UNIVERSITÀ DEGLI STUDI DI TRIESTE

DIPARTIMENTO DI MATEMATICA E GEOSCIENZE

Corso di Laurea Magistrale in Data Science and Scientific Computing

Interpreting Neural Language Models for Linguistic Complexity Assessment

Tesi di Laurea Magistrale

Relatore
Prof. Davide Crepaldi

Correlatore
Dott. Felice Dell'Orletta

Candidato
Gabriele Sarti

Anno Accademico 2019 - 2020

Acknowledgements

The majority of the following work was carried out during the COVID-19 pandemic, an incredibly hard time for our global society as a whole. For this reason, I would like to begin by acknowledging my great privilege in being able to entirely devote my past year's efforts to complete this research work without having to worry about me and my family's health and sustenance.

This thesis would not have been possible without the support of many people, and especially without the help and guidance of my supervisors. Davide and Felice, I would like to thank you with all my heart for being incredibly supportive despite the adverse circumstances and always making me feel a valued part of your labs and your research activities.

I would also like to acknowledge the dedication of professors and fellow students at the master's degree in Data Science and Scientific Computing in creating an environment that is at the same time pleasantly familiar and incredibly stimulating. I could not have asked for a better company during those two years. A special mention to the friends of Cacaopoli for the amazing moments passed together, and to my AI Student Society colleagues for believing in my dream of creating an AI student community in Trieste, and for selflessly bringing it to life to the benefit of future cohorts of students in AI and Data Science.

On the research side, I would like to sincerely thank all the members of the ItaliaNLP Lab in Pisa, who welcomed me in their group for my internship in 2019, first introduced me to natural language processing research, and ultimately motivated me in pursuing a doctorate after the end of this master's degree. My thanks also go to Prof. Elizabeth Schotter for her excellent introductory course to eye-tracking practices in cognitive science that immensely helped me to develop fundamental intuitions about gaze movements during reading, and to Dr. Nora Hollenstein for her precious advice on using gaze metrics in NLP studies.

I cannot be more thankful for the support of my close friends, which made these difficult times bearable for me. A special thanks to Laura, Karen, Alice, and Mattia, with whom I felt close even when we were physically far, and to Vale, for being the best thing this pandemic has brought in my life.

In conclusion, I am truly grateful to my parents and my family for always conciliating hard work with kindness, supporting me at all times, and always making me strive for the best. I aspire to be like you one day.

Abstract

Lo studio della complessità linguistica è un ambito profondamente multidisciplinare, che spazia dallo studio dell'elaborazione cognitiva in lettori umani alla classificazione della complessità strutturale caratterizzante espressioni in linguaggio naturale. In tempi recenti, l'utilizzo di metodi computazionali per il trattamento e l'analisi del linguaggio ha prodotto importanti sviluppi nella comprensione di molteplici fenomeni associati alla complessità linguistica. In linea con lo stato dell'arte del settore, questa tesi presenta uno studio model-driven di molteplici fenomeni associati alla complessità linguistica. In primo luogo, vengono esplorate empiricamente le relazioni che sussistono tra varie metriche estrinseche di complessità – percezione di complessità linguistica, leggibilità, elaborazione cognitiva e prevedibilità – evidenziando similitudini e differenze da una prospettiva linguisticamente e cognitivamente motivata. In seguito, viene studiato come l'informazione alla base delle diverse metriche di complessità possa essere acquisita da modelli del linguaggio basati su reti neurali, a vari livelli di astrazione e granularità, applicando tecniche di interpretabilità derivate dalla letteratura sull'elaborazione del linguaggio naturale. In conclusione, viene valutata la capacità di vari modelli computazionali di complessità nel prevedere difficoltà di elaborazione cognitiva associate a costrutti sintattici atipici, quali le *garden-path sentences*. I risultati sperimentali di questo studio forniscono prove convergenti riguardo alle limitate capacità di astrazione e generalizzazione dei modelli di linguaggio neurali allo stato dell'arte per la previsione della complessità linguistica, e incoraggiano all'adozione di linee di ricerca che integrino informazione simbolica e interpretabile in questo settore. In un'ottica di riproducibilità, il codice utilizzato per gli esperimenti viene reso disponibile al seguente indirizzo: https://github.com/gsarti/interpreting-complexity

Table of Contents

List of Figures

List of Tables

List of Abbreviations

(A)RA	(Automatic) Readability Assessment
ALBERT	A Lite BERT, see Lan et al., 2020
BERT	Bidirectional Encoder Representations from Transformers, see Devlin et al., 2019
CLS	Initial embedding used for sentence-level tasks in BERT-like neural language models.
ET	Eye-tracking, used to denote eye-tracking training involving multiple metrics
FFD	First Fixation Duration
FPD	First Pass Duration, also known as Gaze Duration
FXC	Fixation Count
FXP	Fixation Probability
GECO	Ghent Eye-tracking Corpus, see Cop et al., 2017
LCA	Linguistic Complexity Assessment
MLM	Masked Language Model(ing), e.g. BERT and ALBERT
MV/RR	Main Verb / Reduced Relative garden-path ambiguity, see Section 1.4
(N)LM	(Neural) Language Model
NLP	Natural Language Processing
NP/Z	Noun Phrase / Zero Object garden-path ambiguity, see Section 1.4
OSE	OneStopEnglish Corpus, see Vajjala and Lučić, 2018
PC(P)	Perceived Complexity (Prediction), see Brunato, De Mattei, et al., 2018
PWCCA	Projection-Weighted Canonical Correlation Analysis, see Morcos et al., 2018
RSA	Representation Similarity Analysis, see Kriegeskorte et al., 2008
SVM	Support Vector Machine, see Vapnik, 1998
TFD	Total Fixation Duration, also known as Total Reading Time
TRD	Total Regression Duration
ZuCo	Zurich Cognitive Language Processing Corpus, see Hollenstein, Rotsztejn, et al., 2018

Introduction

The study of complexity in language production and comprehension is a multidisciplinary field encompassing approaches that range from the analysis of cognitive processing phenomena in human subjects to the classification of structural complexity in natural language utterances. Because of its inherently faceted nature, linguistic complexity still defies a univocal definition and depends heavily on the point of view adopted during experimental inquiries. In recent years, as a consequence of the astounding expansion in human technological capabilities, the scientific community witnessed a proliferation of studies leveraging computational methods to investigate different complexity perspectives and develop automatic systems for linguistic complexity assessment. The introduction of neural network models able to automatically learn hierarchical representations of language spurred new lines of research in the field of Natural Language Processing, with researchers aiming to reverse-engineer theoretical intuitions by interpreting results and learning mechanics of those models. Nowadays, deep computational models are routinely adopted to study and evaluate linguistic complexity in applicative settings such as readability assessment, simplification, and first/second language learning.

This thesis fits into this current line of research by pursuing a two-fold aim. On the one hand, it investigates the connection between multiple human-centric perspectives of linguistic complexity – perception of complexity, readability, cognitive processing, and predictability – highlighting similarities and differences between them from a linguistically and cognitively-motivated viewpoint. On the other hand, it studies how those perspectives are learned by deep learning models at various levels of granularity. This work's primary focus concerns the analysis of learned representations using multiple interpretability techniques derived from the natural language processing (NLP) literature and the study of abstraction and generalization capabilities of modern computational models of language. A model-driven approach is adopted throughout this study, following the intuition that learned representations can be leveraged as proxies of the informational content required to perform linguistic complexity assessment. The modeling of linguistic complexity is studied on multiple extensively-used corpora spanning three complexity-related tasks – *perceived complexity prediction, automatic readability assessment, and gaze metrics prediction* – and further validated on ad-hoc psycholinguistic test suites. To further validate the impact of structural factors for complexity assessment, neural network-based annotation pipelines are notably employed alongside neural language models as black-box feature extraction systems.

Chapter 1 marks the beginning of this work by introducing the reader to the multiple facets of linguistic complexity. It starts with a broad categorization of complexity measurements into a spectrum taking into account both the perspective of analysis (intrinsic or extrinsic) and the processing modalities (online or offline). Relevant intrinsic perspectives related to linguistic complexity are then briefly presented, focusing on the extraction and use of morphosyntactic structures in complexity studies and the use of information-theoretic surprisal from language models as a structural measure of complexity. The three extrinsic complexity tasks representing this study's focus and their respective corpora are introduced in detail, focusing on their differences both from a conceptual and a data collection perspective. The chapter ends with an introduction to *garden-path sentences*, peculiar syntactic constructs associated with cognitive processing difficulties, later employed in the experiments of Chapter 5.

Chapter 2 motivates the choice of NLMs as the critical component in our experimental analysis: their ability to encode both semantic and structural properties of language makes them especially suitable in the context of linguistic complexity modeling. After a summary of the ascent of NLMs in the field of NLP, the two neural language models used in experimental sections are presented in detail. To conclude, three interpretability approaches are used to leverage learned representations to study complexity learning across tasks, and abstraction layers are presented.

Chapter 3 is the first experimental section, in which perceived complexity annotations and eye-tracking metrics collected at sentence level are linked to various linguistic phenomena extracted by a linguistic parser. The same analysis is also performed by controlling sentence length to limit the disproportionate influence of length-related features on complexity measures. The predictive performances of NLMs are then evaluated on perceived complexity and various eye-tracking metrics for both length-controlled and unconditional settings. The chapter ends with probing task experiments highlighting how complexity-related linguistic properties become implicitly encoded in model representations after complexity learning, suggesting interesting perspectives in priming models with syntactic information to improve their performances on complexity-related tasks.

Chapter 4 builds upon previous chapters' intuitions to compare the contextual embeddings generated from a single corpus by multiple models trained on the different complexity-related tasks. First, a set of assumptions is formulated to guide the empirical evaluation of how models encode complexity properties after fine-tuning. Similarity scores are then computed layer-wise across language models using two interpretability approaches to evaluate whether the information shared across different complexity perspectives is encoded by models with different fine-tuning objectives. Finally, learned representations are compared across model layers and fine-tuning tasks to highlight whether and how fine-tuning objectives influence the abstraction hierarchy learned by language models.

Chapter 5 concludes the experimental portion of this work by studying the connection between eye-tracking metrics and language modeling surprisal and investigating whether gaze metrics fine-tuning can enable language models to individuate cognitive processing triggers like garden-path sentences. A data-driven strategy is first adopted to establish a conversion coefficient between surprisal units and reading times. This coefficient is then used to evaluate whether a model that correctly highlights increased cognitive processing in specific constructions can also predict the magnitude of such phenomena. Autoregressive and masked language models are fine-tuned on eye-tracking measurements and then leveraged in a zero-shot setting to evaluate their ability in replicating garden-path effects in a controlled setting. Finally, models' performances are evaluated on a set of psycholinguistic benchmarks using surprisal and gaze recordings predictions to estimate the presence and magnitude of garden-path effects.

While studies on natural language complexity usually adopt a cross-lingual perspective, either by performing typological comparisons across language families or studying the impact of interlingual contacts on complexity changes, this work focuses solely on analyzing complexity annotations produced by native speakers of English. The English language was selected due to the broad availability of open-source corpora and resources, and no other languages were included in the study to keep it as self-contained as possible. Readers should be aware that English is widely considered morphologically and inflectionally poor despite its ubiquity in language studies, even compared to its Indo-European siblings. It should thus be avoided to generalize the results of this thesis work to other language families and typologies.[1] Moreover, this study focuses on the written language paradigm, but the importance of phonological phenomena in spoken language in evaluating language complexity is acknowledged (McWhorter, 2001).

This thesis work should be regarded as a broad, high-level exploration of multiple linguistic complexity perspectives employing modern computational approaches. In this sense, both introductory and experimental chapters are not intended to be exhaustive in providing a complete overview of the discussed topics. Instead, they aim to provide the minimal context needed to interpret experimental results correctly. Introductory chapters include pointers to additional resources discussing linguistic complexity for curious readers, and future studies on these topics will likely encompass any other perspective that was not covered by the present work.

[1] See Ruder (2020) for the importance of multilingual studies in NLP.

Both simple and complex types of language of an indefinite number of varieties may be found spoken at any desired level of cultural advance. When it comes to linguistic form, Plato walks with the Macedonian swineherd, Confucius with the head-hunting savage of Assam.

— Edward Sapir (1921), *Language*

1 | Linguistic Complexity

Defining linguistic complexity in a univocal way is challenging, despite the subjective intuition that every individual may have about what should be deemed complex in written or spoken language. Indeed, if the faculty of language allows us to produce a possibly infinite set of sentences from a finite vocabulary, there are infinitely many ways in which a sentence may appear difficult to a reader's eyes. An accurate definition is still debated in research fields like cognitive science, psycholinguistics, and computational linguistics. Nonetheless, it is indisputable that the concept of natural language complexity is closely related to difficulties in knowledge acquisition. This property stands both for human language learners and for computational models learning the distributional behavior of words in a corpus.

This introductory chapter begins with a categorization of linguistic complexity annotations following taxonomical definitions found in the literature. Various complexity metrics are then introduced alongside corpora and resources that were used throughout this study. Finally, the focus will be put on garden-path sentences, peculiar syntactically-ambiguous constructs studied in the experiments of Chapter 5.

1.1 Categorizing Linguistic Complexity Measures

In modern literature about linguistic complexity, two positions, each trying to define the nature of linguistic complexity phenomena, can be identified. In Kusters (2008) words:

> On the one hand, complexity is used as a theory-internal concept, or linguistic tool, that refers only indirectly, by way of the theory, to language reality. On the other hand, complexity is defined as an empirical phenomenon, not part of, but to be explained by a theory.

These definitions are coherent with the **absolute** and **relative complexity** terminology coined by Miestamo (2004), where relative complexity is seen as a factor characterizing the perceptual experience of specific language users. In contrast, absolute complexity is structurally-defined by language constructs and independent from user evaluation. While these two perspectives

seem to identify two opposite viewpoints over linguistic complexity, the distinction between the two becomes blurred when we consider that linguistic theories underlying absolute complexity evaluation are developed by linguists, who still have a subjective perspective despite their competence (Kusters, 2003). Two definitions are now introduced to operationalize absolute and relative complexity in the context of complexity measurements:

Intrinsic Perspective The intrinsic perspective on linguistic complexity is closely related to the notion of absolute complexity. From the intrinsic viewpoint, language productions are evaluated using their distributional and structural properties, without any complexity annotation derived by language users. The linguistic system is characterized by a set of elementary components (lexicon, morphology, syntax *inter alia*) that interact hierarchically (Cangelosi et al., 2002), and their interactions can be measured in terms of complexity by fixing a set of rules and descriptions. The focus is on objectivity and automatic evaluation based on the intrinsic properties of language systems.

Extrinsic Perspective The extrinsic perspective connects to the concept of relative complexity and takes into account the individual perspective of users. Complexity judgments are collected during or after the processing of linguistic productions and are then evaluated in terms of cognitive effort required by language users for comprehension. The extrinsic viewpoint is partaken by cognitive processing theories in psycholinguistics such as the Dependency Locality Theory (Gibson, 1998; Gibson, 2000), the Surprisal Theory (Hale, 2001; Hale, 2016; Levy, 2008), and the more recent Lossy-context Surprisal Theory (Futrell, Gibson, et al., 2020), aiming to disentangle the source of processing difficulties in sentence comprehension. The focus, in this case, is on the subjectivity of language users and their judgments.

Despite being different under many aspects, the two perspectives are highly interdependent: a user's perception of complexity will be strongly influenced by the distributional and structural properties of utterances, and some of those properties will be considered complex in relation to the type of judgments they typically elicit in language users. Provided that the strength of human influence in complexity measurements can vary widely depending on data collection procedures, the two perspectives can be seen as the two ends of a spectrum. A visual representation is provided by the horizontal axis of the complexity measures compass in Figure 1.1.

An additional dimension for categorizing linguistic complexity metrics can be introduced by considering the time at which measures are obtained, relative to the incremental processing paradigm that characterizes natural reading in human subjects. In this context, *processing* is defined as any act aimed at extracting information from linguistic forms and structures, either by employing reasoning (in humans) or through computation (in automatic systems).

