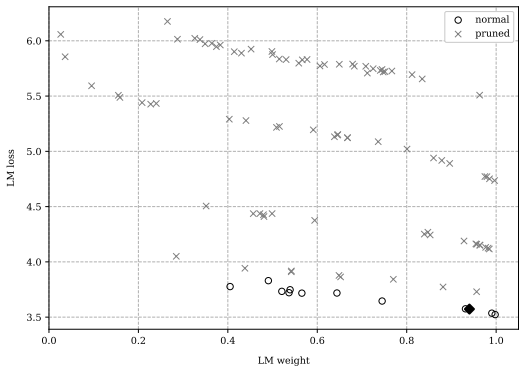


Figure 22: Weights for the language modelling objective sampled during hyperparameter search for $\mathrm{CM}^{\mathrm{dist}}$ plotted against the language modelling loss reported for the respective search trial. The trial with the highest log-likelihood is marked with a diamond symbol.

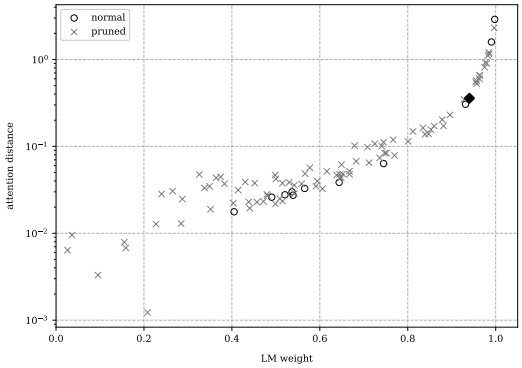


Figure 23: Weights for the language modelling objective sampled during hyperparameter search for $\mathrm{CM}^{\mathrm{dist}}$ plotted against the attention entropy per token reported for the respective search trial; the y-axis is given in log-scale. The trial with the highest log-likelihood is marked with a diamond symbol.

Overall, the integration of attention cost into a model's loss function did not drastically hurt language modelling performance but it did not benefit it either. It can be concluded that transformer models can adapt successfully to weak soft constraints on the dispersion and the temporal range of their attention operations.

8.2.3 Model identification

In the following, the results of the linear mixed-effects model selection process are summarised. Again, as in experiment 1, the simplification strategy outlined in Section 6.4.2 is applied extensively in cases of non-convergence. Full LMEM specifications including details about the removed random coefficients and correlations between random effects are provided to allow for reproduction and contextualisation of the results of the final statistical tests.

Surprisal baseline For each pair of LM and CM, the surprisal baseline is determined by comparing the log-likelihoods of LMEMs including surprisal generated by the LM and the CM respectively. Figure 24 depicts a mixed result: surprisal generated by $\mathrm{CM}_2^{\mathrm{entr}}$ and $\mathrm{CM}_3^{\mathrm{entr}}$ produces a higher likelihood than their LM counterparts (+9 and +5) while for the first run, surprisal gained from the pure language model yields stronger results and is chosen as a baseline (+5). The small difference in likelihood indicates that surprisal is of comparable quality and that variations might be caused simply by randomness in the training process.

The difference for attention entropy are stronger but similarly inconclusive: the log-likelihood of $\mathrm{CM}_1^{\mathrm{dist}}$ is noticeably higher than the log-likelihood of LM_1 (+35). For $\mathrm{CM}_2^{\mathrm{dist}}$ and $\mathrm{CM}_3^{\mathrm{dist}}$ the picture is reversed with likelihood being approximately 13 units lower than for LM_2 and LM_3

size tends to decrease perplexity.

	Run	Train	Val
LM	1	21.76	31.84
	2	21.00	31.68
	3	21.68	31.68
CM^{entr}	1	22.01	32.72
(LM weight 0.981)	2	21.83	32.65
	3	21.93	32.66
CM^{dist}	1	23.59	33.61
(LM weight 0.940)	2	23.57	33.64
	3	23.66	33.71

Table 18: Language modelling perplexities on the WikiText training and validation splits for the three LMs, attention entropy CMs and attention distance CMs. For the cost models, their respective LM weights are given in parentheses.

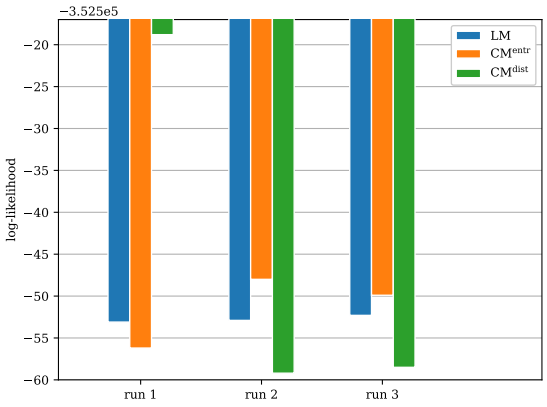


Figure 24: Bar-plot of log-likelihoods of LMEMs including surprisal generated by the LM, CM^{entr} and CM^{dist} models. The individual instantiations are referred to by "run 1", "run 2" and "run 3".

respectively. Again, this does not allow for a general statement about the effect of attention regularisation on the predictiveness of surprisal for RTs.

Against this backdrop, surprisal generated by $\text{LM}_1, \text{CM}_2^{\text{entr}}$ and $\text{CM}_3^{\text{entr}}$ will be used for testing attention entropy and surprisal from $\text{CM}_1^{\text{dist}}, \text{LM}_2$ and LM_3 will be used for testing attention distance run-wise.

Hypothesis models For each of the 12 attention cost candidates to test, a null hypothesis model and an alternative model are fitted based on the design in Equation 174, but with correlations between random effects reintroduced. Since the introduction of the attention measures into the model design leads to issues in model estimation, additional simplifications are adopted. This results in slightly varied models for the 12 predictors which are provided in Table 19. For each covariate, multicollinearity with the baseline predictors is checked, raising no issues.

For all models, the estimation of correlations between random effects in the subject-level grouping was not supported by the data. The other simplifications predominantly pertain to the paragraph grouping where mostly the random coefficients for the candidate predictor and either surprisal or spilled-over surprisal had to be removed. Both the random coefficient for the candidate measure in the subject grouping and in the word grouping were retained by all models, suggesting that participant-induced and stimulus-induced associations can be adequately accounted for.

8.2.4 Statistical testing

Finally, for both the LMs and the CMs, the significance of the effect of attention entropy is estimated using likelihood ratio tests with the null-model including the respective optimal surprisal baseline identified in the preceding step. The resulting statistics can be found in Table 20. The predictor versions will be primarily by means of the change in log-likelihood they induce in the LMEM (Δ_{llh}).

Candidates	Formula
(maximal)	`GPT ~ 1 + freq + len + surp + len.1 + surp.1 + surp.2 + cand + (1 + freq + len + len.1 + surp + cand || subject) + (1 + cand | word) + (1 + surp + surp.1 + cand || paragraph) + (1 + cand | sentence) + (1 | abs.pos)`
a:$\mathrm{CM}_1^{\mathrm{entr}}$, a:$\mathrm{LM}_1$, a:$\mathrm{CM}_2^{\mathrm{entr}}$, d:$\mathrm{LM}_2$	`-(surp + cand || paragraph)`
a:$\mathrm{CM}_3^{\mathrm{entr}}$	`-(surp + surp.1 + cand || paragraph) - (cand | sentence)`
a:LM_2	`-corr:word -(surp + cand || paragraph) -corr:sentence`
a:LM_3	`-(surp.1 + cand || paragraph) - (cand | sentence)`
d:$\mathrm{CM}_1^{\mathrm{dist}}$	`-(cand || paragraph) - (cand | sentence)`
d:$\mathrm{CM}_2^{\mathrm{dist}}$	`-(surp + surp.1 + cand || paragraph) -(cand | sentence)`
d:$\mathrm{CM}_3^{\mathrm{dist}}$	`-corr:word -(supr.1 + cand || paragraph) - (cand | sentence)`
d:LM_1	`-(surp + cand || paragraph) -(cand | sentence)`
d:LM_3	`-(surp + surp.1 + cand || paragraph) -corr:sentence`

Table 19: Simplified LMEMs for testing the attention costs in experiment 2. The null-models omit the fixed effect for the attention cost. The first column specifies the attention cost tested (a: for entropy and d: for distance) and the model it was extracted from. The second column supplies the LME model descriptions in `lme4` syntax, used for the respective tests. The term **`cand`** stands for the respective candidate metric. Rows 2–10 contain the components removed from the maximal model fitted for LRT given in row 1. The maximum already has correlations within the participant-level removed. The notation is explicit – intercepts were not removed. `freq` = frequency, `len` = length, `surp` = surprisal designated as baseline for the respective model in Section 8.2.3. "`.1`" and "`.2`" signify lagged versions of predictors. "`||`" means that correlations between random effects are not accounted for. `-corr:level` signifies that correlations between random effects of the `level` grouping level were removed.

None of the predictors reached a p-value below the chosen significance threshold of 0.05, suggesting that the variations in reading times explained by the attention costs tested are too small to proclaim an effect rather than attributing them to random variation. For this reason, BH-corrected p-values are not provided It is peculiar that the standard LMs produce insignificant effects despite the attention costs computed on transformer models trained for pure language modelling reaching significance in experiment 1. The key differences between the models lie in the size of the architecture (16 parallel heads and 24 layers in experiment 1 vs 12 parallel heads and 12 layers here), in vocabulary size (222 319 vs 150 000) and in the numerical precision (16 bit vs 8 bit). As language modelling performance is comparable, it seems unlikely that precision lead to the disparity. The change in vocabulary size could have an effect caused by the model attributing more mass to the remaining tokens. Possibly, this made surprisal more predictive, inhibiting the ability of the attention costs to provide additional explanatory power. However, this would contradict with previous observations attributing a complementary effect to attention entropy (e.g. Ryu and Lewis, 2025). Furthermore, as discussed above, the perplexities on the WikiText validation split are very similar, with the models in experiment 2 having very slightly higher perplexity. Since perplexity is a function of surprisal (Definition 5.28), it follows that the change could not have been unidirectional, i.e. some tokens would need to receive higher surprisal in the models of this experiment to explain the overall perplexity. This can be neither explained by the change in vocabulary, nor by the change in model size.

Alternatively, it is possible that the smaller architecture decreased the predictive power of the emerging local unscaled attention costs. This options appears to contradict the results of experiment 1, insinuating that multiple heads in the large transformer model introduce noise and/or different kinds of computations which are not retrieval-like. However, if said model is overparameterised as the comparison in Section 8.2.2 suggested, this could imply that specific heads are even more strongly specialised, with types of retrievals less shared across heads and thus allowing focused and less diffuse attention patterns of individual heads to be detectible against the general attentional noise. On the other hand, this raises the question of what effect heads induce in a processing step whose specialisation is not needed at that point. Lastly, language modelling with transformers could necessitate non-retrieval like operations performed through attention heads (as discussed in Section 7.3). In this case, these non-retrieval-like functions could consume a fixed number of heads, thus leaving a lower ratio of heads in the smaller model to perform retrieval-like operations.

Finally, statistical inference is performed in this experiment with half of the observations as in experiment 1. However, comparing with the likelihood ratio statistics in experiment 1 (Table 15) reveals that the statistics are only a quarter as high (entropy: 6.51 vs 1.24 on average, distance: 6.07 vs 1.57 on average). Thus, it is evident that the statistics are not compromised solely by a change in reading time data.

Attention entropy The explanatory powers of attention entropy from the LM models measured using Δ_{llh} clearly surpass the scores of all CM^{entr} models (+1, +0.8, +0.4). Furthermore, for the pure language models, the estimated effect of attention entropy is approximately twice as large as for the cost models, albeit also insignificant.

Leaving aside the causes for the LMs not exhibiting a significant attention entropy effect, the comparison between models of the same architecture conducted here suggests that integrating attention entropy as an objective for transformer training is not improving its predictive quality.

Cost tested	Model	Run	Effect	SE	χ^2	p	Δ_{llh}	Δ_{AIC}	Δ_{BIC}
entropy	LM	1	1.27	1.82	0.88	0.349	1.20	1	10
		2	2.22	1.53	1.90	0.168	0.90	1	9
		3	1.56	1.51	0.95	0.329	0.50	1	10
	CM^{entr}	1	0.87	1.56	0.27	0.601	0.20	2	11
		2	0.48	1.54	0.09	0.768	0.10	1	11
		3	0.89	1.70	0.20	0.652	0.10	1	10
distance	LM	1	1.76	1.60	1.13	0.289	0.60	1	10
		2	3.01	2.00	2.00	0.158	1.00	0	9
		3	2.29	1.75	1.58	0.209	0.80	1	10
	CM^{dist}	1	-1.41	1.74	0.52	0.473	0.30	1	10
		2	-1.02	1.69	0.30	0.583	0.10	1	11
		3	0.05	1.64	0.00	0.983	0.00	2	11

Table 20: Results of likelihood-ratio tests for the effect of attention entropy and attention difference extracted from LMs and CMs on Provo go-past times. For each metric, I report the fixed-effect estimate (β; per 1 SD increase), standard error (SE), likelihood-ratio χ^2, raw p-value, Δ_{llh}, and information-criterion differences relative to the corresponding baseline model (ΔAIC, ΔBIC).

As the estimated coefficients are smaller than for the unconstrained models, it is possible that the optimisation procedure had a strong effect on tokens that produce high attention entropy which coincides with elevated reading times, reducing its variability. This contrasts initial expectations which assumed that inducing an optimisation loss for entropy would lead to a more cognitively calibrated distribution of interference-like effects.

Attention distance The patterns found for entropy repeat in a similar manner for attention distance: Δ_{llh} is higher in all LM models than for the corresponding CM^{dist} models (+0.3, +0.9, -0.8). The estimated effects for the cost models are not only smaller but two out of three are actually negative in sign (-1.41, -1.02). This means that the attention patterns are shaped such that attention focused on tokens that are close to the current token coincides with processing slowdown, or equivalently: such that long-range attention has facilitatory effects.

Possibly the optimisation procedure focused on cases of long-range attention that coincide with actual long-range dependencies and induced an encoding that allows to process them with extremely localised attention operations. This should in principle be supported by transformers through percolating information step- and layer-wise: if at each step in the input sequence, the attention head(s) in all layers attended to the representation of the directly preceding token and made this information available in its output, at step i the attention heads at layer j could have access to tokens at position $i - j$. With 12 layers, this seemingly local attention pattern would provide a context window of 12 tokens. However, as the effects observed here are far from significant, the estimates might very well be just uninterpretable artifacts.