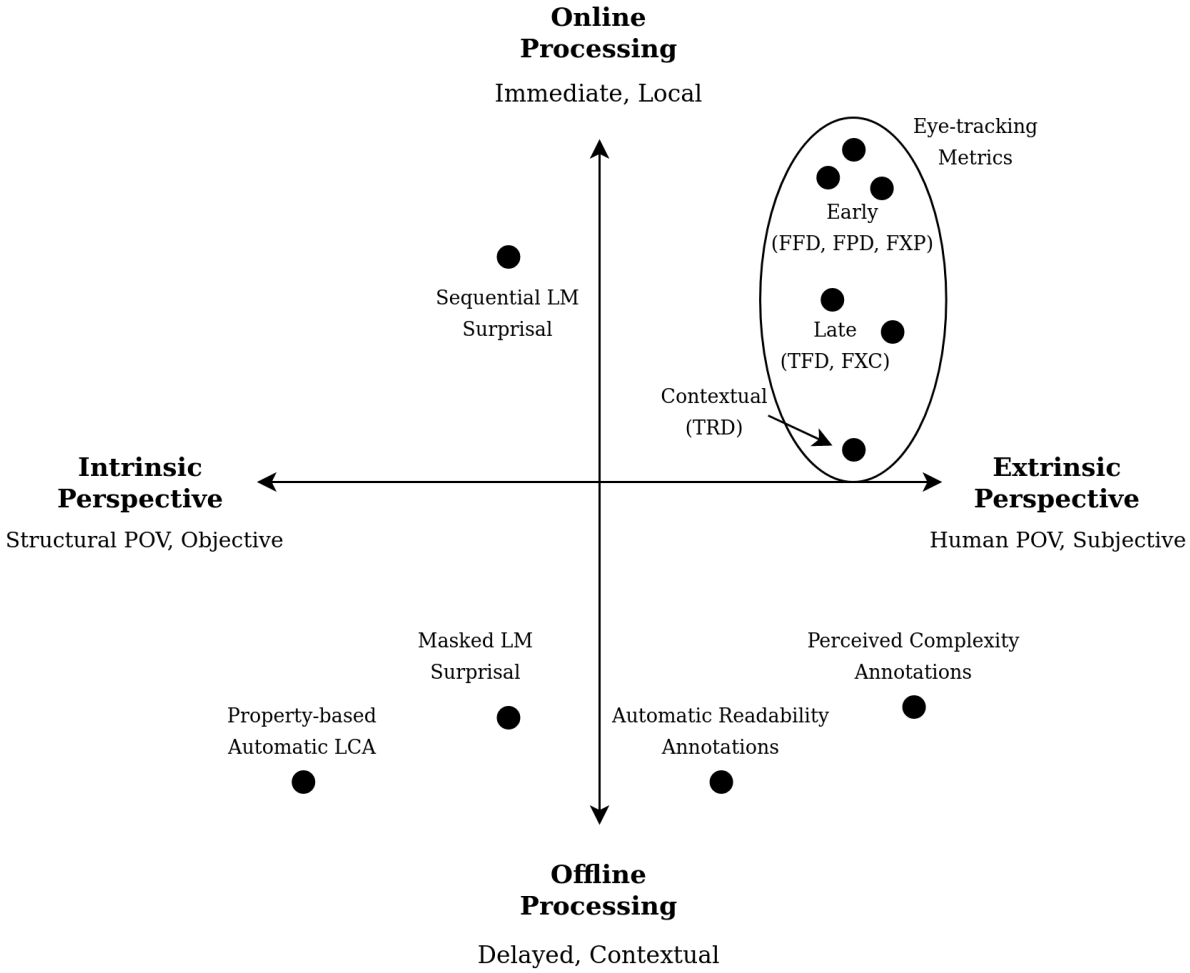


Figure 1.1: Complexity measures' compass.

Again, we can identify the two ends of a spectrum concerning processing modalities, related to the concepts of **local** and **global complexity** found in linguistic literature (Edmonds, 1999; Miestamo, 2004; Miestamo, 2008):

Online processing Online complexity judgments are collected while a language user, be it a human subject or a computational system, is sequentially processing a text. Online processing is widely explored in the cognitive science literature, where behavioral metrics such are fMRI data and gaze recordings are collected from subjects exposed to locally and temporally-immediate inputs and tasks that require fast processing (Iverson et al., 1999). The act of reading is predominantly performed by online cognition (Meyer et al., 1992), making online measures especially suitable for complexity evaluation for natural reading.

Offline processing Offline complexity judgments are collected at a later time when the language user has a complete and contextual view of the text in its entirety. Again, offline complexity

is related to the offline cognition paradigm (Day, 2004) typically used in re-evaluations and future planning. In practice, offline evaluation accounts for contextual and cultural factors closely related to individual subjectivity and is poorly captured by immediate online metrics.

Figure 1.1 situates various linguistic complexity metrics in terms of processing modalities and analyzed perspective by including the processing spectrum on the vertical axis. In the next sections, all these measures will be introduced and their use will be motivated in light of this categorization.

1.2 Intrinsic Perspective

Complexity studies where the intrinsic point of view is adopted rely on annotations describing linguistic phenomena and structures in sentences and aim to map those to complexity levels or ratings, often resorting to formulas parametrized through empirical observation. Given the scarcity of experienced human annotators and the cost of a manual annotation process, computational systems have been primarily employed to extract linguistic information from raw text in an automated yet precise way.

Another intrinsic viewpoint is based on the intuition that frequent constructs should be deemed as less complex than infrequent ones. In this case, terms' co-occurrences are extracted from large corpora, and complexity judgments are derived from their probabilistic likelihood of appearance in a given context. Given the infeasibility of tracking co-occurrences for long sequences in large, typologically-varied corpora, **computational language models** are usually employed to learn approximations of co-occurrence likelihoods for specific constructs.

While this thesis work only partially addresses the use of these approaches, they will be briefly introduced to provide additional context for understanding extrinsic perspectives and their experimental evaluation.

1.2.1 Structural Linguistic Complexity

Language systems can be seen as hierarchies of rules and processes governing various aspects of utterances production and use. For each of those levels, it is possible to identify characteristics leading to higher complexity from a structural standpoint (Sinnemäki, 2011):

- A greater number of parts in a specific language level leads to a greater **syntagmatic complexity** (also known as *constitutional complexity*). This mode is related to the *lexical* and "superficial" properties of language, such as the length of words and sentences.

- A greater variety of parts in a specific language level leads to a greater **paradigmatic complexity** (also known as *taxonomic complexity*). This mode characterizes, in particular, the *phonological* level, where the presence of an elaborated tonal system makes a language more complex (McWhorter, 2001), the *morphologic* level, where inflectional morphology is usually associated to a higher degree of complexity (McWhorter, 2001; Kusters, 2003) when compared to the regularity of derivational rules, and the *semantic* level, where polysemic words are generally considered more complex than monosemic ones (Voghera, 2001).

- A greater variety of interrelation modalities and hierarchical structures leads to greater **organizational and hierarchical complexities**. Those complexity modes are mainly related to the *syntactic level*, where recursive and nested constructs are deemed more complex and possibly determinant in distinguishing human language from animal communication (Hauser et al., 2002).

Focusing on the syntactic level, we can find multiple factors accounting for greater complexity (Berruto et al., 2011):

- Subordinate clauses preceding the main clause, as in *If you need help, let me know"* as opposed to *"Let me know if you need help"*.

- Presence of long-range syntactic dependencies between non-contiguous elements, as in *"The dog that the cat chased for days ran away"* where the subject referent (*dog*) and its verb (*ran*) are far apart in the sentence.

- A high degree of nesting between elements and substructures, as in *"The mouse **that the cat** that the dog bit **ate** was bought at the fair"* where two nested subordinate clauses introduced by the preposition *that* are present.

- Repeated applications of recursive principles to build utterances with different meanings through the compositionality principle, as in *"I am a huge fan of fans of fans of . . . of recursion"*, where the number of recursions defines the final meaning of the sentence.

While all those properties are relevant when evaluating an utterance's complexity, only some can be easily extracted from corpora using automatic approaches. In the specific context of this work, the analysis of complexity-related features in Chapter 3 makes use of the Profiling–UD tool[1] (Brunato, Cimino, et al., 2020), implementing a two-stage process: first, the linguistic annotation process is automatically performed by UDPipe (Straka et al., 2016), a multilingual pipeline

[1] Available at `http://linguistic-profiling.italianlp.it`

leveraging neural parsers and taggers included in the Universal Dependencies initiative (Nivre et al., 2016). During this step, sentences are tokenized, lemmatized, POS-tagged (i.e., words are assigned lexical categories such as "Noun" and "Verb") and parsed (i.e., the hierarchical structure of syntactic dependencies is inferred). Then, a set of about 130 linguistic features representing underlying linguistic properties of sentences is extracted from various levels of annotation. Those features account for multiple morphological, syntactic, and "superficial" properties related to linguistic complexity. A relevant subset of those features is presented in detail in Appendix A.

After deriving linguistic properties from sentences, either automatically as in this study or by manual annotations, two approaches are viable to determine their complexity while maintaining an intrinsic perspective (no human processing data involved):

Formula-based Approach This approach treats linguistic properties of input texts as components of a formula used to determine levels or readability grades. Traditional readability formulas consider multiple factors, such as word length, sentence length, and word frequency. Parameters in those formulas are carefully hand-tuned to match human intuition and correlate well with human-graded readability levels.[2]

Learning-based Approach This approach casts the complexity prediction problem in the supervised machine learning framework. More specifically, linguistic parsers are used to predict linguistic properties, and their accuracy on a set of gold-labeled instances is taken as an indicator of complexity. In the case of dependency parsers (i.e., models trained to extract the syntactic structure of a sentence), two evaluation metrics can be used: the *Unlabeled and Labeled Attachment Scores* (UAS and LAS), where the UAS is the percentage of words assigned to the right dependency head and LAS also consider if the dependency relation was labeled correctly.

Both approaches are represented in Figure 1.1 under the label "Property-based Automatic LCA" and are considered offline since the text is generally not processed incrementally but instead taken as a whole.

1.2.2 Language Modeling Surprisal

The information-theoretic concept of **surprisal**, also known as *information content* of an event, can be seen as a quantification of the level of surprise caused by a specific outcome: an event that is certain yields no information, while the less probable an event is, the more surprising it gets. Formally, an event x with probability $p(x)$ has a surprisal value equal to:

$$I(x) = -\log[p(x)] \tag{1.1}$$

[2]This motivates the previous claim about the interdependence of intrinsic and extrinsic approaches. See Section 2.1 of Martinc et al. (2019) for an overview of the most popular metrics for English.

The idea that probabilistic expectations in the context of language reading are related to greater complexity in terms of cognitive processing was formalized by *surprisal theory* (Hale, 2001; Hale, 2016). Surprisal theory defines processing difficulties D (which can be considered as proxies of complexity) as directly proportional to the surprisal produced in readers by a word w given its previous context c (i.e., preceding words in the sentence):

$$D(w_i|c) \propto -\log p(w_i|c) = -\log p(w_i|w_i-1, w_i-2, \ldots, w_0) \tag{1.2}$$

While processing difficulties imply human subjects' presence, **language models** (LM) can be used to estimate the conceptually similar information-theoretic surprisal without the need of human annotations by learning word occurrences and co-occurrences probabilities from large quantities of text. Concretely, a language model is a probabilistic classifier that learns to predict a probability distribution over words of a vocabulary V given a large number of contexts c in which those words occur (Goodman, 2001):

$$p(w_i|c) \quad \forall\, w_i \in V \tag{1.3}$$

After the training procedure it is possible to estimate the probability $p(s)$ of a sentence s having length m as the product of the conditional probabilities assigned to individual words by the language model, given its context:

$$p(s) = p(w_1, \ldots, w_m) = \prod_{i=1}^{m} p(w_i\,|c) \tag{1.4}$$

We can consider the surprisal $I(s) = -\log p(s)$ as an *intrinsic measure* of linguistic complexity since it is a function of the co-occurrence relations derived by the training corpora. Thus, it describes how likely a construct can be observed in a structurally-sound manner, without relying on human processing data. However, automatic surprisal estimation using language models cannot be considered purely intrinsic since it is highly dependent on a multitude of factors that are arguably "less objective" than the linguistic categories of the previous section, such as the type and dimension of the considered context and the corpora employed by the LM to learn words' distributional behavior.

We can categorize modern language models in two broad categories: **sequential** models (also known as *autoregressive* or *causal* LMs) consider as context only preceding words, while **bidirectional** models (also known as *masked* LMs) consider both preceding and following words

when estimating occurrence probabilities, much like the well-established *cloze test* (Taylor, 1953) in psycholinguistics. Equations (1.5) show how the sentence surprisal equation (1.4) is adapted in both cases, using the product rule for logarithms:

$$\begin{aligned} I_{\text{sequential}}(s) &= -\sum_{i=1}^{m} \log p(w_i \,|w_1, w_2, \ldots, w_{i-1}) \\ I_{\text{bidirectional}}(s) &= -\sum_{i=1}^{m} \log p(w_i \,|w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_m) \end{aligned} \tag{1.5}$$

If the LM used to estimate surprisal was sequential, then surprisal estimation could be considered part of the *online processing paradigm* despite the absence of a human subject.[3] In the bidirectional case, the estimation of surprisals from the whole context can be assimilated with offline processing practices.

The relation between co-occurrence frequencies estimated by a language model and perception of complexity is one of the aspects that make language models especially suitable for predicting extrinsic complexity metrics, as it will be discussed in Chapter 2.

1.3 Extrinsic Perspective

Extrinsic complexity measures elicited from human-produced signals and annotations are the main focus of this thesis work. In this section, three different viewpoints on linguistic complexity assessment from a human perspective are introduced:

- The **readability** point-of-view, as intended in the context of the *automatic readability assessment* (ARA) task, is concerned with collocating similar textual inputs into difficulty levels that are often predetermined by writers and given a clear semantic interpretation (e.g., easy, medium, hard).

- The **perceptual** point-of-view, represented by the *perceived complexity prediction* (PCP) task, is based on human annotations of complexity on a numeric scale, taking into account disparate textual inputs presented sequentially to obtain more generalizable complexity annotations. Unlike ARA, PCP annotations are produced by readers after sentence comprehension.

[3] This is an admittedly simplistic reduction, given the importance of parafoveal processing in reading (Schotter et al., 2012; Schotter, 2018)

- The **cognitive** point-of-view, employing cognitive signals collected by specialized machinery (e.g., electrodes, MRI scanners, eye-trackers) as proxies for the linguistic complexity experienced by users. In this work, the focus will be on the *gaze metrics prediction* task, using gaze data collected from subjects during natural reading.

All three complexity-related tasks will be introduced alongside recent results in the literature. The corpora on which each task relies upon will also be presented in their respective sections.

1.3.1 Automatic Readability Assessment

While the term *readability assessment* is often broadly employed to denote the task of predicting the general reading difficulty of a text, here it is used to describe the typical approach in ARA, relying on corpora categorized by the writer's perception of what is difficult for readers.

We can take as an example the OneStopEnglish (OSE) corpus (Vajjala and Lučić, 2018), which will be used later to study the ARA relation with other complexity tasks in Chapter 4. OSE contains 567 weekly articles from The Guardian newspaper rewritten by language teachers to suit three adult English learners' levels. Each text can be divided into passages spanning one or multiple sentences, each labeled with a readability level ("Elementary", "Intermediate" or "Advanced") based on the original writers' judgment. An example of the same passage at different reading levels is provided in Table 1.1.

From Table 1.1 example, it is evident that the reading level of a specific text should be interpreted only in relation to its other versions, i.e., elementary passages are not necessarily straightforward in absolute terms, but rather *less complicated than their intermediate and advanced counterparts*. This affirmation holds for the OSE corpus and other widely-used readability corpora such as the Newsela corpus (Xu et al., 2015), which contains newspaper articles rewritten by experts to match eleven school grade reading levels. For this reason, and because of its writer-centric perspective relying only on readability judgments formulated by the same writers who composed the passages, readability assessment is fundamentally different from the other extrinsic approaches.[4] ARA can be framed as a machine learning task in which a computational model m is trained to predict the readability level $y \in \mathcal{Y}$ over a set of labeled examples $\mathcal{S} = (s_1, s_2, \ldots, s_n)$ in two possible ways:

- A simple multiclass classification setting, where the model predicts the level of a single sentence s. In this case, the model outputs a prediction $m(s) = \hat{y} \in \mathcal{Y}$. We can then minimize the categorical cross-entropy $H(y, \hat{y})$ between gold and predicted labels during the training process and evaluate the model's performances with standard classification

[4] See Collins-Thompson (2014) for a thorough review of ARA approaches.

Table 1.1: An OSE Corpus passage at different reading levels.