8.3 Discussion

The second experiment investigated whether attention-based costs become more predictive of human reading times when they are included directly in the training objective of a transformer lan-

guage model. The results do not provide support for this hypothesis under the implementation tested here.

The hyperparameter searches yielded a first indication in this direction. For both attention entropy and attention distance, the best-performing configurations assigned most of the loss weight to language modelling. This suggests that preserving the next-token prediction objective was crucial for obtaining model-derived predictors with high psychometric explanatory power. The result is compatible with the broader observation that language modelling quality and reading time predictiveness are closely related (Goodkind and Bicknell, 2018; Wilcox et al., 2020). At the same time, the searches did not show that attention costs are entirely irrelevant. In particular, the search for attention distance exhibited a non-monotonic region in which intermediate cost weights produced elevated LMEM likelihoods (Figure 21). This indicates that attention-cost regularisation can affect the structure of the predictors in a way that is not reducible to language modelling loss alone. However, this effect did not translate into a reliable advantage over pure language models in the final evaluation. This is likely connected to a deterioration in language modelling performance, taking an effect on surprisal's predictive power.

The final model comparisons were more decisive. Across all three runs, attention entropy extracted from pure LMs yielded larger improvements in log-likelihood than attention entropy extracted from the corresponding $\mathrm{CM}^{\mathrm{entr}}$ models. The same pattern was found for attention distance: the LM-derived distance predictors outperformed the $\mathrm{CM}^{\mathrm{dist}}$ predictors in all three paired comparisons. None of the attention-cost predictors reached significance in the likelihood-ratio tests. Therefore, the cost models did not yield stronger reading-time predictors than the pure language models. This is incompatible with hypothesis [H2] as formulated in Section 1.2.

One possible interpretation is that attention entropy and attention distance are useful as diagnostic quantities, but not as direct optimisation targets. In a pure language model, diffuse or long-range attention may emerge at precisely those positions where the model requires more complex contextualisation. Under this interpretation, high entropy or high distance is not merely an inefficient state to be avoided, but a trace of the model performing a computation that benefits its predictive abilities. Under analogy, the human goal of comprehension might strongly surpass the urge of more parsimonious resource usage. Penalising these quantities during training may therefore remove part of the variance that made them psycholinguistically informative in the first place. This suggests that attention entropy is a necessary by-product stemming from the efficient organisation of information partaking in retrieval (keys in memory and queries) optimsed towards a higher-level goal. Mechanistically, entropy minimisation may alter the query-key geometry of the model. Reducing competition between context items could require making their representations more separable in attention space. While this may lower attention entropy, it may also weaken the model's ability to retrieve partially matching context items when such partial matches are useful for prediction. It is an open question what this could teach us about the representational level of cognitive units, since under the cue-based retrieval analogy, the regularised query/key vectors would correspond to chunks in memory.

A related possibility concerns ambiguity. Psycholinguistic findings suggest that ambiguity can have facilitatory effects when multiple analyses remain compatible with the input and no immediate disambiguation is required (Van Gompel and Pickering, 2001). If diffuse attention partly reflects the maintenance of several compatible retrieval candidates, then reducing entropy may not always be beneficial. It could force the model toward more selective retrieval before the input provides sufficient evidence for such a decision. This interpretation is speculative, since the present

experiment did not directly manipulate ambiguity, but it illustrates why lower attention entropy should not automatically be equated with more human-like processing.

Finally, a third interpretation would point to the implementation at hand being inadequate to test the underlying hypothesis. In particular, this could pertain to the minimisation constraint targeting all attention heads equally and simultaneously and the transformer architecture being inadequate as a model of human sentence processing. Unlike recurrent models, transformers do not maintain a single hidden state that is updated sequentially. Instead, each token representation is contextualised through attention over previous token representations. This architectural difference may partly explain why transformer attention is distributed across many heads and positions: attention must support not only retrieval-like operations, but also the propagation of contextual information that a recurrent architecture would encode in its hidden state. If that is the case, uniformly penalising all heads would affect computations that are not intended to correspond to memory retrieval.

9 Conclusion

This thesis investigated the relationship between human sentence processing and transformer-based language models. It brought together expectation-based accounts of reading difficulty, memory-based theories of sentence processing, behavioural reading-time measures and the internal computations of neural language models. The central question was whether attention-derived structural costs can complement surprisal in explaining human reading times and whether such costs become more informative when they are integrated directly into the training objective of a transformer language model.

The first experiment addressed this question by extracting several attention-based predictors from a transformer trained with a standard language modelling objective. In addition to attention entropy, which had already been linked to similarity-based interference in previous work (Ryu and Lewis, 2021; Oh and Schuler, 2022; Ryu and Lewis, 2025), three new measures were proposed: attention distance, activation and difference. They were evaluated in two different aggregation modes across the attention heads of the transformer model, with and without regularisation with respect to context size. The results provide support for hypotheses [H1a] and [H1b]: local unnormalised attention entropy and attention distance explained variance in go-past times beyond lexical surprisal and standard baseline predictors. Similar effects were observed for activation and difference.

However, the effects were small and the interpretation of these predictors requires caution. The attention-derived measures were strongly correlated with each other and with sentence position. This suggests that they capture a shared property of incremental processing, but the present analyses cannot determine whether this property is best understood as similarity-based interference, locality cost, recency, accumulated context size, or a combination of these factors.

The dependence on sentence position is especially important. Since go-past time includes regressions to earlier words, part of the observed effect may reflect the increasing opportunity for regressions as a sentence unfolds, rather than a direct cost of maintaining or retrieving a larger memory representation. Future work should therefore repeat the analysis with earlier eye-tracking measures, such as first fixation duration and gaze duration, as well as with self-paced-reading data, where regressions are not possible. Such comparisons would help determine whether the observed attention effects are tied to high-level integration processes or to properties of the eye-movement measure itself.

The first experiment also showed that the relationship between transformer attention and reading times depends on architectural and methodological choices. The model used here was larger than the GPT-2 small models commonly used in previous work on attention entropy and reading times. Nevertheless, the effects remained modest. Together with the comparison to the smaller models trained in the second experiment, this suggests that model size, number of layers, number of attention heads, tokenisation and the size of the context window can all affect the psycholinguistic usefulness of attention-derived predictors. A general theory of transformer-based sentence processing, such as proposed by (Ryu and Lewis, 2025), therefore needs to explain not only why some attention patterns correlate with reading behaviour, but also why this signal appears to depend on how attention is aggregated across heads and layers and what constitutes a cognitively plausible multi-head architecture.

The second experiment tested a stronger claim. If attention entropy and attention distance reflect

cognitively relevant processing costs, then training a transformer to minimise these costs alongside next-token surprisal might be expected to yield more human-like predictors of reading difficulty. This was tested by incorporating attention entropy and attention distance as additional loss terms during language model training. The loss weights were selected by hyperparameter search using reading-time predictiveness on a separate split of the Provo corpus.

The results do not support hypothesis [H2] under the implementation tested here. The hyperparameter searches favoured configurations with high weights on the language modelling objective and only weak weights on the attention-cost terms. In the final paired comparisons, attention costs extracted from pure language models were consistently at least as predictive as the corresponding costs extracted from cost models. Moreover, none of the attention-cost predictors reached significance in the final likelihood-ratio tests on the held-out Provo test split. Thus, explicitly minimising attention entropy or attention distance did not improve reading-time prediction.

This negative result is still informative. The cost models did not collapse: their language modelling performance was only moderately worse than that of the pure language models and their surprisal predictors remained broadly comparable in psychometric quality. This indicates that transformer models can adapt to weak constraints on the dispersion and locality of attention without losing their next-token prediction ability entirely. However, the resulting attention patterns were not more predictive of human reading times. This suggests that attention-derived structural costs may be useful as diagnostic quantities, but not necessarily as direct optimisation targets in the simple form tested here.

The results also reinforce a limitation of interpreting transformer attention as a direct model of human memory access. A transformer contains many attention heads across many layers and these heads need not correspond to cognitively plausible retrieval operations. Some heads may perform retrieval-like contextualisation, while others may encode positional, lexical, or task-specific information needed for language modelling. Uniformly averaging or uniformly constraining all heads may therefore dilute or distort the behaviourally relevant signal. This is consistent with previous work showing that psycholinguistically relevant effects can be concentrated in specific attention heads (Ryu and Lewis, 2021; Ryu and Lewis, 2022) and with the concern that the parallelism of transformer attention is not compatible with a cognitively plausible single-retrieval mechanism (Timkey and Linzen, 2023).

This thesis has several methodological limitations. First, all behavioural analyses are correlational: the linguistic predictors were derived from naturally occurring corpus material rather than experimentally manipulated. Consequently, the results cannot establish causal processing mechanisms. Second, the final evaluation of the second experiment was based on a split of the Provo corpus, reducing statistical power. Third, the implementation of attention-cost minimisation was deliberately simple. The same penalty was applied across heads, layers, tokens and contexts, although only some attention operations may be relevant to human-like retrieval. Future work should address these limitations in several ways. First, the analyses should be repeated across multiple reading-time corpora and multiple dependent measures, including first fixation duration, gaze duration, go-past time and self-paced reading times. Second, attention aggregation should be made more selective. Learned head weights or weights based on output projections may preserve behaviourally relevant structure better than uniform averaging and may provide more insight why certain heads are more predictive of reading times than others, compared to simply identifying the ones that correlate strongest with syntactic dependencies, such as (Ryu and Lewis, 2021). Third, architectures with a more explicit separation between prediction and retrieval could be ex-

plored. Such models may offer a better testing ground for memory-based theories than standard transformers, whose attention mechanism serves many computational roles at once.

Finally, this work contributes a flexible experimental framework for investigating these questions. The implementation supports multi-GPU training, large-scale hyperparameter search, and on-the-fly integration of linear mixed-effects modelling. Together with dataset loaders and cleaning procedures for the freely available psycholinguistic datasets presented in this work and flexible evaluation pipelines, this framework provides a basis for future research, including extensions to other training objectives.

In summary, the thesis shows that attention-derived structural costs provide a small but systematic complement to surprisal in modelling human reading behaviour and that attention entropy is not unique in this regard. However, directly optimising these costs does not improve reading-time prediction in the present setting. The most plausible conclusion is therefore not that memory-based constraints are irrelevant, but that their relationship to transformer attention is indirect.

References

Doris Aaronson and Hollis S. Scarborough (Feb. 1976). "Performance theories for sentence coding: Some quantitative evidence." In: *Journal of Experimental Psychology: Human Perception and Performance* 2.1, pp. 56–70. ISSN: 0096-1523.

Syed Ahmed, Christian Sarofeen, Mike Ruberry, Eddie Yan, Natalia Gimelshein, Michael Carilli, Szymon Migacz, Piotr Bialecki, Paulius Micikevicius, Dusan Stosic, Dong Yang, and Naoya Maruyama (July 19, 2022). *What Every User Should Know About Mixed Precision Training in PyTorch – PyTorch*. en-US. URL: https://pytorch.org/blog/what-every-user-should-know-about-mixed-precision-training-in-pytorch/ (visited on 05/05/2026).

Alfred V. Aho and Jeffrey D. Ullman (1972). *The theory of parsing, translation, and compiling*. USA: Prentice-Hall, Inc. ISBN: 0139145567.

August Ahrens (1891). "Untersuchungen über die Bewegung der Augen beim Schreiben". PhD thesis. Rostock: medicinische Facultät der Universität Rostock. URL: https://books.google.de/books?id=KOuHuXr3NnsC.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama (2019). "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Mohamed Taher Alrefaie, Nour Eldin Morsy, and Nada Samir (2024). *Exploring Tokenization Strategies and Vocabulary Sizes for Enhanced Arabic Language Models*. arXiv: 2403.11130 [cs.CL]. URL: https://arxiv.org/abs/2403.11130.

Gerry T. M. Altmann and Jelena Mirković (Apr. 2009). "Incrementality and Prediction in Human Sentence Processing". In: *Cognitive Science* 33.4, pp. 583–609. ISSN: 1551-6709.

John R. Anderson (Apr. 1996). "ACT: A simple theory of complex cognition." In: *American Psychologist* 51.4, pp. 355–365. ISSN: 0003-066X.

John R. Anderson (May 2005). "Human Symbol Manipulation Within an Integrated Cognitive Architecture". In: *Cognitive Science* 29.3, pp. 313–341. ISSN: 1551-6709.

John R. Anderson, Daniel Bothell, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin (2004). "An Integrated Theory of the Mind". In: *Psychological Review* 111.4, pp. 1036–1060. ISSN: 1939-1471.

Suhas Arehalli, Brian Dillon, and Tal Linzen (Aug. 2023). *Syntactic Surprisal From Neural Models Predicts, But Underestimates, Human Processing Difficulty From Syntactic Ambiguities*. arXiv:2210.12187 [cs]. URL: http://arxiv.org/abs/2210.12187.

David Arps, Hassan Sajjad, and Laura Kallmeyer (2025). *Understanding Syntactic Generalization in Structure-inducing Language Models*. arXiv: 2508.07969 [cs.CL]. URL: https://arxiv.org/abs/2508.07969.

Giuseppe Attanasio, Debora Nozza, Dirk Hovy, and Elena Baralis (May 2022). "Entropy-based Attention Regularization Frees Unintended Bias Mitigation from Lists". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, pp. 1105–1119. URL: https://aclanthology.org/2022.findings-acl.88/.

Christoph Aurnhammer and Stefan L. Frank (2019). "Evaluating information-theoretic measures of word prediction in naturalistic sentence reading". In: *Neuropsychologia* 134, p. 107198. ISSN: 0028-3932. URL: https://www.sciencedirect.com/science/article/pii/S0028393219302404.

Khai Loong Aw, Syrielle Montariol, Badr AlKhamissi, Martin Schrimpf, and Antoine Bosselut (2024). *Instruction-tuning Aligns LLMs to the Human Brain*. arXiv: 2312.00575 [cs.CL]. URL: https://arxiv.org/abs/2312.00575.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton (2016). *Layer Normalization*. arXiv: 1607.06450 [stat.ML]. URL: https://arxiv.org/abs/1607.06450.

R. H. Baayen, D. J. Davidson, and D. M. Bates (Nov. 2008). "Mixed-effects modeling with crossed random effects for subjects and items". In: *Journal of Memory and Language*. Special Issue: Emerging Data Analysis 59.4, pp. 390–412. ISSN: 0749-596X. URL: https://www.sciencedirect.com/science/article/pii/S0749596X07001398.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio (2016). *Neural Machine Translation by Jointly Learning to Align and Translate*. arXiv: 1409.0473 [cs.CL]. URL: https://arxiv.org/abs/1409.0473.

Karl G. D. Bailey and Fernanda Ferreira (2003). "Disfluencies affect the parsing of garden-path sentences". In: *Journal of Memory and Language* 49.2, pp. 183–200. ISSN: 0749-596X. URL: https://www.sciencedirect.com/science/article/pii/S0749596X03000275.

Dale J. Barr, Roger Levy, Christoph Scheepers, and Harry J. Tily (Apr. 2013). "Random effects structure for confirmatory hypothesis testing: Keep it maximal". In: *Journal of Memory and Language* 68.3, pp. 255–278. ISSN: 0749-596X. URL: https://www.sciencedirect.com/science/article/pii/S0749596X12001180.

Maria Jung Barrett, Zeljko Agic, and Anders Søgaard (2015). "The Dundee Treebank". English. In: *Proceedings of the Fourteenth International Workshop on Treebanks and Linguistic Theories*. Association for Computational Linguistics, pp. 242–248. ISBN: 978-83-63159-18-4.

Douglas Bates, Reinhold Kliegl, Shravan Vasishth, and Harald Baayen (2018). *Parsimonious Mixed Models*. arXiv: 1506.04967 [stat.ME]. URL: https://arxiv.org/abs/1506.04967.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker (2015). "Fitting Linear Mixed-Effects Models Using lme4". In: *Journal of Statistical Software* 67.1, pp. 1–48. URL: https://www.jstatsoft.org/index.php/jss/article/view/v067i01.

Iz Beltagy, Matthew E. Peters, and Arman Cohan (2020). *Longformer: The Long-Document Transformer*. arXiv: 2004.05150 [cs.CL]. URL: https://arxiv.org/abs/2004.05150.

Y. Bengio, P. Simard, and P. Frasconi (Mar. 1994). "Learning long-term dependencies with gradient descent is difficult". In: *IEEE Transactions on Neural Networks* 5.2, pp. 157–166. ISSN: 1941-0093. URL: https://ieeexplore.ieee.org/document/279181.

Yoav Benjamini and Yosef Hochberg (1995). "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 57.1, pp. 289–300. eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1995.tb02031.x. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1995.tb02031.x.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl (2011). "Algorithms for Hyper-Parameter Optimization". In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger. Vol. 24. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

James Bergstra, Daniel Yamins, and David Cox (2013). "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 1. Atlanta, Georgia, USA: PMLR, pp. 115–123. URL: https://proceedings.mlr.press/v28/bergstra13.html.

Steven Bird, Ewan Klein, and Edward Loper (Jan. 2009). *Natural Language Processing with Python*. ISBN: 978-0-596-51649-9.

Christopher M. Bishop (Aug. 17, 2006). *Pattern Recognition and Machine Learning*. en. First. New York, USA: Springer New York. ISBN: 978-0-387-31073-2. URL: https://link.springer.com/book/9780387310732.

Kathryn Bock and Carol A Miller (1991). "Broken agreement". In: *Cognitive Psychology* 23.1, pp. 45–93. ISSN: 0010-0285. URL: https://www.sciencedirect.com/science/article/pii/0010028591900037.

Robert W. Booth and Ulrich W. Weger (Jan. 2013). "The function of regressions in reading: Backward eye movements allow rereading". In: *Memory & Cognition* 41.1, pp. 82–97. ISSN: 0090-502X.

George E. P. Box (Dec. 1976). "Science and Statistics". In: *Journal of the American Statistical Association* 71.356, pp. 791–799. ISSN: 0162-1459. URL: https://doi.org/10.1080/01621459.1976.10480949.

Stephen Boyd and Lieven Vandenberghe (Mar. 2004). *Convex Optimization*. Cambridge University Press. ISBN: 0521833787. URL: http://www.amazon.com/exec/obidos/redirect?tag=citeulike-20%5C&path=ASIN/0521833787.

Trevor Brothers and Gina R. Kuperberg (Feb. 2021). "Word predictability effects are linear, not logarithmic: Implications for probabilistic models of sentence comprehension". In: *Journal of Memory and Language* 116, p. 104174. ISSN: 0749-596X.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020). *Language Models are Few-Shot Learners*. arXiv: 2005.14165 [cs.CL]. URL: https://arxiv.org/abs/2005.14165.

Juliane Burmester, Katharina Spalek, and Isabell Wartenburger (2014). "Context updating during sentence comprehension: The effect of aboutness topic". In: *Brain and Language* 137, pp. 62–76. ISSN: 0093-934X. URL: https://www.sciencedirect.com/science/article/pii/S0093934X14001096.

Jon W. Carr, Valentina N. Pescuma, Michele Furlan, Maria Ktori, and Davide Crepaldi (Feb. 2022). "Algorithms for the automated correction of vertical drift in eye-tracking data". en. In: *Behavior Research Methods* 54.1, pp. 287–310. ISSN: 1554-3528. URL: https://doi.org/10.3758/s13428-021-01554-0.

Manuel Carreiras, Charles Clifton, and Jr (Sept. 2004). *The On-line Study of Sentence Comprehension: Eyetracking, ERPs and Beyond*. New York: Psychology Press. ISBN: 9780203509050.

G. Casella and R.L. Berger (2002). *Statistical Inference*. Duxbury advanced series in statistics and decision sciences. Thomson Learning. ISBN: 9780534243128. URL: https://books.google.de/books?id=0x_vAAAAMAAJ.

Benedetta Cevoli, Chris Watkins, and Kathleen Rastle (June 2022). "Prediction as a basis for skilled reading: insights from modern language models". In: *Royal Society Open Science* 9.6, p. 211837. ISSN: 2054-5703. eprint: https://royalsocietypublishing.org/rsos/article-pdf/doi/10.1098/rsos.211837/995593/rsos.211837.pdf. URL: https://doi.org/10.1098/rsos.211837.

Stanley F. Chen and Joshua Goodman (1999). "An empirical study of smoothing techniques for language modeling". In: *Computer Speech & Language* 13.4, pp. 359–394. ISSN: 0885-2308. URL: https://www.sciencedirect.com/science/article/pii/S0885230899901286.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio (Oct. 2014). "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, pp. 1724–1734. URL: https://aclanthology.org/D14-1179/.

Noam Chomsky (1965). *Aspects of the Theory of Syntax*. 50th ed. The MIT Press. URL: http://www.jstor.org/stable/j.ctt17kk81z.

Noem Chomsky (1957). *Syntactic structures.* Oxford, England: Mouton.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller (2022). *Rethinking Attention with Performers*. arXiv: 2009.14794 [cs.LG]. URL: https://arxiv.org/abs/2009.14794.

Agatha Christie (1920). *The mysterious affair at styles.* Retrieved by Cop et al. (2016) on 7 November, 2012, from Project Gutenberg. URL: http://www.gutenberg.org/.

Christian Clark, Byung-Doh Oh, and William Schuler (Jan. 2025). "Linear Recency Bias During Training Improves Transformers' Fit to Reading Times". In: *Proceedings of the 31st International Conference on Computational Linguistics*. Ed. by Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert. Abu Dhabi, UAE: Association for Computational Linguistics, pp. 7735–7747. URL: https://aclanthology.org/2025.coling-main.517/.

Herbert H. Clark (Aug. 1973). "The language-as-fixed-effect fallacy: A critique of language statistics in psychological research". In: *Journal of Verbal Learning and Verbal Behavior* 12.4, pp. 335–359. ISSN: 0022-5371. URL: https://www.sciencedirect.com/science/article/pii/S0022537173800143.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning (2019). *What Does BERT Look At? An Analysis of BERT's Attention*. arXiv: 1906.04341 [cs.CL]. URL: https://arxiv.org/abs/1906.04341.

Uschi Cop, Nicolas Dirix, Denis Drieghe, and Wouter Duyck (Apr. 2017). "Presenting GECO: An eyetracking corpus of monolingual and bilingual sentence reading". eng. In: *Behavior Research Methods* 49.2, pp. 602–615. ISSN: 1554-3528.

Uschi Cop, Denis Drieghe, and Wouter Duyck (Aug. 2015a). "Eye Movement Patterns in Natural Reading: A Comparison of Monolingual and Bilingual Reading of a Novel". en. In: *PLOS ONE* 10.8, e0134008. ISSN: 1932-6203. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0134008.

Uschi Cop, Emmanuel Keuleers, Denis Drieghe, and Wouter Duyck (Oct. 2015b). "Frequency effects in monolingual and bilingual natural reading". en. In: *Psychonomic Bulletin & Review*

22.5, pp. 1216–1234. ISSN: 1531-5320. URL: https://doi.org/10.3758/s13423-015-0819-2.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi (2019). "On the Relationship between Self-Attention and Convolutional Layers". In: *CoRR* abs/1911.03584. arXiv: 1911.03584. URL: http://arxiv.org/abs/1911.03584.

Thomas M. Cover and Joy A. Thomas (2006). *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience. ISBN: 0471241954.

Aron Culotta, Andrew McCallum, and Jonathan Betz (June 2006). "Integrating Probabilistic Extraction Models and Data Mining to Discover Relations and Patterns in Text". In: *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. Ed. by Robert C. Moore, Jeff Bilmes, Jennifer Chu-Carroll, and Mark Sanderson. New York City, USA: Association for Computational Linguistics, pp. 296–303. URL: https://aclanthology.org/N06-1038/.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov (2019). *Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context*. arXiv: 1901.02860 [cs.LG]. URL: https://arxiv.org/abs/1901.02860.

Joseph H. Danks (June 1986). "Identifying component processes in text comprehension: Comment on Haberlandt and Graesser." In: *Journal of Experimental Psychology: General* 115.2, pp. 193–197. ISSN: 0096-3445.

George B Dantzig (1951). "Application of the simplex method to a transportation problem". In: *Activity analysis and production and allocation*.

Tri Dao and Albert Gu (2024). "Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality". In: *Proceedings of the 41st International Conference on Machine Learning*. Ed. by Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp. Vol. 235. Proceedings of Machine Learning Research. PMLR, pp. 10041–10071. URL: https://proceedings.mlr.press/v235/dao24a.html.

DeepSeek-AI et al. (2024). *DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model*. arXiv: 2405.04434 [cs.CL]. URL: https://arxiv.org/abs/2405.04434.

E. B. Delabarre (July 1898). "A Method of Recording Eye-Movements". In: *The American Journal of Psychology* 9.4, p. 572. ISSN: 0002-9556.

Vera Demberg (2010). "A Broad-Coverage Modelof Prediction in Human Sentence Processing". PhD thesis. Institute for Communicating and Collaborative Systems, School of Informatics, University of Edinburgh. URL: https://www.coli.uni-saarland.de/~vera/PhDThesis.pdf.

Vera Demberg and Frank Keller (2008). "Data from eye-tracking corpora as evidence for theories of syntactic processing complexity". In: *Cognition* 109.2, pp. 193–210. ISSN: 0010-0277. URL: https://www.sciencedirect.com/science/article/pii/S0010027708001741.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer (2021). "8-bit Optimizers via Block-wise Quantization". In: *CoRR* abs/2110.02861. arXiv: 2110.02861. URL: https://arxiv.org/abs/2110.02861.

Jean Dezert and Frédéric Dambreville (June 2023). "Cross-Entropy and Relative Entropy of Basic Belief Assignments". In: pp. 1–8. URL: https://ieeexplore.ieee.org/document/10224207.

Tali Ditman, Phillip J. Holcomb, and Gina R. Kuperberg (Nov. 2007). "An investigation of concurrent ERP and self-paced reading methodologies". English. In: *Psychophysiology* 44.6, pp. 927–935. ISSN: 0048-5772.

R. Dodge (1903). "Five Types of Eye Movement (in the Horizontal Meridian Plane of the Field of Regard)". In: *American Journal of Physiology* VIII, pp. 307–329.

Raymond Dodge (1904). "The participation of the eye movements in the visual perception of motion". In: *Psychological Review* 11.1, pp. 1–14. ISSN: 1939-1471.

Raymond Dodge and Thomas Sparks Cline (1901). "The angle velocity of eye movements". In: *Psychological Review* 8.2, pp. 145–157. ISSN: 1939-1471.

Timothy Dozat and Christopher D. Manning (2017). *Deep Biaffine Attention for Neural Dependency Parsing*. arXiv: 1611.01734 [cs.CL]. URL: https://arxiv.org/abs/1611.01734.

Kate Ehrlich and Keith Rayner (1983). "Pronoun assignment and semantic integration during reading: eye movements and immediacy of processing". In: *Journal of Verbal Learning and Verbal Behavior* 22.1, pp. 75–87. ISSN: 0022-5371. URL: https://www.sciencedirect.com/science/article/pii/S0022537183800073.

Susan F. Ehrlich and Keith Rayner (1981). "Contextual effects on word perception and eye movements during reading". In: *Journal of Verbal Learning and Verbal Behavior* 20.6, pp. 641–655. ISSN: 0022-5371. URL: https://www.sciencedirect.com/science/article/pii/S0022537181902206.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah (2021). "A Mathematical Framework for Transformer Circuits". In: *Transformer Circuits Thread*. https://transformer-circuits.pub/2021/framework/index.html.

Ralf Engbert, Antje Nuthmann, Eike M. Richter, and Reinhold Kliegl (Oct. 2005). "SWIFT: a dynamical model of saccade generation during reading". eng. In: *Psychological Review* 112.4, pp. 777–813. ISSN: 0033-295X.

Michael A. A. Eskenazi (Mar. 2024). "Best practices for cleaning eye movement data in reading research". In: *Behavior Research Methods* 56.3, pp. 2083–2093. ISSN: 1554-351X.

Sidney Evaldo Leal, João Marcos Munguba Vieira, Erica Dos Santos Rodrigues, Elisângela Nogueira Teixeira, and Sandra Aluísio (Dec. 2020). "Using Eye-tracking Data to Predict the Readability of Brazilian Portuguese Sentences in Single-task, Multi-task and Sequential Transfer Learning Approaches". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Ed. by Donia Scott, Nuria Bel, and Chengqing Zong. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 5821–5831. URL: https://aclanthology.org/2020.coling-main.512/.