Reading Level	Example
Advanced (Adv)	Amsterdam still looks liberal to tourists, who were recently assured by the Labour Mayor that the city's marijuana-selling coffee shops would stay open despite a new national law tackling drug tourism. But the Dutch capital may lose its reputation for tolerance over plans to dispatch nuisance neighbours to scum villages made from shipping containers.
Intermediate (Int)	To tourists, Amsterdam still seems very liberal. Recently the city's Mayor assured them that the city's marijuana-selling coffee shops would stay open despite a new national law to prevent drug tourism. But the Dutch capitals plans to send nuisance neighbours to scum villages made from shipping containers may damage its reputation for tolerance.
Elementary (Ele)	To tourists, Amsterdam still seems very liberal. Recently the city's Mayor told them that the coffee shops that sell marijuana would stay open, although there is a new national law to stop drug tourism. But the Dutch capital has a plan to send antisocial neighbours to scum villages made from shipping containers, and so maybe now people wont think it is a liberal city any more.

metrics such as precision and recall. This approach is similar to the ones used for other extrinsic metrics but does not account for readability levels' relative nature.

- A multiple-choice scenario, where the model is provided with two semantically equivalent sentences s_1, s_2 at different readability levels ($s_1 \equiv s_2, y_1 \neq y_2$) and needs to predict which of the sentences has the highest readability level. In this case, which is more coherent with the relative nature of readability judgments, the model is trained to minimize the binary cross-entropy between gold and predicted labels $y, \hat{y} \in \mathcal{Y}_{bin} = \{0, 1\}$ corresponding to the position of the more complex sentence in the pair.

Expert annotations' effectiveness in determining readers' comprehension was recently questioned, as automatic readability scoring did not show a significant correlation to comprehension scores of participants, at least for the OSE Corpus (Vajjala and Lucic, 2019). However, measuring if this observation holds for other corpora and extrinsic approaches is beyond this thesis's scope.

1.3.2 Perceived Complexity Prediction

While ARA measures linguistic complexity in a context-relative and writer-centric sense, the *perceived complexity prediction* (PCP) approach focuses on eliciting absolute complexity judgments directly from target readers, aiming at evaluating difficulties in comprehension rather

Table 1.2: Sample of sentences taken from the English portion of the Perceived Complexity (PC) Corpus with complexity scores from crowdsourced annotators.

Sentence	**A1**	**A2**	**A3**	**...**	**A20**
In other European markets, share prices closed sharply higher in Frankfurt and Zurich and posted moderate rises in Stockholm, Amsterdam and Milan.	4	6	7	...	1
The pound strengthened to $ 1.5795 from $ 1.5765.	2	1	2	...	1
In Connecticut, however, most state judges are appointed by the governor and approved by the state legislature.	1	3	3	...	5
When the market stabilized, he added, the firm sold the bonds and quickly paid the loans back.	2	3	3	...	3
Paribas already holds about 18.7 % of Navigation Mixte, and the acquisition of the additional 48 % would cost it about 11 billion francs under its current bid.	5	2	3	...	6

than production. This approach was pioneered by Brunato, De Mattei, et al. (2018), who collected crowdsourced complexity ratings from native speakers for Italian and English sentences and evaluated how different structural linguistic properties contribute to human complexity perception. The use of annotators recruited on a crowdsourcing platform was intended to better grasp the layman's perspective on linguistic complexity, as opposed to ARA expert writers. If collected properly, crowdsourced annotations were shown to be highly reliable for linguistics and computational linguistics research by the survey of Munro et al. (2010).

Brunato, De Mattei, et al. (2018) extracted 1200 sentences from both the newspaper sections of the Italian Universal Dependency Treebank (IUDT) (Simi et al., 2014) and the Penn Treebank (McDonald et al., 2013), such that those are equally distributed in term of length. To collect human complexity judgments, twenty native speakers were recruited for each language on a crowdsourcing platform. Annotators had to rate each sentence's difficulty on a Likert 7-point scale, with 1 meaning "very simple" and 7 "very complex". Sentences were randomly shuffled and presented in groups of five per web page, with annotators being given a minimum of ten seconds to complete each page to prevent skimming. The quality of annotations was measured using the Krippendorff alpha reliability, obtaining 26% and 24% for Italian and English. Table 1.2 presents an example of English sentences labeled with multiple annotators' perceived complexity judgments.

As can be expected, PC judgments show significant variability across participants since they cannot be easily framed in a relative setting. Since this work's focus is related to a general

notion of complexity, PC judgments are averaged and filtered to obtain a score reflecting the mean perception of complexity of all participants in experimental chapters. The averaged score is later treated as the gold label in a regression task, with machine learning models trained to minimize the *mean square error* between their predictions and gold average annotations. Another possibility, which is not explored in this thesis work, would be to consider only single participants' judgments to model their linguistic complexity perception.

1.3.3 Gaze Metrics Prediction

Gaze data collected from human subjects during reading can provide us with useful insights from an online extrinsic complexity perspective. Patterns found in both *saccades*, i.e., eye movements from one location to another, and *fixations*, where eyes are relatively stable while fixating a specific region, were shown to be reliably linked to a multitude of linguistic factors (Demberg et al., 2008). Because of this, a linking assumption between overt attention and mental processing can be reasonably established, and gaze metrics can be considered as proxies of cognitive effort, and thus of complexity, at various processing levels.[5]

Gaze metrics are widely employed in cognitive processing research because of their multiple benefits: optical eye-tracking systems are non-invasive and relatively inexpensive compared to other approaches that directly measure brain activity, such as electroencephalography (EEG) and all magnetic resonance imaging (MRI) variants. Moreover, gaze data generally have high spatial and temporal precision, limited only by sampling rates, which are generally in the order of few milliseconds. This aspect is crucial for reading research since it allows us to directly associate gaze measures to specific *areas of interest* (AOI, also called region), i.e., small portions of the visual input provided to participants.

Gaze data for NLP Eye-tracking data and other cognitive signals were effectively used in many NLP applications such as POS tagging (Barrett, Bingel, Keller, et al., 2016), sentiment analysis (Mishra et al., 2017), native language identification (Berzak et al., 2018), and dependency parsing (Strzyz et al., 2019) *inter alia*, often providing modest yet consistent improvements across models and tasks through the combination of gaze features and linguistic features or distributed representations.[6] In the context of linguistic complexity assessment, eye-tracking data were applied to the ARA task for both monolingual and bilingual participants, obtaining meaningful results for sentence-level classification in easy and hard-to-read categories (Vasishth et al., 2013; Ambati et al., 2016). For example, Singh et al. (2016) first use a set of linguistic features to learn a reading times model from a set of gaze-annotated sentences and then use models' predicted times over a second set of sentences to perform multiple-choice ARA. González-Garduño et al.

[5] See Rayner (1998) for a comprehensive survey on findings related to eye-tracking research.
[6] See Hollenstein, Barrett, et al. (2020) for an exhaustive overview of current approaches and best practices.

(2018) extend this approach in a multitask learning setting (Caruana, 1997; Ruder, 2017), using eye-movement prediction tasks to produce models able to predict readability levels both from a native speaker and foreign language learner perspective.

Collecting Eye-tracking Data A typical procedure to collect gaze data for reading research, as described by Schotter (2020), usually includes the following steps:

- Textual inputs are selected and split by experiment designers, first in areas of interest directly mapped to pixels (for natural reading, usually word boundaries), then over multiple rows, and finally in screens presented to participants. This step should take into account calibration errors to determine the correct level of tolerance for off-word fixations.
- A participant is placed in a room with a display computer used to present visual inputs and a host computer used to record data from the eye-tracker setup. Optical eye-trackers use infrared light beams, which are reflected differently by different parts of the eye, to measure pupil and corneal reflection and track gaze movements at each timestep. The setup is calibrated and validated for each participant to ensure the quality of results.
- Each participant follows the on-screen instructions to complete a reading task trial while remaining at a fixed distance from the screen. A *fixation report* containing events (saccades, fixations, blinks) is produced for each individual on the host computer.
- Finally, a data preprocessing step is taken for each trial to identify and remove artifacts and possibly decide to reject the trial. Some examples of standard practices are the merge of fixations below 80ms due to eye jittering, the exclusion of fixations caused by track loss after blinks, and vertical drift correction (Carr et al., 2020). An *AOI report* containing gaze metrics grouped at AOI level can be produced.

Eye-tracking Metrics Metrics derived from the AOI report contain information about the processing phases in which subjects incur during sentence comprehension. *Early gaze measures* capture information about lexical access and early processing of syntactic structures, while *late measures* are more likely to reflect comprehension and both syntactic and semantic disambiguation (Demberg et al., 2008). The third kind of measures, referred to as *contextual* following the categorization in Hollenstein and Zhang (2019), capture information from surrounding content. Table 1.3 presents a subset of metrics, spanning the three categories, that will be used in the experimental section.[7] These metrics represent a minimal group spanning various stages of the reading process and are leveraged to study differences between online and offline processing among extrinsic metrics. In the experimental part, gaze scores are often averaged

[7] Appendix B contains information about deriving metric values for all corpora.

Table 1.3: Eye-tracking metrics used in this study.

Type	Metric Name	Description
Early	First Fixation Duration (FFD)	Duration of the first fixation over the region, including single fixations.
	First Pass Duration (FPD)	Duration of the first pass over a region.
	Fixation Probability (FXP)	Boolean value reflecting if the region was fixated or skipped during the first pass.
Late	Fixation Count (FXC)	Number of total fixations over a region.
	Total Fixation Duration (TFD)	Sum of all fixation durations over a region.
Contextual	Total Regression Duration (TRD)	Duration of regressive saccades performed after a region's first access and before going past it.

across participants to reduce noise in measurements and obtain a single label for each metric that can later be used as a reference in a regression setting. The average fixation probability across participants for each AOI is a value comprised in the range $[0, 1]$ and represents the proportion of subjects that accessed the region during their first gaze pass.

Eye-tracking Corpora The experimental part of this thesis work leverages four widely used eye-tracking resources: the Dundee corpus (Kennedy et al., 2003), the GECO corpus (Cop et al., 2017), the ZuCo corpus (Hollenstein, Rotsztejn, et al., 2018), and ZuCo 2.0 (Hollenstein, Troendle, et al., 2020). There are multiple reasons behind the choice of using multiple gaze-annotated corpora for this study. First, those corpora span different domains and provide us with a better intuition of what structures are perceived as complex in different settings and by different pools of subjects. Secondly, neural-network-based complexity models used in this work greatly benefit from a broader availability of annotated data to achieve higher performances in predicting eye-tracking metrics. Finally, while all corpora relied on different procedures and instrumentation, they are all derived from very similar experimental settings (i.e., natural reading on multiple lines), and can be easily merged after an individual normalization procedure (Hollenstein and Zhang, 2019). Table 1.4 presents some descriptive statistics of the four corpora.

- The **Dundee Corpus** developed by Kennedy et al. (2003) contains gaze data for ten native English speakers tasked with reading twenty newspaper articles from *The Independent*. The English section of the Dundee corpus includes 51,240 tokens in 2368 sentences. Texts were presented to subjects on a screen five lines at a time and recorded using a *Dr. Bois Oculometer Eyetracker* with 1 kHz monocular (right) sampling. Dundee corpus data are

Table 1.4: Descriptive statistics of eye-tracking corpora.

	Dundee	**GECO**	**ZuCo**	**ZuCo 2.0**	**Total**
domain(s)	news	literature	movie reviews, Wiki articles	Wiki articles	-
# of sentences	2368	5387	700	349	8804
mean sent. length	21.64	10.47	19.47	19.51	17.77
# of tokens	51240	56409	13630	6810	128089
unique token types	9928	6155	4650	2521	16320
mean token length	4.88	4.6	5.05	5.01	4.89
mean fix. duration	200	210	117	117	161
mean gaze duration	280	234	139	134	197

the oldest among selected corpora and have been extensively used in psycholinguistic research about naturalistic reading.

- The **Ghent Eye-tracking Corpus** (GECO) by Cop et al. (2017) was created more recently to study eye movements of both monolingual and bilingual subjects during naturalistic reading of the novel *The Mysterious Affair at Styles* by Agatha Christie (2003). In the context of this work, only the monolingual portion collected from 14 native English speakers is used, comprising 56,409 tokens in 5,387 sentences. Eye movements were recorded with an *EyeLink 1000* system with 1 kHz binocular sampling (only right eye movements were considered), and the text was presented one paragraph at a time.

- The **Zurich Cognitive Language Processing Corpus** (ZuCo) by Hollenstein, Rotsztejn, et al. (2018) is a dataset including both eye-tracking and EEG measurements collected simultaneously during both natural and task-oriented reading. The corpus contains 1100 English sentences from the Stanford Sentiment Treebank (Socher et al., 2013) and the Wikipedia dump used in Culotta et al. (2006) with gaze data for 12 adult native speakers. Only the first two portions are used for the present work since they contain natural reading data, totalizing 700 sentences and 13,630 tokens. The text was presented on-screen one sentence at a time, and data were collected with an *EyeLink 1000* as for GECO.

- **ZuCo 2.0** is an extension of ZuCo, including 739 sentences extracted from the Wikipedia corpus by Culotta et al. (2006). Only the 349 sentences for which natural reading data were collected are used, and the 100 duplicates shared with ZuCo to evaluate differences in setup and participants are removed. Data were collected from 18 native English speakers using an *EyeLink 1000 Plus* with 500 kHz sampling.

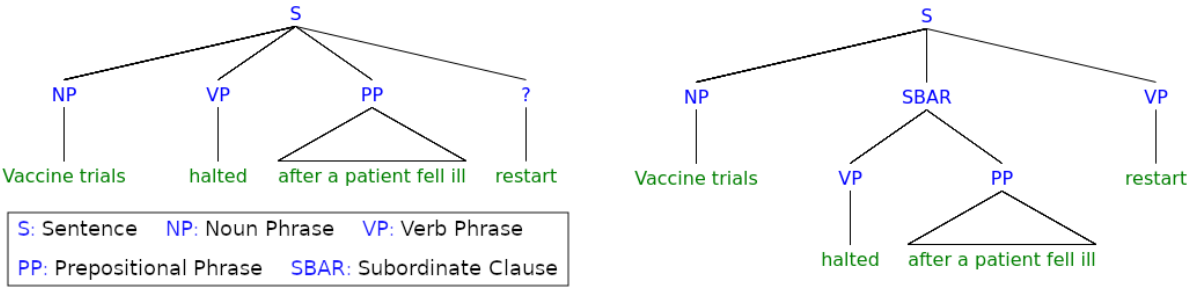


Figure 1.2: Syntax trees for the initial and complete parse of garden-path example (1).

Tokens are obtained using whitespace tokenization, which is the same approach used to perform gaze annotations across all eye-tracking corpora. Mean sentence length is expressed in number of tokens, and the number of unique types is computed as the size of the vocabulary after removing punctuation from all tokens. Approximately 128,000 tokens annotated with gaze recordings from multiple participants were used in the experiments of Chapters 4 and 5, while only GECO was used for the analysis of Chapter 3. Similarly to the PCP task, scores were averaged across subjects to reduce noise and obtain general estimates: in particular, reading times that were missing due to skipping were considered as having the lowest duration across annotators, which is a practice commonly used in literature. Again, considering individual participants' scores is deemed attractive in a personalization perspective but far beyond this work's scope.

1.4 Garden-path Sentences

Garden-path sentences, named from the expression "leading down the garden path" implying deception, are grammatically correct sentences that create a momentarily ambiguous interpretation in readers. The initial interpretation is later falsified by words encountered during sequential reading, becoming a significant source of processing difficulties. For this reason, garden-path constructions are used to evaluate models of linguistic complexity in the experiments of Chapter 5. Consider the following recent headline by the newspaper *The Guardian*:[8]

(1) Vaccine trials halted after patient fell ill restart.

Readers exposed to (1) tend to initially prefer the interpretation in which halted acts as the main verb of the sentence in simple past, i.e., *"Vaccine trials halted after patient fell ill."* is interpreted as a well-formed and semantically meaningful sentence. When the verb *restart* is reached, it suddenly becomes evident that the original parse would lead to an ungrammatical sentence, and a reanalysis requiring nontrivial cognitive processing is triggered. In conclusion,

[8] https://twitter.com/drswissmiss/status/1304856856649756673

one understands that *halted* is used as a passive participle, and *Vaccine trials* are the subordinate clause's direct object, as shown in Figure 1.2. We can rephrase the sentence with minimal changes to make it unambiguous:

(2) Vaccine trials that were halted after patient fell ill restart.