Brian Everitt (2002). *The Cambridge dictionary of statistics*. Cambridge, UK; New York: Cambridge University Press. ISBN: 9780521810999. URL: http://www.worldcat.org/search?qt=worldcat_org_all&q=052181099X.

Robert M. Fano (Mar. 1961). *Transmission of Information: A Statistical Theory of Communication*. en. Cambridge, MA, USA: MIT Press. ISBN: 9780262060011.

Maryia Fedzechkina, T. Florian Jaeger, and Elissa L. Newport (2012). "Language learners restructure their input to facilitate efficient communication". In: *Proceedings of the National Academy of Sciences* 109.44, pp. 17897–17902. eprint: https://www.pnas.org/doi/

pdf/10.1073/pnas.1215776109. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1215776109.

Fernanda Ferreira, Karl G. D. Bailey, and Vittoria Ferraro (2002). "Good-Enough Representations in Language Comprehension". In: *Current Directions in Psychological Science* 11.1, pp. 11–15. eprint: https://doi.org/10.1111/1467-8721.00158. URL: https://doi.org/10.1111/1467-8721.00158.

Fernanda Ferreira, K. Christianson, and A. Hollingworth (Jan. 2001). "Misinterpretations of garden-path sentences: Implications for models of sentence processing and reanalysis". In: *Journal of Psycholinguistic Research* 30.1. 13th Annual CUNY Conference on Human Sentence Processing, La Jolla, California, mar 30–apr 01, 2000, pp. 3–20. ISSN: 0090-6905.

Fernanda Ferreira and John M. Henderson (July 1990). "Use of Verb Information in Syntactic Parsing – Evidence From Eye-Movements and Word-By-Word Self-Paced Reading". English. In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 16.4, pp. 555–568. ISSN: 0278-7393.

R. A. Fisher (Jan. 1919). *XV.—The Correlation between Relatives on the Supposition of Mendelian Inheritance*. URL: https://doi.org/10.1017/s0080456800012163.

J.A. Fodor (1983). *The Modularity of Mind*. A Bradford Book. MIT Press. ISBN: 9780262260701. URL: https://books.google.de/books?id=0vg0AwAAQBAJ.

Victoria Fossum and Roger P. Levy (June 2012). "Sequential vs. Hierarchical Syntactic Models of Human Incremental Sentence Processing". In: *Proceedings of the 3rd Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2012)*. Ed. by David Reitter and Roger Levy. CMCL '12. Montréal, Canada: Association for Computational Linguistics, pp. 61–69. URL: https://aclanthology.org/W12-1706/.

Luca Franceschi, Michele Donini, Valerio Perrone, Aaron Klein, Cédric Archambeau, Matthias Seeger, Massimiliano Pontil, and Paolo Frasconi (2025). "Hyperparameter Optimization in Machine Learning". In: *Foundations and Trends® in Machine Learning* 18.6, pp. 1054–1201. ISSN: 1935-8245. URL: http://dx.doi.org/10.1561/2200000088.

Stefan L. Frank (2013). "Uncertainty Reduction as a Measure of Cognitive Load in Sentence Comprehension". In: *Topics in Cognitive Science* 5.3, pp. 475–494. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/tops.12025. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/tops.12025.

Stefan L. Frank (2017). "Word Embedding Distance Does not Predict Word Reading Time". In: *Annual Meeting of the Cognitive Science Society*. URL: https://api.semanticscholar.org/CorpusID:2742823.

Stefan L. Frank, Irene Fernandez Monsalve, Robin L. Thompson, and Gabriella Vigliocco (Dec. 2013). "Reading time data for evaluating broad-coverage models of English sentence processing". English. In: *Behavior Research Methods* 45.4, pp. 1182–1190. ISSN: 1554-351X.

Lyn Frazier (1978). "On Comprehending Sentences: Syntactic Parsing Strategies". In: *ETD Collection for University of Connecticut*. URL: https://www.researchgate.net/publication/27401728_On_Comprehending_Sentences_Syntactic_Parsing_Strategies.

Richard Futrell, Edward Gibson, and Roger P. Levy (2020). "Lossy-Context Surprisal: An Information-Theoretic Model of Memory Effects in Sentence Processing". en. In: *Cognitive Science* 44.3, e12814. ISSN: 1551-6709. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.12814.

Richard Futrell, Edward Gibson, Harry J. Tily, Idan Blank, Anastasia Vishnevetsky, Steven Piantadosi, and Evelina Fedorenko (May 2018). "The Natural Stories Corpus". In: *Proceedings of the*

Eleventh International Conference on Language Resources and Evaluation (LREC 2018). Ed. by Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://aclanthology.org/L18-1012/.

Richard Futrell, Kyle Mahowald, and Edward Gibsob (2015). "Large-scale evidence of dependency length minimization in 37 languages". In: *Proceedings of the National Academy of Sciences* 112.33, pp. 10336–10341. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1502134112. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1502134112.

David Gaddy, Mitchell Stern, and Dan Klein (June 2018). "What's Going On in Neural Constituency Parsers? An Analysis". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Ed. by Marilyn Walker, Heng Ji, and Amanda Stent. New Orleans, Louisiana: Association for Computational Linguistics, pp. 999–1010. URL: https://aclanthology.org/N18-1091/.

Philip Gage (1994). "A new algorithm for data compression". In: *The C Users Journal archive* 12, pp. 23–38. URL: https://api.semanticscholar.org/CorpusID:59804030.

Andrew Gelman and Jennifer Hill (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Analytical Methods for Social Research. Cambridge University Press.

Luke Gessler and Nathan Schneider (Dec. 2023). "Syntactic Inductive Bias in Transformer Language Models: Especially Helpful for Low-Resource Languages?" In: *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*. Ed. by Jing Jiang, David Reitter, and Shumin Deng. Singapore: Association for Computational Linguistics, pp. 238–253. URL: https://aclanthology.org/2023.conll-1.17/.

Edward Gibson (Aug. 1998). "Linguistic complexity: locality of syntactic dependencies". eng. In: *Cognition* 68.1, pp. 1–76. ISSN: 0010-0277.

Edward Gibson (2000). "The dependency locality theory: A distance-based theory of linguistic complexity". In: *Image, Language, Brain: Papers from theFirst Mind Articulation Project Symposium*. Cambridge, MA, US: The MIT Press, pp. 94–126. ISBN: 9780262133715.

Edward Gibson, Richard Futrell, Steven P. Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Levy (May 2019). "How Efficiency Shapes Human Language". In: *Trends in Cognitive Sciences* 23.5, pp. 389–407. ISSN: 1364-6613.

Mario Giulianelli, Luca Malagutti, Juan Luis Gastaldi, Brian DuSell, Tim Vieira, and Ryan Cotterell (2024). *On the Proper Treatment of Tokenization in Psycholinguistics*. arXiv: 2410.02691 [cs.CL]. URL: https://arxiv.org/abs/2410.02691.

Xavier Glorot and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, pp. 249–256. URL: https://proceedings.mlr.press/v9/glorot10a.html.

Yoav Goldberg (2017). *Neural Network Methods for Natural Language Processing*. Cham, Switzerland: Springer International Publishing. ISBN: 9783031021657. URL: https://link.springer.com/book/10.1007/978-3-031-02165-7.

Harvey Goldstein (May 17, 1995). *Multilevel Statistical Models*. Second. Kendall's Library of Statistics 3. Hodder Arnold. ISBN: 0340595299.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville (Nov. 2016). "Deep Feedforward Networks". en. In: *Deep Learning*. Ed. by Francis Bach. Adaptive Computation and Machine Learning series. Cambridge, MA, USA: MIT Press, pp. 183–184. ISBN: 9780262035613. URL: https://mitpress.mit.edu/9780262035613/deep-learning/.

Adam Goodkind and Klinton Bicknell (Jan. 2018). "Predictive power of word surprisal for reading times is a linear function of language model quality". In: *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*. Ed. by Asad Sayeed, Cassandra Jacobs, Tal Linzen, and Marten van Schijndel. Salt Lake City, Utah: Association for Computational Linguistics, pp. 10–18. URL: https://aclanthology.org/W18-0102/.

Peter C. Gordon, Randall Hendrick, and Marcus Johnson (2001). "Memory interference during language processing." In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 27.6, pp. 1411–1423. ISSN: 0278-7393.

Peter C. Gordon, Randall Hendrick, Marcus Johnson, and Yoonhyoung Lee (Nov. 2006). "Similarity-based interference during language comprehension: Evidence from eye tracking during reading". English. In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 32.6, pp. 1304–1321. ISSN: 0278-7393.

Peter C. Gordon, Randall Hendrick, and William H. Levine (Sept. 2002). "Memory-Load Interference in Syntactic Processing". EN. In: *Psychological Science* 13.5, pp. 425–430. ISSN: 0956-7976. URL: https://doi.org/10.1111/1467-9280.00475.

H. P. Grice (1975). "Logic and Conversation". In: *Syntax and Semantics: Vol. 3: Speech Acts*. Ed. by Peter Cole and Jerry L. Morgan. New York: Academic Press, pp. 41–58. URL: http://www.ucl.ac.uk/ls/studypacks/Grice-Logic.pdf.

Geoffrey R. Grimmett and David R. Stirzaker (1992). *Probability and Random Processes*. 2nd ed. New York, USA: Oxford Science Publications.

Daniel Grodner and Edward Gibson (2005). "Consequences of the Serial Nature of Linguistic Input for Sentenial Complexity". In: *Cognitive Science* 29.2, pp. 261–290. ISSN: 1551-6709.

Daniel Grodner, Edward Gibson, Vered Argaman, and Maria Babyonyshev (Mar. 2003). "Against repair-based reanalysis in sentence comprehension". eng. In: *Journal of Psycholinguistic Research* 32.2, pp. 141–166. ISSN: 0090-6905.

Albert Gu and Tri Dao (2024). *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. arXiv: 2312.00752 [cs.LG]. URL: https://arxiv.org/abs/2312.00752.

Karl Haberlandt (1994). "Methods in Reading Research". In: Morton Ann Gernsbacher. *Handbook of psycholinguistics*. San Diego, CA, US: Academic Press, pp. 1–31.

John Hale (2001). "A probabilistic earley parser as a psycholinguistic model". In: *NAACL '01: Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001*. Pittsburgh, Pennsylvania: Association for Computational Linguistics, pp. 1–8. URL: http://portal.acm.org/citation.cfm?id=1073336.1073357.

John Hale (Mar. 2003). "The Information Conveyed by Words in Sentences". en. In: *Journal of Psycholinguistic Research* 32.2, pp. 101–123. ISSN: 1573-6555. URL: https://doi.org/10.1023/A:1022492123056.

John Hale (2006). "Uncertainty About the Rest of the Sentence". en. In: *Cognitive Science* 30.4, pp. 643–672. ISSN: 1551-6709. URL: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog0000_64.

David A. Harville (1977). "Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems". In: *Journal of the American Statistical Association* 72.358, pp. 320–338. eprint: https://doi.org/10.1080/01621459.1977.10480998. URL: https://doi.org/10.1080/01621459.1977.10480998.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. arXiv: 1502.01852 [cs.CV]. URL: https://arxiv.org/abs/1502.01852.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (June 2016). "Deep Residual Learning for Image Recognition". In: ISSN: 1063-6919, pp. 770–778. URL: https://ieeexplore.ieee.org/document/7780459.

Dan Hendrycks and Kevin Gimpel (2016). "Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units". In: *CoRR* abs/1606.08415. arXiv: 1606.08415. URL: http://arxiv.org/abs/1606.08415.

Ewald Hering (1879). "Über Muskelgeräusche des Auges". In: *Sitzberichte der kaiserlichen Akademie derWissenschaften in Wien. Mathematisch-naturwissenschaftliche Klasse* 79, pp. 137–154.

John Hewitt and Christopher D. Manning (June 2019). "A Structural Probe for Finding Syntax in Word Representations". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4129–4138. URL: https://aclanthology.org/N19-1419/.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2012). "Improving neural networks by preventing co-adaptation of feature detectors". In: *CoRR* abs/1207.0580. arXiv: 1207.0580. URL: http://arxiv.org/abs/1207.0580.

Yosef Hochberg and Ajit C. Tamhane (1987). *Some General Theory of Multiple Comparison Procedures*. John Wiley & Sons, Ltd. ISBN: 9780470316672. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470316672.app1. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470316672.app1.

Sepp Hochreiter and Jürgen Schmidhuber (Nov. 1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780. ISSN: 0899-7667. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

Sven Hohenstein, Jochen Laubrock, and Reinhold Kliegl (Sept. 2010). "Semantic Preview Benefit in Eye Movements During Reading: A Parafoveal Fast-Priming Study". In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 36.5, pp. 1150–1170. ISSN: 0278-7393.

Nora Hollenstein, Federico Pirovano, Ce Zhang, Lena Jäger, and Lisa Beinborn (June 2021). "Multilingual Language Models Predict Human Reading Behavior". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou. Online: Association for Computational Linguistics, pp. 106–123. URL: https://aclanthology.org/2021.naacl-main.10/.

Nora Hollenstein, Jonathan Rotsztejn, Marius Troendle, Andreas Pedroni, Ce Zhang, and Nicolas Langer (Dec. 2018). "ZuCo, a simultaneous EEG and eye-tracking resource for natural sentence reading". English. In: *Scientific Data* 5.

Nora Hollenstein, Marius Troendle, Ce Zhang, and Nicolas Langer (May 2020). "ZuCo 2.0: A Dataset of Physiological Recordings During Natural Reading and Annotation". eng. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Ed. by Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis. Marseille, France: European Language Resources Association, pp. 138–146. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.18/.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd (2020). *spaCy: Industrial-strength Natural Language Processing in Python*.

Jens-Max Hopf, Markus Bader, Michael Meng, and Josef Bayer (Jan. 2003). "Is human sentence parsing serial or parallel?: Evidence from event-related brain potentials". In: *Cognitive Brain Research* 15.2, pp. 165–177. ISSN: 0926-6410. URL: https://www.sciencedirect.com/science/article/pii/S0926641002001490.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White (1989). "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5, pp. 359–366. ISSN: 0893-6080. URL: https://www.sciencedirect.com/science/article/pii/0893608089900208.