The choice for the initial parse can be explained in terms of frequency of occurrence: subject-verb-object sentences are encountered much more frequently than ones containing reduced relatives in everyday settings, making the first parse more likely (Fine et al., 2013). We refer to the verb causing the reanalysis as *disambiguator*, and to the difference in cognitive processing between (1) and (2), measured using proxies such as gaze metrics, as *garden-path effect* (Bever, 1970).

van Schijndel et al. (2020) present two families of cognitive processing theories trying to motivate the underlying difficulties in which humans incur with garden-path sentences:

- *Two-stage accounts* assume that readers consider only one or a subset of possible parses for each sentence that it is reading (Gibson, 1991; Jurafsky, 1996), and processing difficulties arise as a consequence of the reanalysis process need to reconstruct parses that were initially disregarded or not considered (Frazier and Fodor, 1978).

- *One-stage accounts* such as **surprisal theory** (Hale, 2001; Levy, 2008) instead consider difficulties produced by garden paths as the products of a single processing mechanism. Dispreferred parses are not discarded, but rather associated with a lower probability compared to that of likely ones: "processing difficulty on every word in the sentence, including the disambiguating words in garden-path sentences, arises from the extent to which the word shifts the reader's subjective probability distribution over possible parses" (van Schijndel et al., 2020).

There are multiple types of garden-path sentences, usually categorized based on their respective syntactic ambiguities (Frazier, 1978). In this work, two classic garden-path families are studied in three different settings using examples taken from Futrell, Wilcox, et al. (2019). The first type is the **MV/RR ambiguity** presented in example (1), and repeated in (3a):

(3) a. The woman brought the sandwich fell in the dining room. [RED., AMBIG.]

b. The woman who was brought the sandwich fell in the dining room. [UNRED., AMBIG.]

c. The woman given the sandwich fell in the dining room. [RED., UNAMBIG.]

d. The woman who was given the sandwich fell in the dining room. [UNRED., UNAMBIG.]

The label MV/RR indicates that *brought* can be initially parsed either as the main verb (MV) in the past tense of the clause or as a passive participle introducing a reduced relative (RR) clause, which postmodifies the subject. It is possible to rewrite the sentence by changing the ambiguous verb to an equivalent one having different forms for simple past and past participle (such as *gave* vs. *given*). In this case, we expect that the difference in cognitive processing for the disambiguator *fell* between the reduced (3c) and the unreduced (3d) version is smaller since the ambiguity is ruled out from the beginning.

The second type of ambiguity is the **NP/Z ambiguity** presented in (4a):

(4) a. As the criminal shot the woman yelled at the top of her lungs. [TRANS., NO COMMA]

b. As the criminal fled the woman yelled at the top of her lungs. [INTRANS., NO COMMA]

c. As the criminal shot, the woman yelled at the top of her lungs. [TRANS., COMMA]

d. As the criminal fled, the woman yelled at the top of her lungs. [INTRANS., COMMA]

The label NP/Z is used to indicate that the transitive verb *shot* can initially be understood to have either have a noun phrase (NP) object like *the woman* or a zero (Z), i.e., null object if used intransitively as it is the case for (4a). The sentence can be rewritten by substituting the transitive verb generating the ambiguity with an intransitive one, e.g., replacing *shot* with *fled* in (4b), by adding a disambiguating comma to force the null-object parse as in (4c), or by doing both as in (4d). We expect that the cognitive processing difference for the disambiguator *yelled* between the ambiguous (4a) and the unambiguous (4b) is smaller since the ambiguity is ruled out from the beginning.

As an additional NP/Z setting evaluation, consider the case in which an overt object is added to the verb introducing the ambiguity:

(5) a. As the criminal shot the woman yelled at the top of her lungs. [NO OBJ., NO COMMA]

b. As the criminal shot his gun the woman yelled at the top of her lungs. [OBJ., NO COMMA]

c. As the criminal shot, the woman yelled at the top of her lungs. [NO OBJ., COMMA]

d. As the criminal shot his gun, the woman yelled at the top of her lungs. [OBJ., COMMA]

Again, we expect that the difference in cognitive processing for *yelled* is higher in the non-object pair (5a)-(5c), where the first item is a garden-path sentence, rather than in the pair (5b)-(5d) where both sentences are unambiguous.

Gaze metrics and Garden-path Sentences As can be intuitively assumed, garden-path effects are reflected in gaze metrics collected during natural reading. Multiple studies have focused on quantifying the difference between garden-path sentences and their unambiguous counterparts on reading times in human subjects. Sturt et al. (1999) found a massive delay of 152ms for each word in the disambiguating region of NP/Z sentences. Grodner et al. (2003) estimate an average delay of 64ms over the disambiguating region for NP/Z constructs using 53 college students' reading times over a set of 20 ambiguous sentences. More recently, Prasad et al. (2019b) recorded eye measurements for 224 participants recruited through Amazon Mechanical Turk on the same set of NP/Z sentences as Grodner et al. (2003), finding a much lower average delay of 28ms, and suggesting an overestimation in previous studies due to small sample size and publication contingency to significant results. Prasad et al. (2019a) collected self-paced reading times from 73 participants recruited on the Prolific Academic crowdsourcing platform and measured an average delay of 22ms over the disambiguating region for MV/RR constructs.

Given the high variability in results across studies, it can be hypothesized that the way in which stimuli were presented to subjects plays a significant role in determining the magnitude of garden-path effects (Van Schijndel et al., 2018). For example, a sentence presented word-by-word to subjects may yield more ecologically valid reading times estimates than a sentence presented region-by-region. Another problematic factor involves constraining the impact of garden-path effects to the disambiguating region: first, because *parafoveal preview effects may slightly anticipate the start of the effect* (Schotter et al., 2012; Schotter, 2018); and second, because due to *spillover* (Mitchell, 1984), a phenomenon in which the surprisal of a word influences the reading times for itself and at least three subsequent words (Smith et al., 2013), reading times of the disambiguating region are influenced by preceding words, and influence subsequent ones, spreading the garden-path effect on a much broader context. For this reason, eye-tracking metrics are studied for all sentence regions in the experiments of Chapter 5.

Now it would be very remarkable if any system existing in the real world could be exactly represented by any simple model. [...] For such a model there is no need to ask the question "Is the model true?". The only question of interest is "Is the model illuminating and useful?".

— George Box (1976), *Science and Statistics*

2 | Models of Linguistic Complexity

Standard linguistic complexity studies analyze complexity annotations produced by human subjects to evaluate how specific language structures influence our perception of complexity under various viewpoints. For example, one can derive insights about early cognitive processing by looking at early gaze metrics, like first pass duration and first fixation duration, or study language comprehension by evaluating perceived complexity annotations. These approaches rely on a single implicit assumption: that *complexity annotations contain enough information to reflect the input's underlying complexity properties* appropriately. Without this premise, there would be a complete disconnect between human subjective perception, as reflected by annotations and linguistic structures. Given the ever-growing compelling evidence derived from carefully-planned complexity research, I argue that this is a relatively safe assumption to be made.

This work instead adopts a modeling-driven approach for the study of linguistic complexity. Annotations produced by human subjects still play a fundamental role in this context. However, instead of acting as the main subject of analysis, they are used as a source of distant supervision to create computational models of linguistic complexity. More specifically, machine learning models are trained to predict complexity annotation from raw input text by minimizing a task-specific loss function. The **learning step** here is fundamental, given the connection mentioned above between linguistic complexity and knowledge acquisition. After the training process, human annotations are put aside, and the model itself is studied as a complexity-sensitive subject: in particular, this study focuses on how the information encoded in the parameters of complexity-trained models is related to structural linguistic properties (Chapter 3), how this information differs when models are exposed to different complexity perspectives during training (Chapter 4) and finally how the encoded knowledge affects models' generalization capabilities over unseen constructs (Chapter 5).

While this approach still relies on the **annotation pertinence assumption** stated above, it requires making a second, stronger hypothesis: that *models employed can grasp a significant portion of the relations subsisting between language structures and complexity perspectives*.

This assumption can be further declined in two requirements. First, from a **conceptual** point-of-view, we must ensure that the model architecture is endowed with meaningful inductive biases concerning what is currently known about linguistic complexity. This includes having sufficient approximation capabilities to capture linguistic complexity phenomena, which are likely to be highly-nonlinear functions of the input. From a **functional** perspective, then, we should confirm that the quality of model predictions is sufficiently close to human-produced annotations to make their production mechanisms worth investigating.

This chapter justifies the selected modeling approach and introduces models later employed in complexity assessment experiments. Section 2.1 discusses the conceptual requirements for linguistic complexity modeling and motivates the choice of pretrained **neural language models** as primary subjects of this thesis work. Section 2.2 presents the architectures used in experimental sections and their desirable properties regarding the encoding of linguistic structures in latent representations. Finally, Section 2.3 presents the challenge of interpreting NLM's representations and behaviors and introduces various interpretability approaches used throughout this study.

2.1 Desiderata for Models of Linguistic Complexity

From the in-depth analysis of Chapter 1, we can distill some general desiderata for an idealized LCA model M^*. From a linguistic perspective:

- *M^* should distinguish between lexical forms and be informed about their probability of occurrence.* This is a basic (although fundamental) step given the importance of words' variety and frequency in determining our perception of complexity.

- *M^* should be aware of syntactic structures and sensitive to their properties.* As we saw with garden-path sentences, atypical or ambiguous syntax constructs are among the most prominent factors for determining the magnitude of processing difficulties. An ideal model should map complex syntactic constructs to higher complexity scores and discriminate potentially ambiguous or problematic structures from regular ones, even when changes in the form are minimal (e.g., when a single comma is missing).

- *M^* should capture semantic information and relations between entities.* Ideally, this means the ability to frame agents, patients, and actions in a semantic context and evaluate how likely or typical the latter is. For example, semantically unrelated entities occurring together in a sentence should produce an increase in processing difficulties. This includes the ability to disambiguate polysemic terms (e.g., "fly" verb vs. noun) given the surrounding context.

Then, from a technical standpoint:

- M^* *should not rely on hand-crafted features to represent language*. This is an implicit requirement since this study aims to analyze how the model autonomously learns to represent language in its parameters while simultaneously encoding information about its complexity. Chapter 3 presents how complexity models with hand-crafted features compare to those selected for the study.

- M^* *should not rely too heavily on labeled data.* Complexity datasets presented in Chapter 1 are usually composed of a few thousand labeled examples. While this may seem a lot to our eyes, a language model may require a lot more information to achieve sufficient generalization capabilities. A viable option in this context, as we will see with NLMs, is to prime models with general linguistic knowledge through an unsupervised pretraining procedure before training them on complexity-related tasks.

- M^* *should be sufficiently interpretable.* Ideally, we would like to draw direct causal relations from input properties to complexity prediction in a consistent way across complexity perspectives. More realistically, we need at least to find coherent patterns between the model's inputs and its predictive behaviors.

Most standard modeling approaches fail to encompass even a small subset of those non-trivial requirements. For example, one can consider modeling complexity properties with static word representations (Turian et al., 2010) such as Word2Vec or GloVe embeddings (Mikolov, Chen, et al., 2013; Pennington et al., 2014). In these approaches, feature vectors representing words are learned by a neural network through a pretraining procedure to model word co-occurrences. While these approaches were shown to capture a significant amount of semantic information while reducing the dependence on labeled data thanks to pretraining, static word embeddings generally yield modest results when employed for syntactic predictions (Andreas et al., 2014). Moreover, since the model learns a direct mapping $f : t_i \rightarrow \mathbf{v}_i$ from lexical forms to vectorial representations, polysemic terms are reduced to single context-independent representation, and contextual information that often plays a crucial role in determining complexity is mixed and diluted.

Among more sophisticated modeling approaches for representing language, I argue that modern **neural language models** (NLMs) are the approaches that yield a better match for the requirements stated above. These models consist of multi-layer neural networks (Goodfellow et al., 2016) pretrained using standard language modeling or masked language modeling training objectives to produce **contextualized word embeddings**, which were shown to be very effective in downstream syntactic and semantic tasks (Peters et al., 2018) even with relatively few labeled examples. Moreover, being language models, NLMs predict a probability distribution over their

vocabulary at each step, enabling us to compute information-theoretic metrics such as surprisal that we saw being conceptually close to one-stage cognitive processing accounts. Finally, their high parameter counts and the presence of self-attention mechanisms (Bahdanau et al., 2015; Vaswani et al., 2017) as learned weighting functions suggests that NLMs might be capable of learning to approximate highly nonlinear functions effectively.

The most significant downside of NLMs in the context of our analysis is their opaqueness. As for most neural networks, the nonlinear multi-layer structure that characterizes NLMs makes them incredibly valid function approximators. At the same time, though, it hinders our efforts in interpreting their behaviors (Samek et al., 2019). Because of this fact, in recent years, we witnessed a surge in approaches trying to "open the black box" of neural networks by using various techniques borrowed from information theory (Shwartz-Ziv et al., 2017) and cognitive science (Kriegeskorte et al., 2008). Given the wide availability of these approaches, this work joins the choir of interpretability researchers and argues that studying how such performant models encode their knowledge about language complexity is still a matter of interest and worth exploring. In the next section, the architecture and training process of NLMs will be formalized, and their properties will be described in detail.

2.2 Neural Language Models: Unsupervised Multitask Learners

The objective of natural language processing applications such as *summarization*, *machine translation*, and *dialogue generation* is to produce text that is both **fluent** and contextually accurate. As we saw in Chapter 1, a text's fluency can also be used as a significant factor in determining its complexity from a linguistic viewpoint. A possible approach to establishing a sentence's fluency is to rely on **relative frequency estimates** for words in large corpora. Consider a sentence s and a large corpus $\mathcal{C}$. We can estimate its probability of occurrence in natural language as:

$$P(s) = \frac{\text{count}(s)}{|\mathcal{C}|} \tag{2.1}$$

While this is an unbiased estimator since it converges to the actual frequency value when the corpus size is sufficiently large, it is both very data-reliant and highly unreliable. If a sentence happens to be absent in $\mathcal{C}$, it will be assigned probability equal to zero. Therefore, we need to rely on other approaches, such as language models, to obtain reliable estimates from limited training datasets.

As we saw in Chapter 1.2.2, language models assign probabilities to sequences of tokens. Formally, this can be framed as learning words' conditional probability distributions given their context, either *preceding* or *bidirectional* depending on the language modeling approach. I will here refer to sequential language models unless otherwise mentioned.

Language models are trained on sequences $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$ composed by n tokens taken from a predefined vocabulary $\mathcal{V}$. Each token x_i can be represented as a one-hot encoded vector $x_i \in \{0,1\}^{|\mathcal{V}|}$, and the probability of sequence $\mathbf{x}$ is factored using the chain rule:

$$P(\mathbf{x}) = \prod_{t=1}^{n} P(x_t \,|\, x_1, \ldots, x_{t-1}) \tag{2.2}$$

After the training process, we can use the likelihood that the model assigns to **held-out data** $\mathbf{y}$ treated as a single stream of m tokens as an intrinsic evaluation metric for the quality of its predictions:

$$\ell(\mathbf{y}) = \sum_{t=1}^{m} \log P(x_t | x_1, \ldots, x_{t-1}) \tag{2.3}$$

$\ell(\mathbf{y})$ can be rephrased in terms of **perplexity**, an information-theoretic metric independent from the size of the held-out set:

$$\mathrm{PPL}(\mathbf{y}) = 2^{-\ell(\mathbf{y})/m} \tag{2.4}$$

PPL is equal to 1 if the language model is perfect (i.e., predicts all tokens in the held-out corpus with probability 1) and matches the vocabulary size $|\mathcal{V}|$ when the model assign a uniform probability to all tokens in the vocabulary (a "random" language model):

$$\log_2(\mathbf{y}) = \sum_{t=1}^{m} \log_2 \frac{1}{|\mathcal{V}|} = -\sum_{t=1}^{m} \log_2 |\mathcal{V}| = -m \log_2 |\mathcal{V}| \tag{2.5}$$

$$\mathrm{PPL}(\mathbf{y}) = 2^{\frac{1}{m} m \log_2 |\mathcal{V}|} = 2^{\log_2 |\mathcal{V}|} = |\mathcal{V}| \tag{2.6}$$

Perplexity represents the number of bits required to encode the average word in the corpora. For example, reporting a perplexity score of 10 over a held-out corpus means that the language model will predict on average words with the same accuracy as if it had to choose uniformly and independently across ten possibilities for each word.

While tokens used by language models generally correspond to words in most NLP pipelines, recent language modeling work highlighted the effectiveness of using subword tokens (Sennrich et al., 2016; Wu et al., 2016; Kudo et al., 2018) or even single characters to further improve LM's generalization performances. In particular, models used in this work rely on SentencePiece and Byte-Pair Encoding (BPE) subword tokenization (Sennrich et al., 2016; Kudo et al., 2018). The SentencePiece algorithm derives a fixed-size vocabulary from word co-occurrences in a large corpus and treats whitespace as a normal symbol by converting it to "_", while BPE does the same using the "Ġ" character. For example:

Input sentence: Heteroscedasticity is hard to model!