Coşku Can Horuz, Geoffrey Kasenbacher, Saya Higuchi, Sebastian Kairat, Jendrik Stoltz, Moritz Pesl, Bernhard A. Moser, Christoph Linse, Thomas Martinetz, and Sebastian Otte (2025). *The Resurrection of the ReLU*. arXiv: 2505.22074 [cs.LG]. URL: https://arxiv.org/abs/2505.22074.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman (2019). *Do Attention Heads in BERT Track Syntactic Dependencies?* arXiv: 1911.12246 [cs.CL]. URL: https://arxiv.org/abs/1911.12246.

Kuan-Jung Huang, Suhas Arehalli, Mari Kugemoto, Christian Muxica, Grusha Prasad, Brian Dillon, and Tal Linzen (2024). "Large-scale benchmark yields no evidence that language model surprisal explains syntactic disambiguation difficulty". In: *Journal of Memory and Language* 137, p. 104510. ISSN: 0749-596X. URL: https://www.sciencedirect.com/science/article/pii/S0749596X24000135.

Joey Huchette, Gonzalo Muñoz, Thiago Serra, and Calvin Tsay (2025). *When Deep Learning Meets Polyhedral Theory: A Survey*. arXiv: 2305.00241 [math.OC]. URL: https://arxiv.org/abs/2305.00241.

Edmund B. Huey (July 1898). "Preliminary Experiments in the Physiology and Psychology of Reading". In: *The American Journal of Psychology* 9.4, p. 575. ISSN: 0002-9556.

Michael R. Hughes and Thomas J. Fisher (Dec. 16, 2025). *Introduction to Statistical Modeling*. URL: https://tjfisher19.github.io/introStatModeling/ (visited on 05/04/2026).

Sam B. Hutton (2019). "Eye tracking methodology. An Introduction to its Scientific Foundations and Applications". In: *Eye Movement Research*. Ed. by Christoph Klein and Ulrich Ettinger. Studies in Neuroscience, Psychology and Behavioral Economics. Springer Cham, pp. 277–308. ISBN: 978-3-030-20083-1. URL: https://doi.org/10.1007/978-3-030-20085-5_1.

Jukka Hyönä and Johanna K. Kaakinen (2019). "Eye Movements During Reading. An Introduction to its Scientific Foundations and Applications". en. In: *Eye Movement Research*. Ed. by Christoph Klein and Ulrich Ettinger. Cham: Springer Cham, pp. 239–274. ISBN: 9783030200855. URL: https://doi.org/10.1007/978-3-030-20085-5_7.

Shinnosuke Isono (June 2024). "Category Locality Theory: A unified account of locality effects in sentence comprehension". In: *Cognition* 247, p. 105766. ISSN: 0010-0277. URL: https://www.sciencedirect.com/science/article/pii/S0010027724000520.

Lena A. Jäger, Felix Engelmann, and Shravan Vasishth (2017). "Similarity-based interference in sentence comprehension: Literature review and Bayesian meta-analysis". In: *Journal of Memory and Language* 94, pp. 316–339.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani (2021). "Linear Regression". en. In: ed. by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. New York, NY: Springer US, pp. 59–128. ISBN: 9781071614181. URL: https://doi.org/10.1007/978-1-0716-1418-1_3.

Kevin Jamieson and Ameet Talwalkar (2015). *Non-stochastic Best Arm Identification and Hyperparameter Optimization*. arXiv: 1502.07943 [cs.LG]. URL: https://arxiv.org/abs/1502.07943.

Louis Émile Javal (1878). "Essai sur la physiologie de la lecture". In: *Annales d'Oculistique* 80, pp. 242–253.

Richard Johansson and Pierre Nugues (May 2007). "Extended Constituent-to-Dependency Conversion for English". In: *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*. Ed. by Joakim Nivre, Heiki-Jaan Kaalep, Kadri Muischnek, and Mare Koit. Tartu, Estonia: University of Tartu, Estonia, pp. 105–112. URL: https://aclanthology.org/W07-2416/.

Daniel Jurafsky and James H. Martin (2026). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models*. 3rd. Online manuscript released January 6, 2026. URL: https://web.stanford.edu/~jurafsky/slp3/.

Marcel A. Just and Patricia A. Carpenter (1980). "A Theory of Reading – From Eye Fixations to Comprehension". In: *Psychological Review* 87.4, pp. 329–354. ISSN: 0033-295X.

Marcel A. Just and Patricia A. Carpenter (1992). "A capacity theory of comprehension: Individual differences in working memory." In: *Psychological Review* 99.1, pp. 122–149. ISSN: 0033-295X.

Marcel A. Just, Patricia A. Carpenter, and Jacqueline D. Woolley (1982). "Paradigms and processes in reading comprehension." In: *Journal of Experimental Psychology: General* 111.2, pp. 228–238. ISSN: 0096-3445.

Laura Kallmeyer (Aug. 2010). *Parsing Beyond Context-Free Grammars*. Springer. ISBN: 978-3-642-14845-3.

Bob Kapteijns and Florian Hintz (July 2021). "Comparing predictors of sentence self-paced reading times: Syntactic complexity versus transitional probability metrics". English. In: *PLOS ONE* 16.7. ISSN: 1932-6203.

Frank Keller, Subahshini Gunasekharan, Neil Mayo, and Martin Corley (Feb. 2009). "Timing accuracy of Web experiments: A case study using the WebExp software package". In: *Behavior Research Methods* 41.1, pp. 1–12. ISSN: 1554-3528.

Susan Kemper and Chiung-Ju Liu (2007). "Eye movements of young and older adults during reading". In: *Psychology and Aging* 22.1, pp. 84–93. ISSN: 1939-1498.

Alan Kennedy (2003). *The dundee corpus*. [CD-ROM].

Alan Kennedy, Robin Hill, and Joël Pynte (Aug. 2003). "The Dundee corpus". Poster presented at ECEM12: 12th European Conference on Eye Movements. ECEM12: 12th European Conference on Eye Movements. Poster presented at ECEM12: 12th European Conference on Eye Movements. Dundee, UK.

Alan Kennedy and Wayne S. Murray (1984). "Reading Without Eye Movements". In: *Theoretical and Applied Aspects of Eye Movement Research*. Ed. by Alastair G. Gale and Frank Johnson. Vol. 22. Advances in Psychology. North-Holland, pp. 187–193. URL: https://www.sciencedirect.com/science/article/pii/S0166411508618336.

Alan Kennedy, Joël Pynte, Wayne S. Murray, and Shirley-Anne Paul (2013). "Frequency and predictability effects in the Dundee Corpus: An eye movement analysis". In: *Quarterly Journal of Experimental Psychology* 66.3. PMID: 22643118, pp. 601–618. eprint: https://doi.org/10.1080/17470218.2012.676054. URL: https://doi.org/10.1080/17470218.2012.676054.

Samuel Kiegeland, Ethan Gotlieb Wilcox, Afra Amini, David Robert Reich, and Ryan Cotterell (Nov. 2024). "Reverse-Engineering the Reader". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, pp. 9367–9389. URL: https://aclanthology.org/2024.emnlp-main.526/.

Jonathan King and Marcel Adam Just (Oct. 1991). "Individual differences in syntactic processing: The role of working memory". In: *Journal of Memory and Language* 30.5, pp. 580–602. ISSN: 0749-596X. URL: https://www.sciencedirect.com/science/article/pii/0749596X9190027H.

Diederik P. Kingma and Jimmy Ba (2017). *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980 [cs.LG]. URL: https://arxiv.org/abs/1412.6980.

Christoph Klein, Divya Seernani, Chara Ioannou, Yoanna Schulz-Zhecheva, Monica Biscaldi, and Michael Kavšek (2019). "Typical and Atypical Development of Eye Movements". en. In: ed. by Christoph Klein and Ulrich Ettinger. Cham: Springer International Publishing, pp. 635–701. ISBN: 9783030200855. URL: https://doi.org/10.1007/978-3-030-20085-5_15.

Reinhold Kliegl, Antje Nuthmann, and Ralf Engbert (2006). "Tracking the mind during reading: The influence of past, present, and future words on fixation durations." In: *Journal of Experimental Psychology: General* 135.1, pp. 12–35. ISSN: 0096-3445.

Reinhard Kneser and Hermann Ney (1995). "Improved backing-off for M-gram language modeling". In: *1995 International Conference on Acoustics, Speech, and Signal Processing* 1, 181–184 vol.1. URL: https://api.semanticscholar.org/CorpusID:9685476.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui (Nov. 2020). "Attention is Not Only a Weight: Analyzing Transformers with Vector Norms". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, pp. 7057–7075. URL: https://aclanthology.org/2020.emnlp-main.574/.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui (Nov. 2021). "Incorporating Residual and Normalization Layers into Analysis of Masked Language Models". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 4547–4568. URL: https://aclanthology.org/2021.emnlp-main.373/.

Tom Kocmi and Ondřej Bojar (Sept. 2017). "Curriculum Learning and Minibatch Bucketing in Neural Machine Translation". In: *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. Ed. by Ruslan Mitkov and Galia An-

gelova. Varna, Bulgaria: INCOMA Ltd., pp. 379–386. URL: https://aclanthology.org/R17-1050/.

John F. Kolen and Stefan C. Kremer (2001). "Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies". In: *A Field Guide to Dynamical Recurrent Networks*. IEEE, pp. 237–243. ISBN: 9780470544037. URL: https://ieeexplore.ieee.org/document/5264952.

Lars Konieczny (Nov. 2000). "Locality and Parsing Complexity". en. In: *Journal of Psycholinguistic Research* 29.6, pp. 627–645. ISSN: 1573-6555. URL: https://doi.org/10.1023/A:1026528912821.

Henry Kučera and W. Nelson Francis (1964). *Manual of Information to accompany A Standard Corpus of Present-Day Edited American English, for use with Digital Computers.* Revised and amplified 1979. Providence, Rhode Island: Department of Linguistics, Brown University.

S. Kullback and R. A. Leibler (Mar. 1951). "On Information and Sufficiency". en. In: *The Annals of Mathematical Statistics* 22.1, pp. 79–86. ISSN: 0003-4851, 2168-8990. URL: https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-22/issue-1/On-Information-and-Sufficiency/10.1214/aoms/1177729694.full.

Gina R. Kuperberg and T. Florian Jaeger (2016). "What do we mean by prediction in language comprehension?" eng. In: *Language, Cognition and Neuroscience* 31.1, pp. 32–59. ISSN: 2327-3798.

Victor Kuperman, Sascha Schroeder, Cengiz Acartürk, Niket Agrawal, Dominick M. Alexandre, Lena S. Bolliger, Jan Brasser, César Campos-Rojas, Denis Drieghe, Dušica Filipović Đurđević, and et al. (2025). "New data on text reading in English as a second language: The Wave 2 expansion of the Multilingual Eye-Movement Corpus (MECO)". In: *Studies in Second Language Acquisition* 47.2, pp. 677–695.

Victor Kuperman, Noam Siegelman, Sascha Schroeder, Cengiz Acartürk, Svetlana Alexeeva, Simona Amenta, Raymond Bertram, Rolando Bonandrini, Marc Brysbaert, Daria Chernova, and et al. (2023). "Text reading in English as a second language: Evidence from the Multilingual Eye-Movements Corpus". In: *Studies in Second Language Acquisition* 45.1, pp. 3–37.

Tatsuki Kuribayashi, Yohei Oseki, Ana Brassard, and Kentaro Inui (Dec. 2022). "Context Limitations Make Neural Language Models More Human-Like". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 10421–10436. URL: https://aclanthology.org/2022.emnlp-main.712/.

Tatsuki Kuribayashi, Yohei Oseki, Takumi Ito, Ryo Yoshida, Masayuki Asahara, and Kentaro Inui (Aug. 2021). "Lower Perplexity is Not Always Human-Like". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Online: Association for Computational Linguistics, pp. 5203–5217. URL: https://aclanthology.org/2021.acl-long.405/.

Marta Kutas, Katherine A. DeLong, and Nathaniel J. Smith (Apr. 2011). "A Look around at What Lies Ahead: Prediction and Predictability in Language Processing". In: *Predictions in the Brain*. Oxford University Press, pp. 190–207. ISBN: 9780195395518.

Marta Kutas and S. A. Hillyard (Jan. 1984). "Brain potentials during reading reflect word expectancy and semantic association". eng. In: *Nature* 307.5947, pp. 161–163. ISSN: 0028-0836.

M. Lamare (1892). “Des mouvements des yeux dans la lecture”. In: *Bulletins et Mémoires de la SociétéFrançaise d’Ophthalmologie* 10, pp. 354–364.

E. L. Lehmann and Joseph P. Romano (2022). “Quadratic Mean Differentiable Families”. In: *Testing Statistical Hypotheses*. Springer International Publishing, pp. 651–707. ISBN: 9783030705787.

Roger Levy, Evelina Fedorenko, and Edward Gibson (2013). “The syntactic complexity of Russian relative clauses”. In: *Journal of Memory and Language* 69.4, pp. 461–495. ISSN: 0749-596X. URL: https://www.sciencedirect.com/science/article/pii/S0749596X12001209.

Roger P. Levy (Mar. 2008). “Expectation-based syntactic comprehension”. In: *Cognition* 106.3, pp. 1126–1177. ISSN: 0010-0277. URL: https://www.sciencedirect.com/science/article/pii/S0010027707001436.

Roger P. Levy (2013). “Memory and surprisal in human sentence comprehension”. In: Current issues in the psychology of language. New York, NY, US: Psychology Press, pp. 78–114.

Richard L. Lewis and Shravan Vasishth (May 2005). “An activation-based model of sentence processing as skilled memory retrieval”. In: *Cognitive Science* 29.3, pp. 375–419. ISSN: 0364-0213.

Richard L. Lewis, Shravan Vasishth, and Julie A. Van Dyke (Oct. 2006). “Computational principles of working memory in sentence comprehension”. In: *Trends in cognitive sciences* 10.10, pp. 447–454. ISSN: 1364-6613. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC2239011/.

Bingzhi Li, Guillaume Wisniewski, and Benoît Crabbé (2023). “Assessing the Capacity of Transformer to Abstract Syntactic Representations: A Contrastive Analysis Based on Long-distance Agreement”. In: *Transactions of the Association for Computational Linguistics* 11, pp. 18–33. URL: https://aclanthology.org/2023.tacl-1.2/.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar (2018). *Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization*. arXiv: 1603.06560 [cs.LG]. URL: https://arxiv.org/abs/1603.06560.

D. V. Lindley and A. F. M. Smith (1972). “Bayes Estimates for the Linear Model”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 34.1, pp. 1–41. ISSN: 0035-9246. URL: https://www.jstor.org/stable/2985048.

Tal Linzen and Florian Jaeger (June 2014). “Investigating the role of entropy in sentence processing”. In: *Proceedings of the Fifth Workshop on Cognitive Modeling and Computational Linguistics*. Ed. by Vera Demberg and Timothy O’Donnell. Baltimore, Maryland, USA: Association for Computational Linguistics, pp. 10–18. URL: https://aclanthology.org/W14-2002/.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer (2018). *Generating Wikipedia by Summarizing Long Sequences*. arXiv: 1801.10198 [cs.CL]. URL: https://arxiv.org/abs/1801.10198.

Tong Liu, Iza Škrjanec, and Vera Demberg (Aug. 2024). “Temperature-scaling surprisal estimates improve fit to human reading times – but does it do so for the “right reasons”?” In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, pp. 9598–9619. URL: https://aclanthology.org/2024.acl-long.519/.

Ilya Loshchilov and Frank Hutter (2017). *SGDR: Stochastic Gradient Descent with Warm Restarts*. arXiv: 1608.03983 [cs.LG]. URL: https://arxiv.org/abs/1608.03983.

Steven G. Luke and Kiel Christianson (Aug. 2016). "Limits on lexical prediction during reading". In: *Cognitive Psychology* 88, pp. 22–60. ISSN: 0010-0285. URL: https://www.sciencedirect.com/science/article/pii/S0010028516301384.

Steven G. Luke and Kiel Christianson (Apr. 2018). "The Provo Corpus: A large eye-tracking corpus with predictability norms". en. In: *Behavior Research Methods* 50.2, pp. 826–833. ISSN: 1554-3528. URL: https://doi.org/10.3758/s13428-017-0908-4.

Thang Luong, Timothy O'donnell, and Noah Goodman (Sept. 2015). "Evaluating Models of Computation and Storage in Human Sentence Processing". In: *Proceedings of the Sixth Workshop on Cognitive Aspects of Computational Language Learning*. Ed. by Robert Berwick, Anna Korhonen, Alessandro Lenci, Thierry Poibeau, and Aline Villavicencio. Lisbon, Portugal: Association for Computational Linguistics, pp. 14–21. URL: https://aclanthology.org/W15-2403/.

Maryellen C. MacDonald (2013). "How language production shapes language form and comprehension". In: *Frontiers in Psychology* Volume 4 - 2013. ISSN: 1664-1078. URL: https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2013.00226.

Maryellen C. MacDonald, Neal J. Pearlmutter, and Mark S. Seidenberg (1994). "The lexical nature of syntactic ambiguity resolution." In: *Psychological Review* 101.4, pp. 676–703. ISSN: 0033-295X.

David J. C. MacKay (2002). *Information Theory, Inference & Learning Algorithms*. USA: Cambridge University Press. ISBN: 0521642981.

Emma Marsden, Sophie Thompson, and Luke Plonsky (2018). "A methodological synthesis of self-paced reading in second language research". In: *Applied Psycholinguistics* 39.5, pp. 861–904.

Hannes Matuschek, Reinhold Kliegl, Shravan Vasishth, Harald Baayen, and Douglas Bates (June 2017). "Balancing Type I error and power in linear mixed models". In: *Journal of Memory and Language* 94, pp. 305–315. ISSN: 0749-596X. URL: https://www.sciencedirect.com/science/article/pii/S0749596X17300013.

R. Thomas McCoy, Robert Frank, and Tal Linzen (2020). "Does Syntax Need to Grow on Trees? Sources of Hierarchical Inductive Bias in Sequence-to-Sequence Networks". In: *Transactions of the Association for Computational Linguistics* 8. Ed. by Mark Johnson, Brian Roark, and Ani Nenkova, pp. 125–140. URL: https://aclanthology.org/2020.tacl-1.9/.

Brian McElree, Stephani Foraker, and Lisbeth Dyer (2003). "Memory structures that subserve sentence comprehension". In: *Journal of Memory and Language* 48.1, pp. 67–91. ISSN: 1096-0821.

Ken McRae, Michael J. Spivey-Knowlton, and Michael K. Tanenhaus (1998). "Modeling the Influence of Thematic Fit (and Other Constraints) in On-line Sentence Comprehension". In: *Journal of Memory and Language* 38.3, pp. 283–312. ISSN: 0749-596X. URL: https://www.sciencedirect.com/science/article/pii/S0749596X97925432.

Stephen Merity, Nitish Shirish Keskar, James Bradbury, and Richard Socher (2018). "Scalable Language Modeling: WikiText-103 on a Single GPU in 12 hours Extended". In: URL: https://api.semanticscholar.org/CorpusID:219962598.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher (2016). *WikiText-103: A Large Language Modeling Corpus*. en. Salesforce Research. URL: https://huggingface.co/datasets/Salesforce/wikitext.

Danny Merkx and Stefan L. Frank (June 2021). "Human Sentence Processing: Recurrence or Attention?" In: *Proceedings of the Workshop on Cognitive Modeling and Computational Lin-*

guistics. Ed. by Emmanuele Chersoni, Nora Hollenstein, Cassandra Jacobs, Yohei Oseki, Laurent Prévot, and Enrico Santus. Online: Association for Computational Linguistics, pp. 12–22. URL: https://aclanthology.org/2021.cmcl-1.2/.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu (2018). *Mixed Precision Training*. arXiv: 1710.03740 [cs.AI]. URL: https://arxiv.org/abs/1710.03740.

Lukas Mielczarek, Timothée Bernard, Laura Kallmeyer, Katharina Spalek, and Benoit Crabbé (Sept. 2025). "Modelling Expectation-based and Memory-based Predictors of Human Reading Times with Syntax-guided Attention". In: *Proceedings of the Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics (BriGap-2)*. Ed. by Timothée Bernard and Timothee Mickus. Düsseldorf, Germany: Association for Computational Linguistics, pp. 52–71. ISBN: 979-8-89176-317-3. URL: https://aclanthology.org/2025.brigap-1.7/.

George A. Miller and Noam Chomsky (1963). "Finitary Models of Language Users". In: *Handbook of Mathematical Psychology*. Ed. by D. Luce. John Wiley & Sons., pp. 2–419.

Jeff Mitchell, Mirella Lapata, Vera Demberg, and Frank Keller (July 2010). "Syntactic and Semantic Factors in Processing Difficulty: An Integrated Measure". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Ed. by Jan Hajič, Sandra Carberry, Stephen Clark, and Joakim Nivre. Uppsala, Sweden: Association for Computational Linguistics, pp. 196–206. URL: https://aclanthology.org/P10-1021/.

Aaron Mueller, Robert Frank, Tal Linzen, Luheng Wang, and Sebastian Schuster (May 2022). "Coloring the Blank Slate: Pre-training Imparts a Hierarchical Inductive Bias to Sequence-to-sequence Models". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, pp. 1352–1368. URL: https://aclanthology.org/2022.findings-acl.106/.

Sathvik Nair and Philip Resnik (Dec. 2023). "Words, Subwords, and Morphemes: What Really Matters in the Surprisal-Reading Time Relationship?" In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 11251–11260. URL: https://aclanthology.org/2023.findings-emnlp.752/.

I. Nation (Sept. 2006). "How Large a Vocabulary is Needed For Reading and Listening?" In: *The Canadian Modern Language Review* 63.1, pp. 59–82. ISSN: 0008-4506. URL: https://utppublishing.com/doi/10.3138/cmlr.63.1.59.

Yunhao Ni, Yuxin Guo, Junlong Jia, and Lei Huang (2024). *On the Nonlinearity of Layer Normalization*. arXiv: 2406.01255 [cs.LG]. URL: https://arxiv.org/abs/2406.01255.

Marcus Nyström and Kenneth Holmqvist (Feb. 2010). "An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data". en. In: *Behavior Research Methods* 42.1, pp. 188–204. ISSN: 1554-3528. URL: https://doi.org/10.3758/BRM.42.1.188.

Byung-Doh Oh, Christian Clark, and William Schuler (Aug. 2021). "Surprisal Estimators for Human Reading Times Need Character Models". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Online: Association for Computational Linguistics, pp. 3746–3757. URL: https://aclanthology.org/2021.acl-long.290/.

Byung-Doh Oh and William Schuler (Dec. 2022). *Entropy- and Distance-Based Predictors From GPT-2 Attention Patterns Predict Reading Times Over and Above GPT-2 Surprisal*. arXiv:2212.11185 [cs]. URL: http://arxiv.org/abs/2212.11185.

Byung-Doh Oh and William Schuler (Dec. 2023a). "Transformer-Based Language Model Surprisal Predicts Human Reading Times Best with About Two Billion Training Tokens". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 1915–1921. URL: https://aclanthology.org/2023.findings-emnlp.128/.

Byung-Doh Oh and William Schuler (2023b). "Why Does Surprisal From Larger Transformer-Based Language Models Provide a Poorer Fit to Human Reading Times?" In: *Transactions of the Association for Computational Linguistics* 11, pp. 336–350. URL: https://aclanthology.org/2023.tacl-1.20/.

Byung-Doh Oh and William Schuler (July 2025). "The Impact of Token Granularity on the Predictive Power of Language Model Surprisal". In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar. Vienna, Austria: Association for Computational Linguistics, pp. 4150–4162. ISBN: 979-8-89176-251-0. URL: https://aclanthology.org/2025.acl-long.209/.

Takeru Ohta (Jan. 17, 2020). *Comment on: Update 'TPESampler' to support 'HyperbandPruner' by crcrpar · Pull Request #828 · optuna/optuna*. en. URL: https://github.com/optuna/optuna/pull/828#issuecomment-575457360 (visited on 05/05/2026).

Yan Pan and Yuanzhi Li (2023). *Toward Understanding Why Adam Converges Faster Than SGD for Transformers*. arXiv: 2306.00204 [cs.LG]. URL: https://arxiv.org/abs/2306.00204.

Dan Parker, Michael Shvartsman, and Julie A. Van Dyke (2017). "The cue-based retrieval theory of sentence comprehension: New findings and new challenges". In: URL: https://api.semanticscholar.org/CorpusID:211267072.

H. D. Patterson and R. Thompson (1971). "Recovery of inter-block information when block sizes are unequal". In: *Biometrika* 58.3, pp. 545–554. ISSN: 1464-3510.

Jackson Petty and Robert Frank (2021). *Transformers Generalize Linearly*. arXiv: 2109.12036 [cs.CL]. URL: https://arxiv.org/abs/2109.12036.

Martin J. Pickering and Simon Garrod (Mar. 2007). "Do people use language production to make predictions during comprehension?" eng. In: *Trends in Cognitive Sciences* 11.3, pp. 105–110. ISSN: 1364-6613.

Martin J. Pickering and Simon Garrod (2013). "An integrated theory of language production and comprehension". In: *Behavioral and Brain Sciences* 36.4, pp. 329–347.

Tiago Pimentel, Clara Meister, Ethan Gotlieb Wilcox, Roger P. Levy, and Ryan Cotterell (Dec. 2023). "On the Effect of Anticipation on Reading Times". In: *Transactions of the Association for Computational Linguistics* 11, pp. 1624–1642. ISSN: 2307-387X. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00603/2196892/tacl_a_00603.pdf. URL: https://doi.org/10.1162/tacl_a_00603.

José C. Pinheiro and Douglas M. Bates (May 5, 2000). *Mixed-Effects Models in S and S-PLUS*. First. Statistics and Computing. Springer New York, NY. ISBN: 978-0-387-98957-0. URL: https://link.springer.com/book/10.1007/b98882.

Michael A. Poole and Patrick N. O'farrell (1971). "The Assumptions of the Linear Regression Model". In: *Transactions of the Institute of British Geographers*, p. 145. URL: https://api.semanticscholar.org/CorpusID:8290837.

Karl Popper (2005). *The logic of scientific discovery*. London/New York: Routledge. ISBN: 1134470029.

Pierre Pouget (2019). "Introduction to the Study of Eye Movements. An Introduction to its Scientific Foundations and Applications". en. In: *Eye Movement Research*. Ed. by Christoph Klein and Ulrich Ettinger. Cham: Springer Cham, pp. 3–10. ISBN: 9783030200855. URL: https://doi.org/10.1007/978-3-030-20085-5_7.

M. J. D. Powell (2009). *The BOBYQA algorithm for bound constrained optimization without derivatives*. URL: https://api.semanticscholar.org/CorpusID:2488733.

Ofir Press and Lior Wolf (2017). *Using the Output Embedding to Improve Language Models*. arXiv: 1608.05859 [cs.CL]. URL: https://arxiv.org/abs/1608.05859.

Ralph Radach, Lynn Huestegge, and Ronan Reilly (Nov. 2008). "The role of global top-down factors in local eye-movement control in reading". In: *Psychologische Research (formerly: Psychologische Forschung)* 72.6, pp. 675–688. ISSN: 0340-0727.

Ralph Radach, Albrecht Inhoff, Lisa Glover, and Christian Vorstius (Feb. 2013). "Contextual constraint and N + 2 preview effects in reading". In: *Quarterly journal of experimental psychology (2006)* 66.

Ralph Radach and Alan Kennedy (Mar. 2013). "Eye movements in reading: Some theoretical context". In: *Quarterly Journal of Experimental Psychology* 66.3, SI, pp. 429–452. ISSN: 1747-0218.

Alec Radford and Karthik Narasimhan (2018). "Improving Language Understanding by Generative Pre-Training". In: URL: https://api.semanticscholar.org/CorpusID:49313245.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). *Language Models are Unsupervised Multitask Learners*.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap (2019). *Compressive Transformers for Long-Range Sequence Modelling*. arXiv: 1911.05507 [cs.LG]. URL: https://arxiv.org/abs/1911.05507.

E. Raehlmann (Dec. 1878). "Ueber den Nystagmus und seine Aetiologie". de. In: *Albrecht von Graefes Archiv für Ophthalmologie* 24.4, pp. 237–317. ISSN: 1435-702X. URL: https://doi.org/10.1007/BF01945601.

Neil Rathi (June 2021). "Dependency Locality and Neural Surprisal as Predictors of Processing Difficulty: Evidence from Reading Times". In: *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*. Ed. by Emmanuele Chersoni, Nora Hollenstein, Cassandra Jacobs, Yohei Oseki, Laurent Prévot, and Enrico Santus. Online: Association for Computational Linguistics, pp. 171–176. URL: https://aclanthology.org/2021.cmcl-1.21/.