SentencePiece tokenization: _Hetero s ced astic ity _is _hard _to _model !

BPE tokenization: H eter os ced astic ity Ġis Ġhard Ġto Ġmodel !

where whitespaces correspond to separators after tokenization. From the example, we can observe that frequent words like *hard*, *to* and *model* are treated similarly by both tokenizers, while rare words like *heteroscedasticity* are split into subwords depending on their observed frequency inside the tokenizer's training corpus.

In recent years n-gram language models, which were the most common approach to estimate probabilities from relative frequencies, have been largely supplanted by neural networks. A significant advantage of neural approaches is the overcoming of context restrictions: relevant information can be incorporated from arbitrarily distant contexts while preserving the tractability of the problem from both a statistical and a computational viewpoint.

Neural language models treat language modeling as a *discriminative* learning task aimed at maximizing the log conditional probability of a corpus. Formally, the probability distribution $p(x|c)$ is reparametrized as the dot product of two dense numeric vectors $\theta_x, h_c \in \mathbb{R}^H$ under a softmax transformation:

$$P(x|c) = \frac{\exp(\theta_x \cdot h_c)}{\sum_{x' \in \mathcal{V}} \exp(\theta_{x'} \cdot h_c)} \tag{2.7}$$

In (2.7), the denominator is present to ensure that the probability distribution is properly normalized over vocabulary $\mathcal{V}$. θ_x represent model parameters that can be learned through an iterative procedure, while h_c is the contextual information that can be computed in different ways depending on the model. For example, a neural language model based on the **recurrent neural network** architecture (RNN; Mikolov, Karafiát, et al. (2010)) recurrently updates context

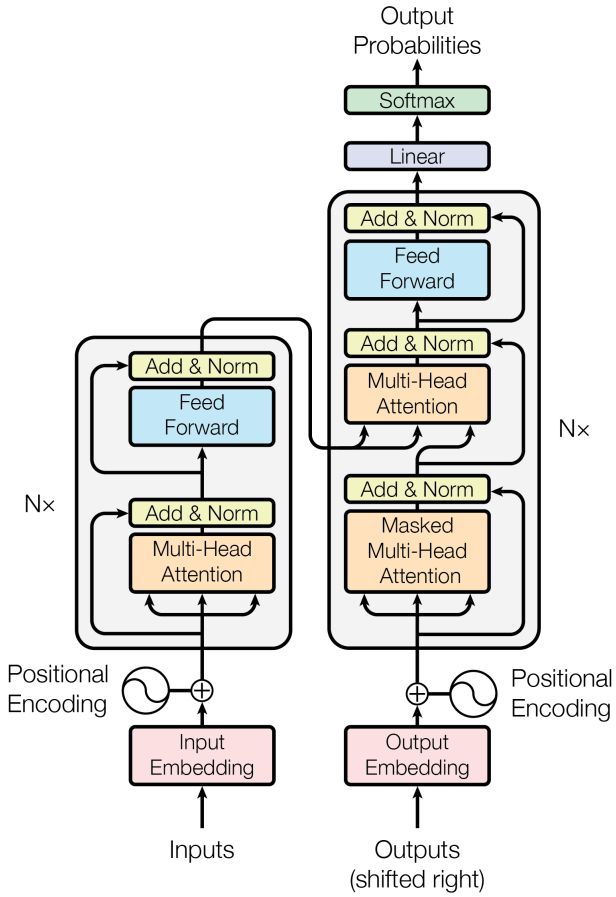


Figure 2.1: The original Transformer model architecture by Vaswani et al. (2017).

vectors initialized at random with relevant information that needs to be preserved while moving through the sequence.[1]

This work leverages models belonging to the most recent and influential family of neural language models at the time of writing, that is, the one based on the **Transformer** architecture (Vaswani et al., 2017). Transformers are deep learning models designed to handle sequential data and were conceived to compensate for a significant downside of recurrent models: the need to process data in an orderly manner to perform backpropagation through time. By replacing recurrent computations with attention mechanisms to maintain contextual information throughout the model, Transformers' operations are entirely parallelizable on dedicated hardware and *therefore lead to reduced training times*. This fact is especially relevant considering the massive corpora size used to pretrain neural language models to obtain contextual representations. **Self-attention** was also shown to behave better than other approaches at learning long-range dependencies, avoiding the *vanishing gradient* problem that plagued non-gated recurrent NLMs altogether (Pascanu et al., 2013).

[1] Refer to Chapter 6.3 of Eisenstein (2019) for additional details about recurrent language models.

The original Transformer architecture comprises an encoder and a decoder, each composed of a stacked sequence of identical layers that transform input embeddings in outputs with the same dimension (hence the name). First, the encoder maps the sequence $(x_1, \ldots, x_n)$ to a sequence of embeddings $z = (z_1, \ldots, z_n)$. Given z, the decoder then autoregressively produces an output token sequence $(y_1, \ldots, y_m)$. Each layer of the Transformer encoder comprises two sublayers, a **multi-head self-attention mechanism** and a **feed-forward network**, surrounded by residual connections and followed by layer normalization. The decoder includes a third layer that performs multi-head self-attention over the encoder output and modifies the original self-attention sublayer to prevent attending to future context, as required by the language modeling objective. Figure 2.1 presents the original architecture for a N-layer Transformer. I will now proceed to describe the main components of the Transformer model.

Positional Encodings The original Transformer relies on two sets of embeddings to represent the input sequence: learned **word embeddings**, used as vector representations for each token in the vocabulary, and fixed **positional encodings** (PEs) used to inject information about the position of tokens in the sequence. Those are needed since no information about the sequential nature of the input would otherwise be preserved. For position *pos* and dimension i, PEs correspond to sinusoidal periodic functions that were empirically shown to perform on par with learned embeddings, and were chosen to enable extrapolation for longer sequences:

$$PE_{pos,2i} = \sin(\text{pos}/10000^{2i/|h|}) \tag{2.8}$$

$$PE_{pos,2i+1} = \cos(\text{pos}/10000^{2i/|h|}) \tag{2.9}$$

where $|h|$ is the model's hidden layer size. Embeddings and PEs are summed and passed to the attention layer.

Self-Attention The *scaled dot-product self-attention* mechanisms is the driving force of the Transformer architecture. Given an input embedding matrix X, we multiply it by three weight matrices W^Q, W^K, W^V obtaining the projections Q (**queries**), K (**keys**) and V (**values**). Those are then combined by the self-attention function as follows:

$$\text{Attention(Q,K,V)} = \text{softmax}\Big(\frac{QK^T}{\sqrt{d_k}}\Big)V \tag{2.10}$$

where d_k is the size of individual query and key vectors. The output of this operation is a matrix Z which will be passed to the feed-forward layer. The self attention mechanism is further extended to **multi-head self-attention** in Transformer architectures. In the multi-head variant, the attention function is applied in parallel to n version of queries, keys and values projected with learned parameter matrices, and outputs are finally concatenated and projected again to obtain final values:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\ldots,\text{head}_n)W^O \tag{2.11}$$

$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.12}$$

Where $W_i^Q \in \mathbb{R}^{|h|\times d_k}$, $W_i^K \in \mathbb{R}^{|h|\times d_k}$, $W_i^V \in \mathbb{R}^{|h|\times d_v}$ and $W^O \in \mathbb{R}^{nd_v\times|h|}$. In multi-head attention layers of Figure 2.1, each position can attend to all position from the previous layer, while in the **masked multi-head attention** layer only previous positions in the sequence can be attended by applying a triangular mask to attention matrices. This additional step is needed to preserve the autoregressive property during decoding.

Feed-forward Layer Each block in the encoder and the decoder contains an independent fully connected 2-layer feed-forward network with a ReLU nonlinearity applied separately to each position of the sequence:

$$\text{FFN}(Z) = \max(0, Z\Theta_1 + b_1)\Theta_2 + b_2 \tag{2.13}$$

where Z are the representations passed forward from the attention sublayer, Θ_1, Θ_2 are two learned independent parameter matrices for each layer and b_1, b_2 are their respective bias vectors.

Now that the main concepts regarding the Transformer architecture have been introduced, the two Transformer-based models used in this study will be presented.

GPT-2 GPT-2 (Radford et al., 2019) is a transformer model built using only the decoder blocks with masked self-attention, alongside BPE tokenization. The latter's autoregressive capabilities, i.e. being able to iteratively add a newly predicted token to the existing sequence in the next steps, make it especially suitable for text generation and related tasks. The learning of model parameters is performed in two stages. First, an **unsupervised pretraining** is carried out to learn a high capacity language model on a large corpus: in particular, here the model is trained to maximize the likelihood of sequential language modeling over **WebText**, a corpus containing roughly 8 million documents (40GB of text), by adapting its parameters using stochastic gradient

descent. The purpose of this step is to learn contextual word embeddings encoding both low and high-level information that can be recycled in downstream tasks, following the **transfer learning** approach inspired by the field of computer vision and initially proposed by Howard et al. (2018) for NLP. The second step is a **supervised fine-tuning**, where the language modeling softmax layer is replaced by a task-specific layer (called **head**) with parameters W_y receiving final transformer activations h_l and predicting a label y (e.g. in a classification task) as:

$$P(y|x_1,\dots,x_m) = \text{softmax}(h_l^{sent} W_y) \tag{2.14}$$

where h_l^{sent} is the sentence-level representation for $(x_1,\dots,x_m)$. The parameters of the whole model, including transformer blocks and task-specific heads, can then be tuned by minimizing the loss $\mathcal{L}$ over the whole supervised corpus $\mathcal{C}$:

$$\mathcal{L}(\mathcal{C}) = -\sum_{(x,y)} \log P(y|x_1,\dots,x_m) \tag{2.15}$$

Figure 2.2 visualizes the forward pass through the GPT-2 architecture. We see from the figure that attention patterns learned during pre-trained are often interpretable. Here, the token *it* is correctly identified as the pronoun referring to the subject *a robot*. Authors show how large NLMs such as GPT-2 become strong unsupervised multitask learners when trained on sufficiently large corpora, providing the initial motivation for choosing pretrained Transformer models for experiments throughout this study. GPT-2 will be specifically be employed in the experiments of Chapter 5, where its autoregressive nature is ideal for replicating human surprisal estimates during sequential reading on garden-path sentences.

ALBERT ALBERT (Lan et al., 2020) is an efficient variant of the Bidirectional Encoder Representations from Transformers (**BERT**) approach by Devlin et al. (2019). BERT was built following the intuition that many sentence-level tasks would greatly benefit from an approach capable of incorporating bidirectional context inside language representations. This is not the case for decoder-based approaches like GPT-2 that, being aimed at generation-oriented tasks, could only leverage the previous context using masked self-attention. BERT tackles the unidirectional constraint by introducing **masked language modeling** (MLM, see Equation (1.5)) and using a stack of transformer encoder layers with GELU nonlinearities (Hendrycks et al., 2016).

As for GPT-2, the pretraining and fine-tuning steps are taken to provide the model with general language knowledge and subsequently adapt it to specific downstream tasks. At each pretraining step, a fixed portion of input tokens get masked, and the model predicts the original

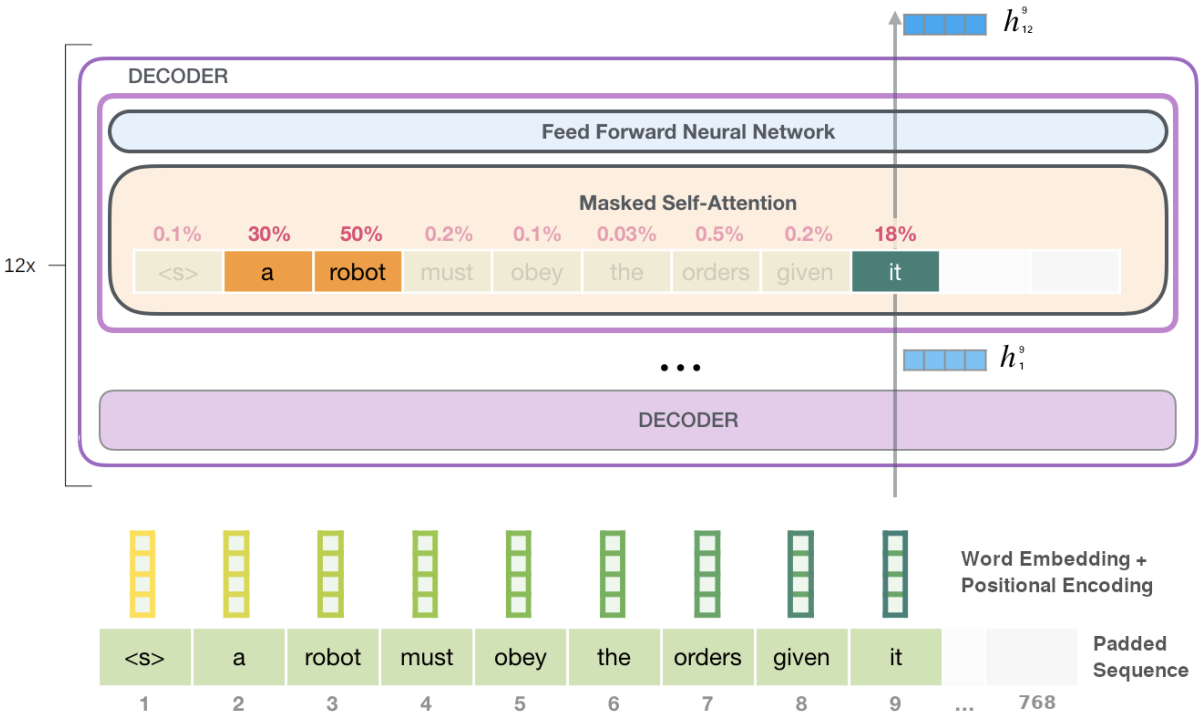


Figure 2.2: An overview of the forward pass in GPT-2. Adapted from Alammar (2018b).

vocabulary id of masked tokens. Moreover, a sentence-level task is used to improve discourse coherence. For BERT, the **next sentence prediction** (NSP) task is adopted, i.e. determining whether, given two sentences, they are consecutive or not in the original text using both positive and negative pairs. NSP was found unreliable by subsequent studies and was replaced in ALBERT by a **sentence ordering prediction** loss that is more challenging for the model. A third set of **segment embeddings** is added to initial representations to distinguish input sentences in multi-sentence tasks. Special tokens `[CLS]` and `[SEP]` are added as sentence-level representations.

ALBERT introduces two main contributions aimed at reducing the final number of model parameters inside BERT:

- **Factorized embedding parametrization**: a projection layer is introduced between the embedding matrix E and the hidden layer H of the model so that the dimensions of the two are untied. This approach modifies embedding parameter count from $O(|V| \times |E|)$ to $O(|\mathcal{V}| \times |E| + |E| \times |h|)$, with $|\mathcal{V}|, |E|, |h|$ being respectively the sizes of vocabulary, embedding vectors and hidden layers. A significant reduction in model parameters is therefore produce when $|h| \gg |E|$, which is desirable since H contains *context-dependent representations* that encode more information than the *context-independent* ones of E.