Keith Rayner (Nov. 1998). "Eye movements in reading and information processing: 20 years of research". In: *Psychological Bulletin* 124.3, pp. 372–422. ISSN: 0033-2909.

Keith Rayner (2009). "The 35th Sir Frederick Bartlett Lecture: Eye movements and attention in reading, scene perception, and visual search". In: *Quarterly Journal of Experimental Psychology* 62.8. PMID: 19449261, pp. 1457–1506. eprint: https://doi.org/10.1080/17470210902816461. URL: https://doi.org/10.1080/17470210902816461.

Keith Rayner (2015). "Eye Movements in Reading". In: *International Encyclopedia of the Social Behavioral Sciences (Second Edition)*. Ed. by James D. Wright. Second Edition. Oxford: Elsevier, pp. 631–634. ISBN: 978-0-08-097087-5. URL: https://www.sciencedirect.com/science/article/pii/B9780080970868540082.

Keith Rayner, J. Ashby, A. Pollatsek, and Ed. Reichle (Aug. 2004). "The effects of frequency and predictability on eye fixations in reading: Implications for the E-Z reader model". In: *Journal of Experimental Psychology: Human Perception and Performance* 30.4, pp. 720–732. ISSN: 0096-1523.

Keith Rayner, Gretchen Kambe, and Susan A. Duffy (Nov. 2000). "The Effect of Clause Wrap-Up on Eye Movements during Reading". In: *The Quarterly Journal of Experimental Psychology Section A* 53.4, pp. 1061–1080. ISSN: 1464-0740.

Keith Rayner, Sc. Sereno, and Ge. Raney (Oct. 1996). "Eye movement control in reading: A comparison of two types of models". In: *Journal of Experimental Psychology: Human Perception and Performance* 22.5, pp. 1188–1200. ISSN: 0096-1523.

Keith Rayner, Timothy J. Slattery, and Nathalie N. Belanger (Dec. 2010). "Eye movements, the perceptual span, and reading speed". In: *Psychonomic Bulletin & Review* 17.6, pp. 834–839. ISSN: 1069-9384.

Keith Rayner and Ad. Well (Dec. 1996). "Effects of contextual constraint on eye movements in reading: A further examination". In: *Psychonomic Bulletin & Review* 3.4, pp. 504–509. ISSN: 1069-9384.

Adrielli Tina Lopes Rego, Joshua Snell, and Martijn Meeter (Sept. 2024). "Language models outperform cloze predictability in a cognitive model of reading". en. In: *PLOS Computational Biology* 20.9, e1012117. ISSN: 1553-7358. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1012117.

Erik D. Reichle (Sept. 2021). "Models of Sentence Processing". In: *Computational Models of Reading: A Handbook*. Oxford University Press. ISBN: 9780195370669. eprint: https://academic.oup.com/book/0/chapter/339847574/chapter-pdf/42868281/oso-9780195370669-chapter-4.pdf. URL: https://doi.org/10.1093/oso/9780195370669.003.0004.

Brian Roark, Asaf Bachrach, Carlos Cardenas, and Christophe Pallier (Aug. 2009). "Deriving lexical and syntactic expectation-based measures for psycholinguistic modeling via incremental top-down parsing". In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Ed. by Philipp Koehn and Rada Mihalcea. Singapore: Association for Computational Linguistics, pp. 324–333. URL: https://aclanthology.org/D09-1034/.

Doug Rohde (2001). *Linger*. **a flexible platform for language processing experiments**. URL: https://web.archive.org/web/20191220181934/http://tedlab.mit.edu/~dr/Linger/.

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas (Nov. 2000). "The Earth Mover's Distance as a Metric for Image Retrieval". en. In: *International Journal of Computer Vision* 40.2, pp. 99–121. ISSN: 1573-1405. URL: https://doi.org/10.1023/A:1026543900054.

Rachel Ryskin and Mante S. Nieuwland (Nov. 2023). "Prediction during language comprehension: what is next?" In: *Trends in Cognitive Sciences* 27.11, pp. 1032–1052. ISSN: 1364-6613. URL: https://www.sciencedirect.com/science/article/pii/S1364661323001997.

Soo Hyun Ryu and Richard L. Lewis (June 2021). "Accounting for Agreement Phenomena in Sentence Comprehension with Transformer Language Models: Effects of Similarity-based Inter-

ference on Surprisal and Attention". In: *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*. Ed. by Emmanuele Chersoni, Nora Hollenstein, Cassandra Jacobs, Yohei Oseki, Laurent Prévot, and Enrico Santus. Online: Association for Computational Linguistics, pp. 61–71. URL: https://aclanthology.org/2021.cmcl-1.6/.

Soo Hyun Ryu and Richard L. Lewis (2022). "Using Transformer Language model to Integrate Surprisal, Entropy, and Working Memory Retrieval Accounts of Sentence Processing". In: *35th Annual Conference on Human Sentence Processing*. URL: https://virtual.oxfordabstracts.com/event/2562/submission/231.

Soo Hyun Ryu and Richard L. Lewis (Dec. 2025). "Memory for prediction: A Transformer-based theory of sentence processing". In: *Journal of Memory and Language* 145, p. 104670. ISSN: 0749-596X. URL: https://www.sciencedirect.com/science/article/pii/S0749596X25000634.

Jacqueline Strunk Sachs (Sept. 1967). "Recognition memory for syntactic and semantic aspects of connected discourse". In: *Perception & Psychophysics* 2.9, pp. 437–442. ISSN: 1532-5962.

Edward W. Samson (1951). *Tech Report No. E5079*. Air Force Cambridge Research Center.

Edward W. Samson (1953). "Fundamental Natural Concepts of Information Theory". In: *ETC: A Review of General Semantics* 10.4, pp. 283–297. ISSN: 0014164X, 21689245. URL: http://www.jstor.org/stable/42581366.

Laurent Sartran, Samuel Barrett, Adhiguna Kuncoro, Miloš Stanojević, Phil Blunsom, and Chris Dyer (2022). "Transformer Grammars: Augmenting Transformer Language Models with Syntactic Inductive Biases at Scale". In: *Transactions of the Association for Computational Linguistics* 10. Ed. by Brian Roark and Ani Nenkova, pp. 1423–1439. URL: https://aclanthology.org/2022.tacl-1.81/.

Marten van Schijndel and Tal Linzen (2021). "Single-Stage Prediction Models Do Not Explain the Magnitude of Syntactic Disambiguation Difficulty". en. In: *Cognitive Science* 45.6, e12988. ISSN: 1551-6709. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.12988.

Darryl W. Schneider and John R. Anderson (May 2012). "Modeling fan effects on the time course of associative recognition". In: *Cognitive Psychology* 64.3, pp. 127–160. ISSN: 0010-0285.

Elizabeth R. Schotter, Bernhard Angele, and Keith Rayner (Oct. 2011). "Parafoveal processing in reading". In: *Attention, Perception, amp; Psychophysics* 74.1, pp. 5–35. ISSN: 1943-393X.

Martin Schrimpf, Idan Asher Blank, Greta Tuckute, Carina Kauf, Eghbal A. Hosseini, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko (Nov. 2021). "The neural architecture of language: Integrative modeling converges on predictive processing". In: *Proceedings of the National Academy of Sciences* 118.45, e2105646118. URL: https://www.pnas.org/doi/10.1073/pnas.2105646118.

Sascha Schroeder (Aug. 2019). *popEye - An R package to analyse eye movement data from reading experiments*. URL: https://github.com/sascha2schroeder/popEye.

Irina A. Sekerina, Eva M. Fernández, Harald Clahsen, Irina A. Sekerina, Eva M. Fernández, and Harald Clahsen (2008). *Developmental psycholinguistics : on-line methods in children's language processing*. eng. Language acquisition and language disorders ; 44. Amsterdam: Benjamins.

Rico Sennrich, Barry Haddow, and Alexandra Birch (Aug. 2016). "Neural Machine Translation of Rare Words with Subword Units". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Katrin Erk and Noah A. Smith. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725. URL: https://aclanthology.org/P16-1162/.

William R. Shadish, Thomas D. Cook, and Donald T. Campbell (2002). *Experimental and quasi-experimental designs for generalized causal inference*. Experimental and quasi-experimental designs for generalized causal inference. Boston, MA, US: Houghton, Mifflin and Company, pp. xxi, 623. ISBN: 9780395615560.

Cory Shain (June 2019). "A large-scale study of the effects of word frequency and predictability in naturalistic reading". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4086–4094. URL: https://aclanthology.org/N19-1413/.

Cory Shain, Clara Meister, Tiago Pimentel, Ryan Cotterell, and Roger P. Levy (Nov. 2022). *Large-Scale Evidence for Logarithmic Effects of Word Predictability on Reading Time*. URL: osf.io/preprints/psyarxiv/4hyna_v1.

C. E. Shannon (July 1948). "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3, pp. 379–423. ISSN: 0005-8580. URL: https://ieeexplore.ieee.org/document/6773024.

H. S. Sichel (Jan. 1974). "On a Distribution Representing Sentence-Length in Written Prose". In: *Royal Statistical Society. Journal. Series A: General* 137.1, pp. 25–34. ISSN: 0035-9238. eprint: https://academic.oup.com/jrsssa/article-pdf/137/1/25/49753309/jrsssa_137_1_25.pdf. URL: https://doi.org/10.2307/2345142.

Noam Siegelman and Victor Kuperman (2025). "The Multilingual Eye-Movement Corpus (MECO): A Collection of Eye-Tracking Reading Data Worldwide". In: *Reference Module in Social Sciences*. Elsevier. ISBN: 978-0-443-15785-1. URL: https://www.sciencedirect.com/science/article/pii/B9780323955041004944.

Noam Siegelman, Sascha Schroeder, Cengiz Acartuerk, Hee-Don Ahn, Svetlana Alexeeva, Simona Amenta, Raymond Bertram, Rolando Bonandrini, Marc Brysbaert, Daria Chernova, Sara Maria Da Fonseca, Nicolas Dirix, Wouter Duyck, Argyro Fella, Ram Frost, Carolina A. Gattei, Areti Kalaitzi, Nayoung Kwon, Kaidi Loo, Marco Marelli, Timothy C. Papadopoulos, Athanassios Protopapas, Satu Savo, Diego E. Shalom, Natalia Slioussar, Roni Stein, Longjiao Sui, Anali Taboh, Veronica Tonnesen, Kerem Alp Usal, and Victor Kuperman (Dec. 2022). "Expanding horizons of cross-linguistic research on reading: The Multilingual Eye-movement Corpus (MECO)". In: *Behavior Research Methods* 54.6, pp. 2843–2863. ISSN: 1554-351X.

Noam Siegelman, Sascha Schroeder, Yaqian Borogjoon Bao, Cengiz Acarturk, Niket Agrawal, Lena S. Bolliger, Jan Brasser, Cesar Campos-Rojas, Denis Drieghe, Dusica Filipovic Durdevic, Sofya Goldina, Romualdo Ibanez Orellana, Lena A. Jager, Omar I. Johannesson, Anurag Khare, Nik Kharlamov, Hanne B. S. Knudsen, Arni Kristjansson, Charlotte E. Lee, Jun Ren Lee, Marina P. T. Leite, Simona Mancini, Natasa Mihajlovic, Ksenija Misic, Miloslava Orekhova, Olga Parshina, Milica Popovic Stijacic, Athanassios Protopapas, David R. Reich, Anurag Rimzhim, Rui Rothe-Neves, Thais M. M. Sa, Andrea Santana-Covarrubias, Irina Sekerina, Heida M. Sigurdardottir, Anna Smirnova, Priyanka Srivastava, Elisangela N. Teixeira, Ivana Ugrinic, Kerem Alp Usal, Karolina Vakulya, Ark Verma, Joao M. M. Vieira, Denise H. Wu, Jin Xue, Suncica Zdravkovic, Junjing Zhuo, Laoura Ziaka, and Victor Kuperman (July 2025). "Wave 2 of the Multilingual Eye-Movement Corpus (MECO): New text reading data across languages". In: *Scientific Data* 12.1.

Fabio Sigrist (June 2021). *Generalized Linear Mixed Effects Models in R and Python with GPBoost*. en. URL: https://medium.com/data-science/generalized-

linear-mixed-effects-models-in-r-and-python-with-gpboost-89297622820c.

Fabio Sigrist (2022). "Gaussian Process Boosting". In: *Journal of Machine Learning Research* 23.232, pp. 1–46. URL: http://jmlr.org/papers/v23/20-322.html.

Fabio Sigrist (Apr. 2026). *fabsig/GPBoost*. original-date: 2020-04-03T15:15:15Z. URL: https://github.com/fabsig/GPBoost.

Sophie Slaats and Andrea E. Martin (June 2025). "What's Surprising About Surprisal". en. In: *Computational Brain & Behavior* 8.2, pp. 233–248. ISSN: 2522-087X. URL: https://doi.org/10.1007/s42113-025-00237-9.

Nathaniel J. Smith and Roger P. Levy (Sept. 2013). "The effect of word predictability on reading time is logarithmic". In: *Cognition* 128.3, pp. 302–319. ISSN: 0010-0277. URL: https://www.sciencedirect.com/science/article/pii/S0010027713000413.

Ming Song, Jing Wang, and Qing Cai (Dec. 2024). "The unique contribution of uncertainty reduction during naturalistic language comprehension". In: *Cortex* 181, pp. 12–25. ISSN: 0010-9452. URL: https://www.sciencedirect.com/science/article/pii/S0010945224002545.

Robyn Speer (Sept. 2022). *rspeer/wordfreq: v3.0*. Version v3.0.2. URL: https://doi.org/10.5281/zenodo.7199437.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Trevor Standley, Amir R. Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese (2020). *Which Tasks Should Be Learned Together in Multi-task Learning?* arXiv: 1905.07553 [cs.CV]. URL: https://arxiv.org/abs/1905.07553.

Mark Steedman (1996). *Surface Structure and Interpretation*. Linguistic inquiry monographs. MIT Press. ISBN: 9780262691932. URL: https://books.google.de/books?id=Mh1vQgAACAAJ.

Andreas Stolcke (1995). "An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilities". In: *Computational Linguistics* 21.2. Ed. by Julia Hirschberg, pp. 165–201. URL: https://aclanthology.org/J95-2002/.

Patrick Sturt and Vincenzo Lombardo (Mar. 2005). "Processing Coordinated Structures: Incrementality and Connectedness". In: *Cognitive Science* 29.2, pp. 291–305. ISSN: 1551-6709.

Patrick Sturt, Martin J. Pickering, and Matthew W. Crocker (Jan. 1999). "Structural Change and Reanalysis Difficulty in Language Comprehension". In: *Journal of Memory and Language* 40.1, pp. 136–150. ISSN: 0749-596X. URL: https://www.sciencedirect.com/science/article/pii/S0749596X98926067.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu (2023). *RoFormer: Enhanced Transformer with Rotary Position Embedding*. arXiv: 2104.09864 [cs.CL]. URL: https://arxiv.org/abs/2104.09864.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le (2014). *Sequence to Sequence Learning with Neural Networks*. arXiv: 1409.3215 [cs.CL]. URL: https://arxiv.org/abs/1409.3215.

Benjamin Swets, Timothy Desmet, Charles Clifron Jr., and Fernanda Ferreira (Jan. 2008). "Underspecification of syntactic ambiguities: Evidence from self-paced reading". English. In: *MEMORY & COGNITION* 36.1, pp. 201–216. ISSN: 0090-502X.

Michael K. Tanenhaus, Michael J. Spivey-Knowlton, Kathleen M. Eberhard, and Julie C. Sedivy (June 1995). “Integration of Visual and Linguistic Information in Spoken Language Comprehension”. In: *Science* 268.5217, pp. 1632–1634. ISSN: 1095-9203.

Wilson L. Taylor (1953). “"Cloze procedure": a new tool for measuring readability”. In: *Journalism Quarterly* 30, pp. 415–433. ISSN: 0196-3031.

Ian Tenney, Dipanjan Das, and Ellie Pavlick (July 2019). “BERT Rediscovers the Classical NLP Pipeline”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, pp. 4593–4601. URL: https://aclanthology.org/P19-1452/.

Lucien Tesnière (1959). *Eléments de Syntaxe Structurale*. Paris: Klincksieck.

Lucien Tesnière (Feb. 2015). *Elements of structural syntax*. Trans. by Timothy F. Osborne and Sylvain Kahane. John Benjamins Publishing Company. ISBN: 9789027212122. URL: https://benjamins.com/catalog/z.185.

The R Core Team (Oct. 31, 2025). *R: A Language and Environment forStatistical Computing*. R Foundation for Statistical Computing. URL: https://cran.r-project.org/doc/manuals/r-release/fullrefman.pdf.

William Timkey and Tal Linzen (Dec. 2023). “A Language Model with Limited Memory Capacity Captures Interference in Human Sentence Processing”. In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 8705–8720. URL: https://aclanthology.org/2023.findings-emnlp.582/.

William Timkey and Tal Linzen (Mar. 2025). *LM surprisal underpredicts garden path effects even with limited syntactic parallelism*. Abstract, 38th Annual Conference on Human Sentence Processing. University of Maryland, College Park, USA. URL: https://hsp2025.github.io/abstracts/171.pdf.

Mariya Toneva and Leila Wehbe (2019). “Interpreting and improving natural-language processing (in machines) with natural language-processing (in the brain)”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc. URL: https://papers.nips.cc/paper_files/paper/2019/hash/749a8e6c231831ef7756db230b4359c8-Abstract.html.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample (2023). *LLaMA: Open and Efficient Foundation Language Models*. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov (Nov. 2019). “Transformer Dissection: An Unified Understanding for Transformer’s Attention via the Lens of Kernel”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, pp. 4344–4353. URL: https://aclanthology.org/D19-1443.

John W. Tukey (1980). “We need both exploratory and confirmatory”. English. In: *American Statistician* 34.1, pp. 23–25. ISSN: 0003-1305.

Julie A. Van Dyke (2007). "Interference effects from grammatically unavailable constituents during sentence processing." In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 33.2, pp. 407–430. ISSN: 0278-7393.

Julie A. Van Dyke and Richard L. Lewis (Oct. 2003). "Distinguishing effects of structure and decay on attachment and repair: A cue-based parsing account of recovery from misanalyzed ambiguities". In: *Journal of Memory and Language* 49.3, pp. 285–316. ISSN: 0749-596X. URL: https://www.sciencedirect.com/science/article/pii/S0749596X03000810.

Julie A. Van Dyke and Brian McElree (2006). "Retrieval interference in sentence comprehension". In: *Journal of Memory and Language* 55.2, pp. 157–166. ISSN: 1096-0821.

Julie A. Van Dyke and Brian McElree (2011). "Cue-dependent interference in comprehension". In: *Journal of Memory and Language* 65.3, pp. 247–263. ISSN: 0749-596X. URL: https://www.sciencedirect.com/science/article/pii/S0749596X11000581.

Rpg Van Gompel and Mj Pickering (Aug. 2001). "Reanalysis in sentence processing: Evidence against current constraint-based and two-stage models". In: *Journal of Memory and Language* 45.2, pp. 225–258. ISSN: 0749-596X.

Shravan Vasishth and Richard L. Lewis (July 2004). "Modeling Sentence Processing in ACT-R". In: *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*. IncrementParsing '04. Barcelona, Spain: Association for Computational Linguistics, pp. 82–87. URL: https://aclanthology.org/W04-0313/.

Shravan Vasishth, Daniel Schad, Audrey Bürki, and Reinhold Kliegl (Aug. 14, 2022). *Draft: Linear Mixed Models in Linguistics and Psychology: A Comprehensive Introduction*. URL: https://vasishth.github.io/Freq_CogSci/ (visited on 01/26/2025).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

Geert Verbeke and Geert Molenberghs (2009). *Linear Mixed Models for Longitudinal Data*. Springer Series in Statistics. Springer New York. ISBN: 9781441902993.

Jesse Vig and Yonatan Belinkov (2019). *Analyzing the Structure of Attention in a Transformer Language Model*. arXiv: 1906.04284 [cs.CL]. URL: https://arxiv.org/abs/1906.04284.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov (July 2019). "Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, pp. 5797–5808. URL: https://aclanthology.org/P19-1580.

Nicholas J. Wade (Jan. 2010). "Pioneers of Eye Movement Research". In: *i-Perception* 1.2, pp. 33–68. ISSN: 2041-6695.

Matthew W. Wagers, Ellen F. Lau, and Colin Phillips (2009). "Agreement attraction in comprehension: Representations and processes". In: *Journal of Memory and Language* 61.2, pp. 206–237. ISSN: 0749-596X. URL: https://www.sciencedirect.com/science/article/pii/S0749596X09000448.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma (2020). *Linformer: Self-Attention with Linear Complexity*. arXiv: 2006.04768 [cs.LG]. URL: https://arxiv.org/abs/2006.04768.

Yau-Shian Wang, Hung-Yi Lee, and Yun-Nung Chen (2019). *Tree Transformer: Integrating Tree Structures into Self-Attention*. arXiv: 1909.06639 [cs.CL]. URL: https://arxiv.org/abs/1909.06639.

Tessa Warren and Edward Gibson (Aug. 2002). "The influence of referential processing on sentence complexity". In: *Cognition* 85.1, pp. 79–112. ISSN: 0010-0277.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Gotlieb Wilcox, and Chengxu Zhuang (2023). *Call for Papers – The BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus*. arXiv: 2301.11796 [cs.CL]. URL: https://arxiv.org/abs/2301.11796.

Larry Wasserman (2004). *All of Statistics*. en. Ed. by Larry Wasserman. New York, USA: Springer, pp. 149–173. ISBN: 9780387217369. URL: https://link.springer.com/book/10.1007/978-0-387-21736-9.

Ronald L. Wasserstein and Nicole A. Lazar (2016). "The ASA Statement on p-Values: Context, Process, and Purpose". In: *The American Statistician* 70.2, pp. 129–133. eprint: https://doi.org/10.1080/00031305.2016.1154108. URL: https://doi.org/10.1080/00031305.2016.1154108.

Shuhei Watanabe (2023). *Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance*. arXiv: 2304.11127 [cs.LG]. URL: https://arxiv.org/abs/2304.11127.

Bonnie Lynn Webber (June 1988). "Tense as discourse anaphor". In: *Computational Linguistics* 14.2, pp. 61–73. ISSN: 0891-2017.

William Wen (Mar. 15, 2023). *Introduction to torch.compile — PyTorch Tutorials 2.12.0+cu130 documentation*. en. URL: https://docs.pytorch.org/tutorials/intermediate/torch_compile_tutorial.html (visited on 05/05/2026).

Jacob Whitehill (2013). *Understanding ACT-R - an Outsider's Perspective*. arXiv: 1306.0125 [cs.LG]. URL: https://arxiv.org/abs/1306.0125.

Ethan Gotlieb Wilcox (Dec. 17, 2025). *Going Beyond "Humanlike": Using LLMs to advance psycholinguistic theories*. Online talk given at ILFC seminar. URL: https://gdr-lift.loria.fr/monthy-online-ilfc-seminar/.

Ethan Gotlieb Wilcox, Jon Gauthier, Jennifer Hu, Peng Qian, and Roger P. Levy (2020). "On the Predictive Power of Neural Language Models for Human Real-Time Comprehension Behavior". In: *CoRR* abs/2006.01912. arXiv: 2006.01912. URL: https://arxiv.org/abs/2006.01912.

Ethan Gotlieb Wilcox, Tiago Pimentel, Clara Meister, Ryan Cotterell, and Roger P. Levy (2023). "Testing the Predictions of Surprisal Theory in 11 Languages". In: *Transactions of the Association for Computational Linguistics* 11, pp. 1451–1470. URL: https://aclanthology.org/2023.tacl-1.82/.

Ethan Gotlieb Wilcox, Tiago Pimentel, Clara Meister, Ryan Cotterell, and Roger P. Levy (2025). *Testing the Predictions of Surprisal Theory in 11 Languages*. arXiv: 2307.03667 [cs.CL]. URL: https://arxiv.org/abs/2307.03667.

Ethan Gotlieb Wilcox, Pranali Vani, and Roger Levy (Aug. 2021). "A Targeted Assessment of Incremental Processing in Neural Language Models and Humans". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Online: Association for Computational Linguistics, pp. 939–952. URL: https://aclanthology.org/2021.acl-long.76/.

Naoko Witzel, Jeffrey Witzel, and Kenneth Forster (Apr. 2012). "Comparisons of Online Reading Paradigms: Eye Tracking, Moving-Window, and Maze". English. In: *Journal of Psycholinguistic Research* 41.2, pp. 105–128. ISSN: 0090-6905.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush (2020). *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. arXiv: 1910.03771 [cs.CL]. URL: https://arxiv.org/abs/1910.03771.

Sascha Wolfer (Jan. 2015). "Reading corpora as an instrument for studying a relevance-based account of language processing. A case study using a reading corpus of German jurisdictional texts." In: *Proceedings of the 6th Conference on Quantitative Investigations in Theoretical Linguistics*.

WordNet (2010). *About WordNet*. Princeton University. URL: https://wordnet.princeton.edu (visited on 05/06/2026).

Stephen Wu, Asaf Bachrach, Carlos Cardenas, and William Schuler (July 2010). "Complexity Metrics in an Incremental Right-Corner Parser". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Ed. by Jan Hajič, Sandra Carberry, Stephen Clark, and Joakim Nivre. Uppsala, Sweden: Association for Computational Linguistics, pp. 1189–1198. URL: https://aclanthology.org/P10-1121/.

Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, and Christian Szegedy (2022). *Memorizing Transformers*. arXiv: 2203.08913 [cs.LG]. URL: https://arxiv.org/abs/2203.08913.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu (2020). "On Layer Normalization in the Transformer Architecture". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 10524–10533. URL: https://proceedings.mlr.press/v119/xiong20b.html.

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin (2019). *Understanding and Improving Layer Normalization*. arXiv: 1911.07013 [cs.LG]. URL: https://arxiv.org/abs/1911.07013.

Weijie Xu, Jason Chon, Tianran Liu, and Richard Futrell (Dec. 2023). "The Linearity of the Effect of Surprisal on Reading Times across Languages". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 15711–15721. URL: https://aclanthology.org/2023.findings-emnlp.1052/.

Hiroyuki Vincent Yamazaki (Aug. 2024). *"Multivariate" TPE Makes Optuna Even More Powerful*. en. URL: https://medium.com/optuna/multivariate-tpe-makes-optuna-even-more-powerful-63c4bfbaebe2 (visited on 09/01/2025).

Adam Yedidia (July 2023a). *GPT-2's positional embedding matrix is a helix — LessWrong*. URL: https://www.lesswrong.com/posts/qvWP3aBDBaqXvPNhS/gpt-2-s-positional-embedding-matrix-is-a-helix (visited on 02/19/2026).

Adam Yedidia (Aug. 2023b). *The positional embedding matrix and previous-token heads: how do they actually work? — LessWrong*. URL: https://www.lesswrong.com/posts/zRA8B2FJLtTYRgie6/the-positional-embedding-matrix-and-previous-token-heads-how (visited on 02/19/2026).

Daniel Yekutieli and Yoav Benjamini (1999). “Resampling-based false discovery rate controlling multiple test procedures for correlated test statistics”. In: *Journal of Statistical Planning and Inference* 82.1, pp. 171–196. ISSN: 0378-3758. URL: https://www.sciencedirect.com/science/article/pii/S0378375899000415.

Kaichao You, Runsheng Bai, Meng Cao, Jianmin Wang, Ion Stoica, and Mingsheng Long (2024). *depyf: Open the Opaque Box of PyTorch Compiler for Machine Learning Researchers*. arXiv: 2403.13839 [cs.LG]. URL: https://arxiv.org/abs/2403.13839.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar (2019). “Are Transformers universal approximators of sequence-to-sequence functions?” In: *ArXiv* abs/1912.10077. URL: https://api.semanticscholar.org/CorpusID:209444410.

Jinle Zeng, Min Li, Zhihua Wu, Jiaqi Liu, Yuang Liu, Dianhai Yu, and Yanjun Ma (2022). *Boosting Distributed Training Performance of the Unpadded BERT Model*. arXiv: 2208.08124 [cs.DC]. URL: https://arxiv.org/abs/2208.08124.

Biao Zhang and Rico Sennrich (2019). *Root Mean Square Layer Normalization*. arXiv: 1910.07467 [cs.LG]. URL: https://arxiv.org/abs/1910.07467.

David Zola (1984). “Redundancy and word perception during reading”. In: *Perception & Psychophysics* 36, pp. 277–284. URL: https://api.semanticscholar.org/CorpusID:7661189.