- **Cross-layer parameter sharing**: All layers of ALBERT share the same set of feed-forward and self-attention parameters. Therefore, we can see ALBERT as an iterated

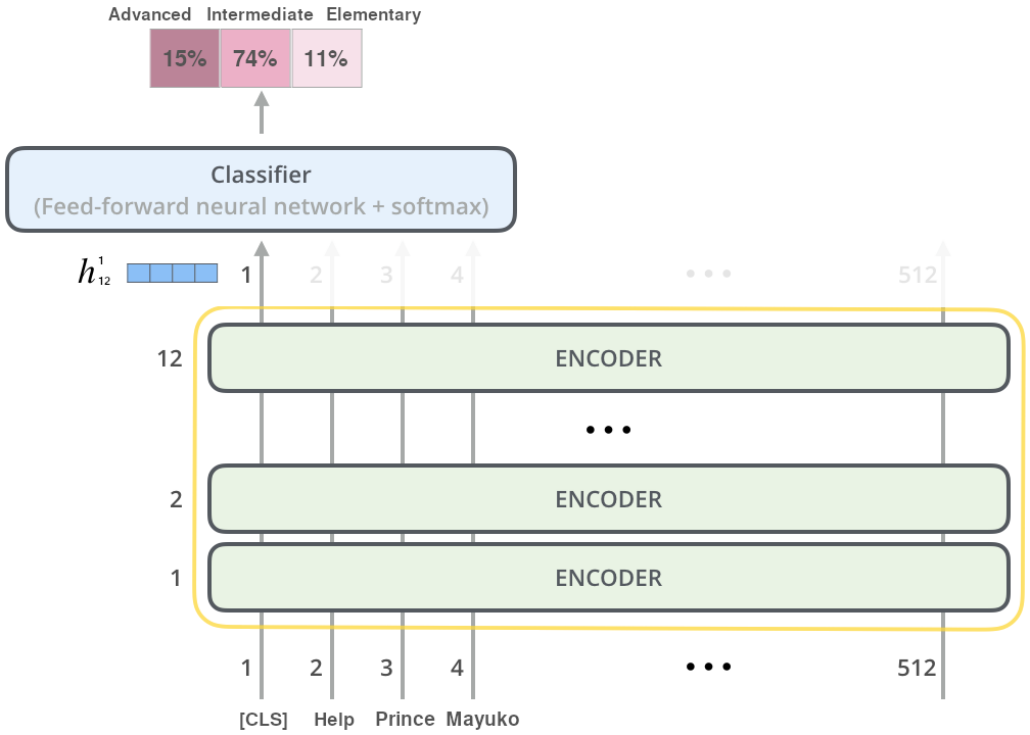


Figure 2.3: Using a pretrained ALBERT model for the ARA task. Adapted from Alammar (2018a).

function $f_A^n : h \rightarrow h'$, where n is the number of encoder layers present in the model (in this study $n = 12$), with parameters trained using end-to-end stochastic gradient descent.

Both factors significantly contribute to reducing the computational complexity of the model without affecting too much its performances: the ALBERT base used in all experimental chapters of this study have 9x fewer parameters than a regular BERT base model (12M vs. 108M) while performing comparably well on many natural language understanding benchmarks such as GLUE (Wang et al., 2018) and SQuAD (Rajpurkar et al., 2016).

Figure 2.3 presents how a pretrained ALBERT model can be leveraged for sentence classification, using the ARA task as an example. We note that the procedure is the same as for GPT-2: a task-specific classification head is initialized with random weights, and the whole model-head architecture is fine-tuned on the target task end-to-end. The figure also shows how the common choice for BERT-based models is to use their `[CLS]` token h_{12}^{1} as the full-sentence representation equivalent h_{12}^{sent}.

To conclude, the fine-tuning approach relying on a pretrained model "body" and a task-specific head adopted in both GPT-2 and ALBERT can be extended out-of-the-box to a **multitask learning** scenario. A multitask approach can prove useful when considering parallel annotations on the same corpus that provide similar but complementary information about a studied phenomenon's nature. We can interpret this as an inductive bias that encourages finding knowledge

representations to explain multiple sets of annotations at once.[2] More specifically, multitask learning with **hard parameter sharing** (Caruana, 1997) is performed in all experimental sections over eye-tracking scores to produce representations encompassing the whole set of phenomena related to natural reading. For doing so, each metric was associated with a task-specific head, and the whole set of heads was trained while sharing the same underlying model.

2.2.1 Emergent Linguistic Structures in Neural Language Models

This section presents evidence in support of the ability of pretrained language models to effectively encode language-related properties in their learned representations.[3]

Lin et al. (2019) were among the first to highlight how BERT representations encode hierarchical structures akin to syntax trees, despite the absence of syntactic information or recurrent biases during pretraining. Liu et al. (2019) and Tenney et al. (2019) further showed that contextualized embeddings produced by BERT encode information about part-of-speech, entity roles, and partial syntactic structures.

Hewitt and Manning (2019) formulate the **syntax distance hypothesis**, assuming that there exists a linear transformation B of the word representation space under which vector distance encodes parse trees. They proceed to test this assumption equating L2 distance in the 2-dimensional space of representations projected by $B \in \mathbb{R}^{2\times|h|}$ and tree distances in parse trees, finding a close match between BERT representational space and Penn Treebank formalisms. The approach is visualized in Figure 2.4. Jawahar et al. (2019) work support these findings, highlighting a close match between BERT representation and dependency trees after testing multiple decomposition schemes. The syntax distance hypothesis's validity is especially relevant to this work, given the aforementioned importance of syntactic properties in driving human subjects' perception of complexity.

Despite the evidence of syntactic knowledge in contextual word representations, recent results suggest that the model may not leverage this for its predictions. Ettinger (2020) highlights the insensitivity of BERT to negation and malformed inputs using psycholinguistic diagnostics commonly used with human subjects, while Wallace et al. (2019) show that nonsensical inputs do not affect the prediction quality of BERT, despite having a clear input on underlying syntactic structures. These results are coherent with the experimental findings of this study and will be further discussed in later sections.

[2] See Ruder (2017) for a comprehensive overview

[3] Rogers et al. (2020) and Linzen and Baroni (2021) are surveys covering this topic.

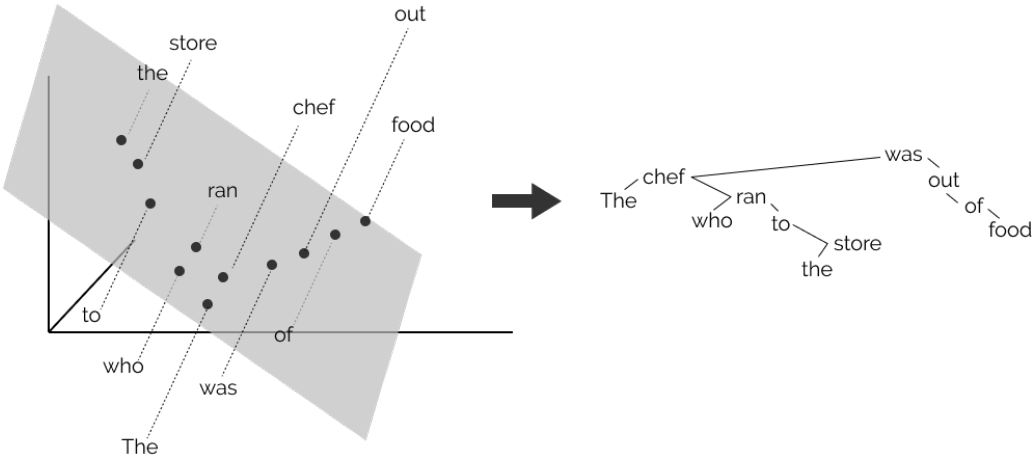


Figure 2.4: The mapping from 2D representation space to syntax tree distances adopted in Hewitt and Manning (2019).

2.3 Analyzing Neural Models of Complexity

Having introduced the model architectures that will be used throughout this study, we will now focus on the interpretability approaches allowing us to analyze and compare neural network representations.

When training deep neural networks, we would like to go beyond predictive performance and understand how different design choices and training objectives affect learned representations from a qualitative viewpoint. This fact is especially crucial in the model-driven approach adopted in this work, as stated at the end of Section 2.1. While for linear models, the direct correspondence between the magnitude of feature coefficients and feature importance provides us with some out-of-the-box insights about decision boundaries and feature importance, the hierarchical and nonlinear structure that characterizes neural networks produce model weights that are relatively uninformative when taken in isolation.

This work focuses on two interpretability perspectives: highlighting linguistic knowledge encoded in model representations (Chapter 3) and comparing representations across models trained on different complexity-related tasks (Chapter 4). For the first objective, *probing classifiers*, which have become the de-facto standard in the interpretability literature, are used to evaluate the amount of information encoded in each layer of the model.[4] In the second case, two multivariate statistical analysis methods, namely *representational similarity analysis* and *canonical correlation analysis*, are leveraged to quantify the relation between model embeddings by evaluating their second-order similarity and learning a mapping to a shared low-dimensional space, respectively. The following sections conclude the chapter by presenting the three approaches in detail.

[4]See Belinkov and Glass (2019) survey and Belinkov, Gehrmann, et al. (2020) tutorial.

2.3.1 Probing classifiers

The **probing task approach** is a natural way to estimate the mutual information shared by a neural network's parameters and some latent property that the model could have implicitly learned during training. During probing experiments, a supervised model (*probe*) is trained to predict the latent information from the network's learned representations. If the probe does well, we may conclude that the network effectively encodes some knowledge related to the selected property.

Formally speaking, let $f : x_i \rightarrow y_i$ be a neural network model mapping a corpus of input sentences $X = (x_1, \ldots, x_n)$ to a set of outputs $Y = (y_1, \ldots, y_n)$. Assume that each sentence x_i is also labeled with some linguistic annotations z_i, reflecting the underlying properties we aim to detect. Let also $h_l(x_i)$ be the network's output at the l-th layer given the sentence x_i as input. To estimate the quality of representations h_l with respect to property z, a supervised model $g : h_l(x_i) \rightarrow z_i$ mapping representations to property values is trained. We take such model's performances as a proxy of $H(h_l(x), z)$. In information theoretic terms, the probe is trained to minimize entropy $H(z|h_l(x))$, and by doing that it maximizes mutual information between the two quantities.

The probe g does not need to be a linear model. While historically simple linear probes were used to minimize the risk of memorization, recent results show that more complex probes produce tighter estimates for the actual underlying information (Pimentel et al., 2020). To account for the probe's ability to learn the task through sheer memorization, Hewitt and Liang (2019) introduce *control tasks* using the performances of a probe exposed to random labels as baselines.

Alain et al. (2016) were among the first to use linear probing classifiers as tools to evaluate the presence of task-specific information inside neural networks' layers. The approach was later extended to the field of NLP by Conneau et al. (2018) and Zhang et al. (2018) *inter alia*, which evaluated the presence of semantic and syntactic information inside sentence embeddings generated by LSTM encoders (Hochreiter et al., 1997) pretrained on different objectives using probing task suites. Recently, Miaschi and Dell'Orletta (2020) showed how contextual representations produced by pretrained Transformer models could encode sentence-level properties within single-word embeddings. Moreover, Miaschi, Brunato, et al. (2020) highlighted the tendency of pretrained NLMs to lose general linguistic information during the fine-tuning process and found a positive relation between encoded linguistic information and the downstream performances of the model.

2.3.2 Representational Similarity Analysis

Representational similarity analysis (RSA, Laakso et al. (2000)) is a technique developed in the field of cognitive science to evaluate the similarity of fMRI responses in selected regions of the brain after a stimulus (Kriegeskorte et al., 2008). The technique can be extended to compare

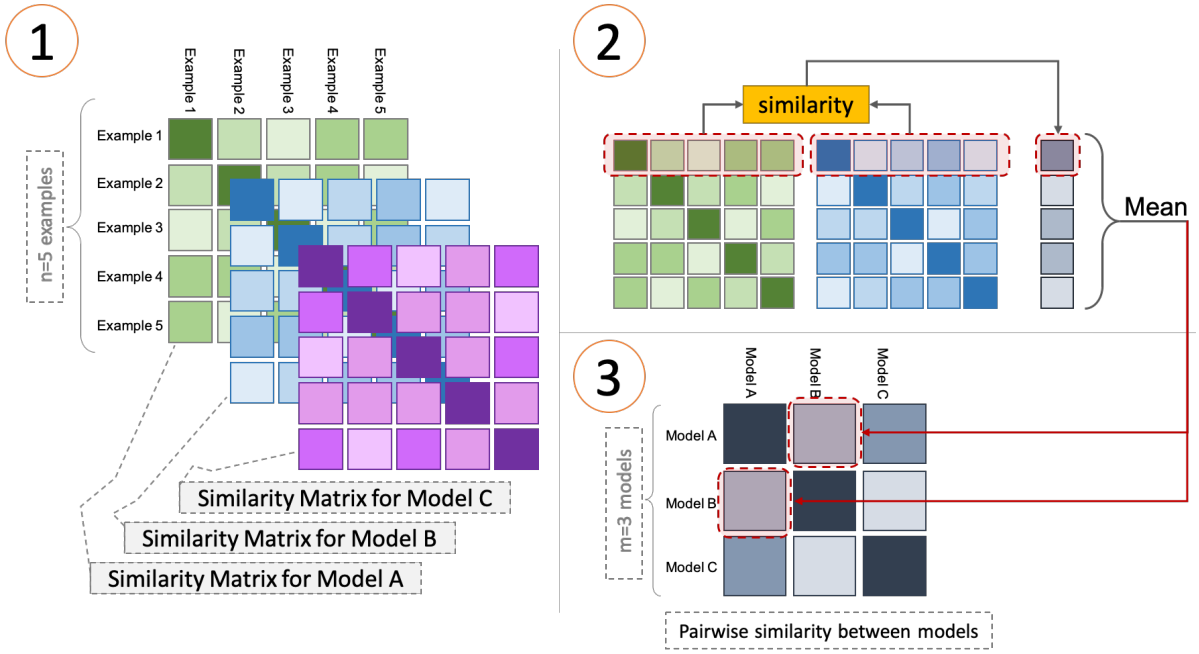


Figure 2.5: The Representational Similarity Analysis (RSA) algorithm applied to the representations of three models. Image taken from Abnar (2020).

the heterogeneous representational spaces formed by a set of computational models m exposed to a shared set of observations. Figure 2.5 visualizes the approach. First, each model is fed with a shared corpus of n sentences to produce a set of matrix embeddings $(E^1, \ldots, E^m)$, where E^i_j represents the embedding produced by the last layer of the i-th model on the j-th sentence of the corpus.[5] Next, for each matrix E^i a representational distance matrix S^i is produced such that $S^i_{j,k} = \text{sim}(E^i_j, E^i_k)$, $S^i \in \mathbb{R}^{n \times n}$ where sim_1 is a similarity function (here, *dot product*). S_i encodes information on the similarity subsisting between model activations across different observations. Finally, a second-level *representational similarity matrix* S' is computed, where for each pair of matrices (S^i, S^j) the corresponding $S'_{i,j}$ entry has value:

$$S'_{i,j} = S'_{j,i} = \frac{1}{n} \sum_{k=1}^{n} \text{sim}_2\big(\eta\,(S^i_k), \eta\,(S^j_k)\big) \tag{2.16}$$

where η is the L1 normalization function and sim_2 is a similarity function (here, *Pearson's correlation coefficient*). Each entry $S'_{i,j}$ corresponds to a similarity score between activity patterns of model i and model j across the entire set of n observations.

In the context of NLP, Abnar, Beinborn, et al. (2019) recently used RSA to compare the activations of multiple neural language models and evaluated the impact of parameter values on

[5] This can be any layer; embeddings can be produced by different layers of the same model.

the representations formed by a single model. Interestingly, they also use RSA to compare fMRI imaging data collected from human subjects and NLMs activations. Abdou et al. (2019) use RSA to highlight the connection between processing difficulties (measured by high gaze metrics values) and the representational divergence, both inter and intra-encoder. Abnar, Dehghani, et al. (2020) visualize training paths of various neural network architectures as 2D projections of RSA and show how different inductive biases can be transferred across network categories using knowledge distillation (Hinton et al., 2015).

2.3.3 Projection-Weighted Canonical Correlation Analysis

Canonical correlation analysis (CCA, Thompson (1984)) is a statistical technique for relating two sets of observations arising from an underlying unknown process. In the context of this work, the underlying process is represented by NLMs being trained on complexity-related tasks. Given a corpus of sentences $X = (x_1, \ldots, x_m)$ annotated with complexity labels, we have that $z^l_=(z^l_i(x_1), \ldots z^l_i(x_m))$ corresponds to all activations of neuron z_i at layer l stacked to form a vector.[6] If we consider all activations of all neurons in a layer $L_i = (z^i_1, \ldots, z^i_n)$ for all inputs, we can represent them as a matrix $A_i \in \mathbb{R}^{m \times n}$, i.e. a set of multidimensional variates where n is the number of neurons in the layer. The CCA algorithm aims to *identify the best* (i.e. most correlated) *linear relationship under mutual orthogonality and norm constraints between two sets of multidimensional variates*, which in this case are activation matrices like L_1. This approach was used, among other things, to study the coherence between modeled and real brain activations (Sussillo et al., 2015).

Formally, if we have two activation matrices $A_1, A_2 \in \mathbb{R}^{m \times n}$ we aim to find vectors $w, v \in \mathbb{R}^m$ such that the correlation:

$$\rho = \frac{\langle w^T A_1, v^T A_2 \rangle}{\|w^T A_1\| \cdot \|v^T A_2\|} \tag{2.17}$$

is maximized. The formula can be solved by changing the basis and recurring to singular value decomposition. The output of CCA is a set of singular pairwise orthogonal vectors u, v and their canonical correlation coefficients $\rho \in [0, 1]$ representing the correlation of vectors $w^T A_1$ and $v^T A_2$.

The SVCCA method (Raghu et al., 2017) extends the CCA approach for deep learning research by pruning neurons through a singular value decomposition step before computing canonical correlation coefficients. As the authors mention, "This is especially important in neural network representations, where as we will show many low variance directions (neurons)

[6] Different from the activation vector, i.e. all neurons' activations for a single input $(z^l_1(x_1), \ldots, z^l_n(x_1))$

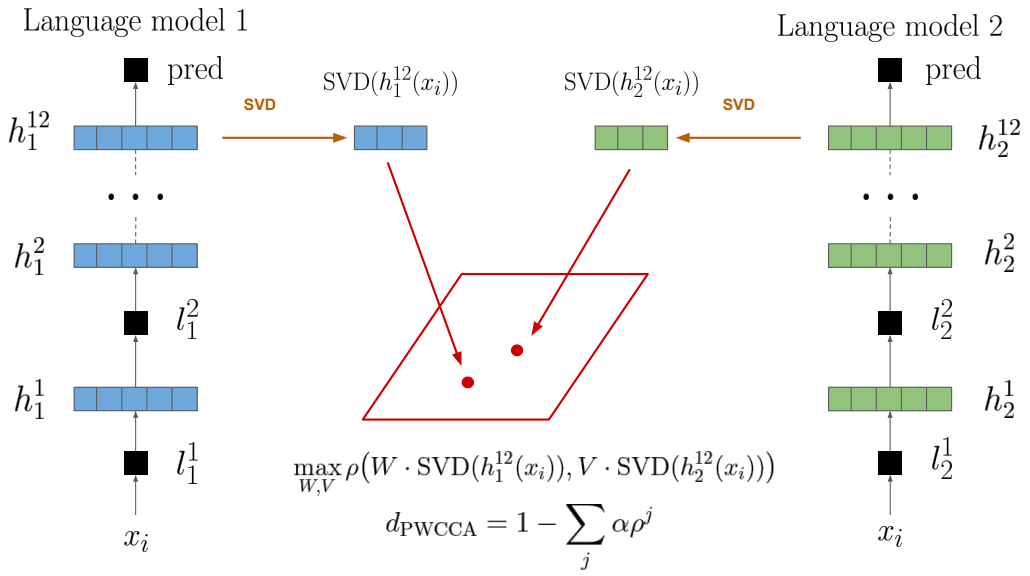


Figure 2.6: Projection-Weighted Canonical Correlation Analysis (PWCCA) applied to last-layer representations of two language models.

are primarily noise". Then, the similarity between two layers L_1, L_2 is computed as the mean correlation coefficient produce by SVCCA, and adapted to a distance measure for evaluation:

$$d_{\mathrm{SVCCA}}(A_1, A_2) = 1 - \frac{1}{|\rho|}\sum_{i=1}^{|\rho|}\rho^{(i)} \tag{2.18}$$

Morcos et al. (2018) suggest that the equal importance given to all the $|\rho|$ SVCCA vectors during the final averaging step may be problematic since it has been extensively shown that overparametrized neural networks often do not recur to their full dimensionality for representing solutions (Frankle et al., 2018). They suggest replacing the mean with a weighted mean:

$$d_{\mathrm{PWCCA}}(A_1, A_2) = 1 - \sum_{i=1}^{|\rho|}\alpha\rho^{(i)} \text{ with } \tilde{\alpha}_i = \sum_j |\langle h_i, x_j\rangle| \tag{2.19}$$

where weights α corresponds to the portion of inputs x accounted for by CCA vectors h and $\tilde{\alpha}_i$ values are normalized such that $\sum_i \alpha_i = 1$. The resulting approach, *projection-weighted canonical correlation analysis* (PWCCA), is used in this study and was shown to be much more robust than SVCCA to filter noise in activations. Figure 2.6 visualizes the selected approach.

Notable applications of CCA-related methods in NLP are Saphra et al. (2019), where SVCCA is used to study the evolution of LSTM language models' representations during training, and Voita et al. (2019), where PWCCA is used to compare Transformer language models across layers and pretraining objectives.

3 | Complexity Phenomena in Linguistic Annotations and Language Models

This chapter investigates the relationship between online gaze metrics and offline perceived complexity judgments by studying how the two viewpoints are represented by a neural language model trained on human-produced data. First, a preliminary analysis of linguistic phenomena associated with the two complexity viewpoints is performed, highlighting similarities and differences across metrics. The effectiveness of a regressor based on explicit linguistic features is then evaluated for sentence complexity prediction and compared to the results obtained by a fine-tuned neural language model with contextual representations. In conclusion, the linguistic competence inside the language model's embeddings is probed before and after fine-tuning, showing how linguistic information encoded in representations changes as the model learns to predict complexity.

Given the conceptual similarity between raw cognitive processing and human perception of complexity, this chapter investigates whether the relation between eye-tracking metrics and complexity judgments can be highlighted empirically in human annotations and language model representations. With this aim, linguistic features associated with various sentence-level structural phenomena are analyzed in terms of their correlation with offline and online complexity metrics. The performance of models using either complexity-related explicit features or contextualized word embeddings is evaluated, focusing mainly on the neural language model ALBERT (Lan et al., 2020) introduced in Section 2.2. The results highlight how both explicit features and learned representations obtain comparable performances when predicting complexity scores. Finally, the focus is shifted to studying how complexity-related properties are encoded in the representations of ALBERT.

This perspective goes in the direction of exploiting human processing data to address the interpretability issues of unsupervised language representations (Hollenstein, de la Torre, et al., 2019; Gauthier and Levy, 2019; Abnar, Beinborn, et al., 2019), leveraging the *probing task* approach introduced in Section 2.3.1. It is observed that online and offline complexity fine-tuning produces a consequent increase in probing performances for complexity-related features during probing experiments. This investigation has the specific purpose of studying whether and how learning a new task affects the linguistic properties encoded in pretrained representations. While pre-trained models have been widely studied using probing methods, the effect of fine-tuning on encoded information was seldom investigated. To my best knowledge, no previous work

has taken into account sentence complexity assessment as a fine-tuning task for NLMs. Results suggest that the model's abilities during training are interpretable from a linguistic perspective and are possibly related to its predictive capabilities for complexity assessment.

Contributions This is the first work displaying the connection between online and offline complexity metrics and studying how a neural language model represents them. This work:

- Provides a comprehensive analysis of linguistic phenomena correlated with eye-tracking data and human perception of complexity, addressing similarities and differences from a linguistically-motivated perspective across metrics and at different levels of granularity;
- Compares the performance of models using both explicit features and unsupervised contextual representations when predicting online and offline sentence complexity; and
- Shows the natural emergence of complexity-related linguistic phenomena in the representations of language models trained on complexity metrics.[1]

3.1 Data and Preprocessing

The experiments of this chapter leverage two corpora, each capturing different aspects of linguistic complexity:

Eye-tracking For online complexity metrics, only the monolingual English portion of GECO (Cop et al., 2017), presented in Section 1.3.3, was used. Four online metrics spanning multiple phases of cognitive processing are selected, respectively: *first pass duration* (FPD), *total fixation count* (FXC), *total fixation duration* (TFD) and *total regression duration* (TRD) (see Table 1.3 for more details). Metrics are sum-aggregated at sentence-level and averaged across participants to obtain a single label for each metric-sentence pair. As a final step to make the corpus more suitable for linguistic complexity analysis, all utterances with fewer than five words, deemed uninteresting from a cognitive processing perspective, are removed.

Perceived Complexity For the offline evaluation of sentence complexity, the English portion of the corpus by Brunato, De Mattei, et al. (2018) was used (Section 1.3.2). Sentences in the corpus have uniformly-distributed lengths ranging between 10 and 35 tokens. Each sentence is associated with 20 ratings of perceived-complexity on a 1-to-7 point scale. Duplicates and sentences for which less than half of the annotators agreed on a score in the range $\mu_n \pm \sigma_n$, where

[1] Code available at https://github.com/gsarti/interpreting-complexity

Table 3.1: Descriptive statistics of the two sentence-level corpora after the preprocessing procedure.

	Perceived Complexity	**Eye-tracking (GECO)**
labels	PC	FPD, FXC, TFD, TRD
domain(s)	financial news	literature
aggregation steps	avg. annotators	sentence sum-aggregation + avg. participants
filtering steps	filtering by agreement + remove duplicates	min. length > 5
# of sentences	1115	4041
# of tokens	21723	52131
avg. sent. length	19.48	12.9
avg. token length	4.95	4.6
Length-binned subsets (# of sentences)		
Bin 10±1 size	173	899
Bin 15±1 size	163	568
Bin 20±1 size	164	341
Bin 25±1 size	151	215
Bin 30±1 size	165	131
Bin 35±1 size	147	63

μ_n and σ_n are respectively the average and standard deviation of all annotators' judgments for sentence n were removed to reduce noise coming from the annotation procedure. Again, scores are averaged across annotators to obtain a single metric for each sentence.

Table 3.1 presents an overview of the two corpora after preprocessing. The resulting eye-tracking (ET) corpus contains roughly four times more sentences than the perceived complexity (PC) one, with shorter words and sentences on average. The differences in sizes and domains between the two corpora account for multi-genre linguistic phenomena in the following analysis.

3.2 Analysis of Linguistic Phenomena

As a first step to investigate the connection between the two complexity paradigms, the correlation of online and offline complexity labels with various linguistic phenomena is evaluated. The Profiling-UD tool (Brunato, Cimino, et al., 2020) introduced in Section 1.2.1 is used to annotate each sentence in our corpora and extract from it ~100 features representing their linguistic structure according to the Universal Dependencies formalism (Nivre et al., 2016). These features capture a comprehensive set of phenomena, from basic information (e.g. sentence and word length) to more complex aspects of sentence structure (e.g. parse tree depth, verb arity), including properties related to sentence complexity at different levels of description. A summary of the most relevant features is presented in Appendix A. Features are ranked using their Spearman's

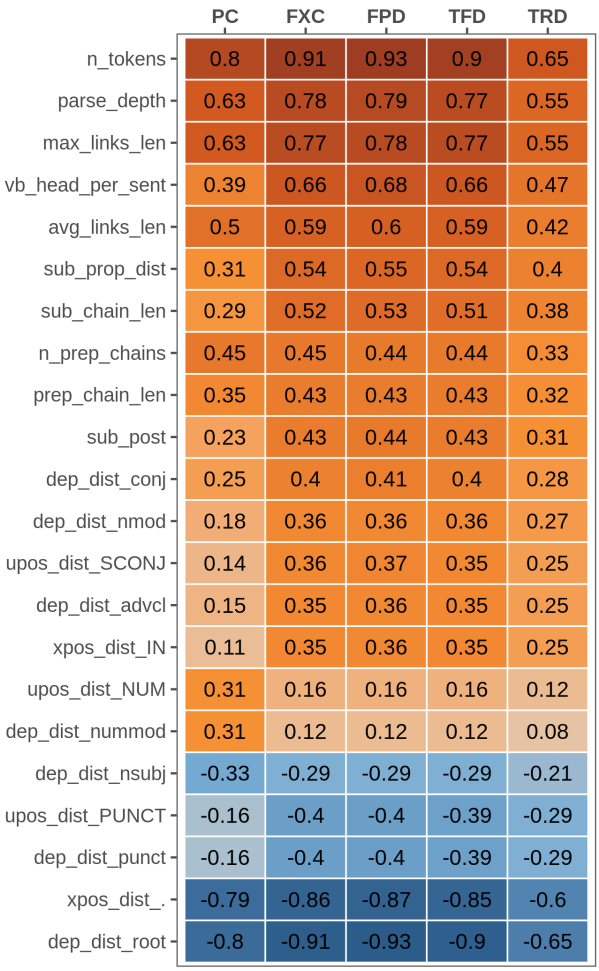


Figure 3.1: Ranking of the most correlated linguistic features for selected metrics. All of Spearman's correlation coefficients have $p < 0.001$.

correlation score with complexity metrics, and scores are leveraged to highlight the relation between linguistic phenomena and complexity paradigms.

The correlation scores analysis highlights how features showing a significant correlation with eye-tracking metrics are twice as many as those correlating with PC scores and generally tend to have higher coefficients, except for the total regression duration (TRD) metric. Nevertheless, the most correlated features are the same across all metrics. Figure 3.1 reports correlation scores for features showing a strong connection ($|\rho| > 0.3$) with at least one of the evaluated metrics. As expected, sentence length (*n_tokens*) and other related features capturing structural complexity aspects occupy the top positions in the ranking. Among those, we can note the length of dependency links (*max_links_len, avg_links_len*) and the depth of the whole parse tree or selected sub-trees, i.e. nominal chains headed by a preposition (*parse_depth, n_prep_chains*). Similarly, the distribution of subordinate clauses (*sub_prop_dist, sub_post*) is positively correlated with all metrics but with a more substantial effect for eye-tracking ones, especially in the presence of

longer embedded chains (*sub_chain_len*). Interestingly, the presence of numbers (*upos_NUM, dep_nummod*) affects only the offline perception of complexity, while it is never strongly correlated with all eye-tracking metrics. This finding is expected since numbers are very short tokens and, like other functional POS, were never found to be strongly correlated with online reading in our results. Conversely, numerical information has been identified as a factor hampering sentence readability and understanding (Rello et al., 2013).

3.2.1 Linguistic Phenomena in Length-controlled Bins

Unsurprisingly, sentence length is the most correlated predictor for all complexity metrics. Since many linguistic features highlighted in our analysis are strongly related to sentence length, we tested whether they maintain a relevant influence when this parameter is controlled. To this end, Spearman's correlation was computed between features and complexity tasks, but this time considering bins of sentences having approximately the same length. Specifically, we split each corpus into six bins of sentences with 10, 15, 20, 25, 30, and 35 tokens, respectively, with a range of ±1 tokens per bin to select a reasonable number of sentences for our analysis. Resulting subsets have a relatively constant size for the PC corpus, which was constructed ad-hoc to have such uniform length distribution, but have a sharply decreasing size for the eye-tracking corpus (see Table 3.1, bott. While deemed appropriate in the context of this correlation analysis, the disparity in bin sizes may play a significant role in hampering the performances of models trained on binned linguistic complexity data. This perspective is discussed in Section 3.3.

Figure 3.2 reports the new rankings of the most correlated linguistic features within each bin across complexity metrics ($|\rho| > 0.2$). Again, we observe that features showing a significant correlation with complexity scores are fewer for PC bins than for eye-tracking ones. This fact depends on controlling for sentence length and the small size of bins for the whole dataset. As in the coarse-grained analysis, TRD is the eye-tracking metric less correlated to linguistic features, while the other three (FXC, FPD, TFD) show a homogeneous behavior across bins. For the latter, vocabulary-related features (token-type ratio, average word length, lexical density) are always positive and top-ranked in all bins, especially when considering shorter sentences (i.e. from 10 to 20 tokens). For PC, this is true only for some of them (word length and lexical density). On another note, features encoding numerical information are still highly correlated with the offline perception of complexity in almost all bins.

Interestingly, features modeling subordination phenomena extracted from fixed-length sentences exhibit a reverse trend than when extracted from the whole corpus, i.e. they are negatively correlated with judgments. If, on the one hand, an increase in the presence of subordination for longer sentences (possibly making sentences more convoluted) was expected,

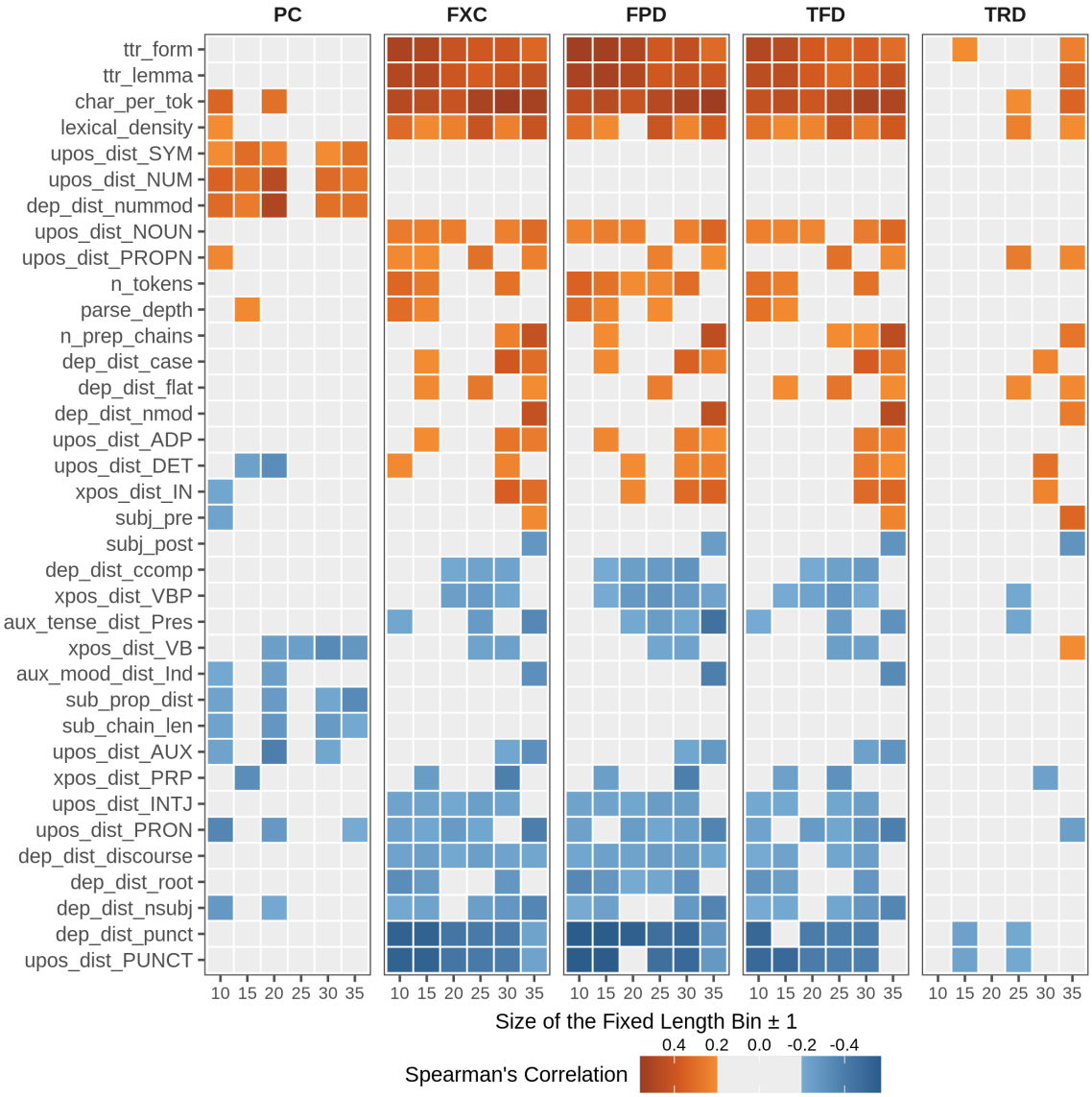


Figure 3.2: Rankings of the most correlated linguistic features for metrics within length-binned subsets of the two corpora. Squares show the correlation between features (left axis) and a complexity metric (top) at a specific bin of length (bottom). Coefficients ≥ 0.2 or $\leq$ -0.2 are highlighted, and have $p < 0.001$.

on the other hand, when the length is controlled, findings suggest that subordinate structures are not necessarily perceived as a symptom of sentence complexity.

The analysis also highlights how linguistic features relevant to online and offline complexity are different when controlling for sentence length. This aspect, in particular, was not evident from the previous coarse-grained analysis. Despite blocking sentence length, gaze measures are still significantly connected to length-related phenomena (high correlation with *n_tokens* at various length bins). This observation can be possibly due to the ±1 margin applied for sentence selection and the high sensitivity of behavioral metrics to small input changes.

Table 3.2: Average Root-Mean-Square Error ($\sqrt{E^2}$) and R^2 score values for sentence-level complexity predictions using 5-fold cross-validation. Lower $\sqrt{E^2}$ and higher R^2 are better.

	PC		FXC		FPD		TFD		TRD	
	$\sqrt{E^2}$	R^2	$\sqrt{E^2}$	R^2	$\sqrt{E^2}$	R^2	$\sqrt{E^2}$	R^2	$\sqrt{E^2}$	R^2
Statistical baselines										
Avg. score	0.87	0	6.17	0.06	1078	0.06	1297	0.06	540	0.03
Bin average	0.53	0.62	2.36	0.86	374	0.89	532	0.85	403	0.45
Explicit features										
SVM length	0.54	0.62	2.19	0.88	343	0.9	494	0.86	405	0.45
SVM feats	**0.44**	0.74	**1.77**	**0.92**	**287**	**0.93**	**435**	**0.92**	400	0.46
Learned representations										
ALBERT	**0.44**	**0.75**	1.98	**0.92**	302	**0.93**	**435**	0.9	**382**	**0.49**

3.3 Modeling Online and Offline Linguistic Complexity

Given the high correlations reported above, the next step involves quantifying the importance of explicit linguistic features from a modeling standpoint. Table 3.2 presents the RMSE and R^2 scores of predictions made by baselines and models for the selected complexity metrics. Performances are tested with a 5-fold cross-validation regression with a fixed random seed on each metric. Our baselines use average metric scores of all training sentences (*Avg. score*) and average scores of sentences binned by their length, expressed in number of tokens, as predictions (*Bin average*). The two linear SVM models leverage explicit linguistic features, using respectively only the *n_tokens* feature (*SVM length*) and the whole set of linguistic features presented above (*SVM feats*). Besides those, the performances of a state-of-the-art Transformer neural language model relying entirely on contextual word embeddings are equally tested. *ALBERT* (Lan et al. (2020); see Section 2.2) as a lightweight yet effective alternative to BERT (Devlin et al., 2019) for obtaining contextual word representations, using its last-layer `[CLS]` sentence embedding as input for a linear regressor during fine-tuning and testing. We selected the last layer representations, despite strong evidence on the importance of intermediate representation in encoding language properties, because we aim to investigate how superficial layers encode complexity-related competence. Given the availability of parallel eye-tracking annotations, we train ALBERT using multitask learning with hard parameter sharing (Caruana, 1997) on gaze metrics.[2]

From Table 3.2 it can be noted that:

- The length-binned average baseline is very effective in predicting complexity scores and gaze metrics, which is unsurprising given the extreme correlation between length and complexity metrics presented in Figure 3.1;

[2]Training procedure and parameters are thoroughly described in Appendix F.

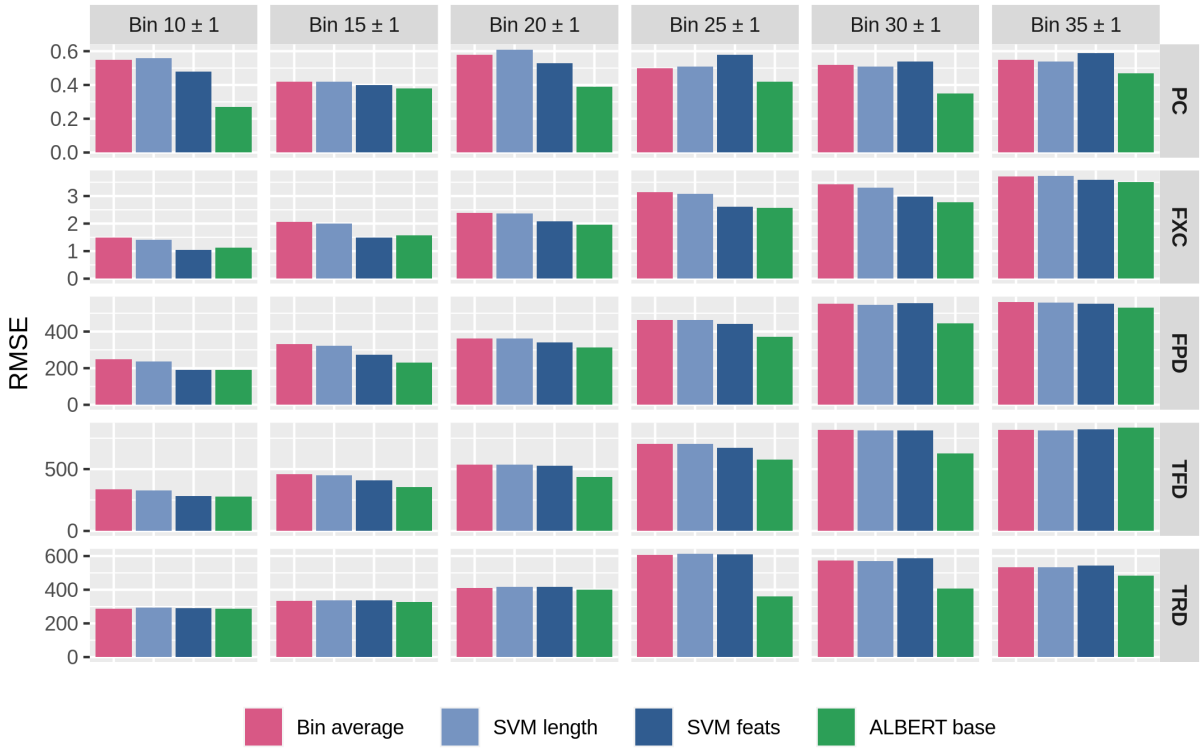


Figure 3.3: Average Root-Mean-Square Error (RMSE) scores for models in Table 3.2, performing 5-fold cross-validation on the length-binned subsets used for Figure 3.2. Lower scores are better.

- The *SVM feats* model shows considerable improvements if compared to the length-only SVM model for all complexity metrics, highlighting how length alone accounts for much but not for the entirety of variance in complexity scores;
- ALBERT performs on-par with the SVM feats model on all complexity metrics despite the small dimension of the fine-tuning corpora and the absence of explicit linguistic information.

A possible interpretation of ALBERT's strong performances is that the model implicitly develops competence related to phenomena encoded by linguistic features while training on online and offline complexity prediction. We explore this perspective in Section 3.4.

3.3.1 Modeling Complexity in Length-controlled Bins

Similarly to the approach adopted in Section 3.2.1, the performances of models are tested on length-binned data to verify their consistency in the context of length-controlled sequences. Figure 3.3 presents RMSE scores averaged with 5-fold cross-validation over the length-binned sentences subsets for all complexity metrics. It can be observed that ALBERT outperforms

the SVM with linguistic features on nearly all bins and metrics, showing the largest gains on intermediate bins for PC and gaze durations (FPD, TFD, TRD). Interestingly, models' overall performances follow a length-dependent increasing trend for eye-tracking metrics, but not for PC. This behavior can be possibly explained in terms of the high sensibility to length previously highlighted for online metrics, as well as the broad variability in bin dimensions. It can also be observed how the SVM model based on explicit linguistic features (*SVM feats*) performs poorly on larger bins for all tasks, sometimes being even worse than the bin-average baseline. While this behavior seems surprising given the positive influence of features highlighted in Table 3.2, this phenomenon can be attributed to the small dimension of longer bins, which negatively impacts the generalization capabilities of the regressor. The relatively better scores achieved by ALBERT in those, instead, support the effectiveness of information stored in pretrained language representations when a limited number of examples are available.

3.4 Probing Linguistic Phenomena in ALBERT Representations

As shown in the previous section, ALBERT performances in complexity predictions are comparable to those of an SVM relying on explicit linguistic features and even better than those when controlling for length. The *probing task* interpretability paradigm (Section 2.3.1) is adopted to investigate if ALBERT encodes the linguistic knowledge that we identified as strongly correlated with online and perceived sentence complexity during training and prediction. In particular, the aim of this investigation is two-fold:

- Probing ALBERT's innate competence in relation to the broad spectrum of linguistic features described in Appendix A; and
- Verifying whether, and in which respect, this competence is affected by a fine-tuning process on the complexity assessment metrics.

Three UD English treebanks spanning different textual genres – **EWT, GUM, and ParTUT** respectively by Silveira et al. (2014), Zeldes (2017), and Sanguinetti et al. (2015) – were aggregated, obtaining a final corpus of 18,079 sentences with gold linguistic information which was used to conduct probing experiments. The Profiling-UD tool was again leveraged to extract n sentence-level linguistic features $\mathcal{Z} = z_1, \ldots, z_n$ from gold linguistic annotations. Representations $A(x)$ were generated for all corpus sentences using the last-layer `[CLS]` embedding of a pretrained ALBERT base model without additional fine-tuning, and n single-layer perceptron regressors $g_i : A(x) \rightarrow z_i$ are trained to map representations $A(x)$ to each linguistic feature z_i. Finally, the error and R^2 scores of each g_i were evaluated as proxies for the quality of representations $A(x)$

Table 3.3: Root MSE ($\sqrt{E^2}$) and R^2 scores for diagnostic regressors trained on ALBERT representations, respectively, without fine-tuning (Base), with PC and eye-tracking (ET) fine-tuning on all data (left) and on the 10 ± 1 length-binned subset (right). **Bold** values highlight relevant increases in R^2 from Base.

	Base		**PC**		**ET**		**PC10±1**		**ET10±1**	
	$\sqrt{E^2}$	R^2	$\sqrt{E^2}$	R^2	$\sqrt{E^2}$	R^2	$\sqrt{E^2}$	R^2	$\sqrt{E^2}$	R^2
n_tokens	8.19	0.26	4.66	**0.76**	2.87	**0.91**	8.66	0.18	6.71	**0.51**
parse_depth	1.47	0.18	1.18	**0.48**	1.04	**0.6**	1.50	0.16	1.22	**0.43**
vb_head_per_sent	1.38	0.15	1.26	**0.3**	1.14	**0.42**	1.44	0.09	1.30	**0.25**
xpos_dist_.	0.05	0.13	0.04	**0.41**	0.04	**0.42**	0.04	0.18	0.04	**0.38**
avg_links_len	0.58	0.12	0.53	**0.29**	0.52	**0.31**	0.59	0.1	0.56	**0.2**
max_links_len	5.20	0.12	4.08	**0.46**	3.75	**0.54**	5.24	0.11	4.73	**0.28**
n_prep_chains	0.74	0.11	0.67	**0.26**	0.66	**0.29**	0.72	0.14	0.69	**0.21**
sub_prop_dist	0.35	0.09	0.33	0.13	0.31	**0.22**	0.34	0.05	0.32	0.15
upos_dist_PRON	0.08	0.09	0.08	0.14	0.08	0.07	0.07	**0.23**	0.08	0.15
pos_dist_NUM	0.05	0.08	0.05	0.06	0.05	0.02	0.05	**0.16**	0.05	0.06
dep_dist_nsubj	0.06	0.08	0.06	0.1	0.06	0.05	0.05	**0.17**	0.06	0.11
char_per_tok	0.89	0.07	0.87	0.12	0.90	0.05	0.82	**0.22**	0.86	0.14
prep_chain_len	0.60	0.07	0.57	**0.17**	0.56	**0.19**	0.59	0.12	0.56	**0.18**
sub_chain_len	0.70	0.07	0.67	**0.15**	0.62	**0.26**	0.71	0.04	0.66	**0.16**
dep_dist_punct	0.07	0.06	0.07	0.06	0.07	**0.14**	0.07	0.06	0.07	**0.14**
dep_dist_nmod	0.05	0.06	0.05	0.07	0.05	0.06	0.05	0.09	0.05	0.09
sub_post	0.44	0.05	0.46	0.12	0.44	**0.18**	0.47	0.05	0.45	**0.14**
dep_dist_case	0.07	0.05	0.06	0.06	0.07	0.08	0.07	0.07	0.07	0.1
lexical_density	0.14	0.05	0.13	0.03	0.13	0.03	0.13	**0.13**	0.13	**0.13**
dep_dist_compound	0.06	0.04	0.06	0.05	0.06	0.03	0.06	0.1	0.06	0.07
dep_dist_conj	0.04	0.03	0.04	0.04	0.04	0.04	0.05	0.02	0.04	0.03
ttr_form	0.08	0.03	0.08	0.05	0.08	0.05	0.08	0.05	0.08	0.05
dep_dist_det	0.06	0.03	0.06	0.02	0.06	0.04	0.06	0.03	0.06	0.03
dep_dist_aux	0.04	0.02	0.04	0.01	0.04	0.01	0.04	0.06	0.04	0.04
pos_dist_VBN	0.03	0.01	0.03	0	0.03	0	0.03	0.01	0.03	0
xpos_dist_VBZ	0.04	0.01	0.04	0.01	0.04	0.02	0.04	0.02	0.04	0.02
ttr_lemma	0.09	0.01	0.09	0.06	0.09	0.06	0.09	0.04	0.09	0.03

in encoding their respective linguistic feature z_i. The same evaluation is repeated for ALBERTs fine-tuned respectively on perceived complexity labels (PC) and on all eye-tracking labels with multitask learning (ET), averaging scores with 5-fold cross-validation. A selected subset of results is shown on the left side of Table 3.3.

As it can be observed, ALBERT's last-layer sentence representations have relatively low knowledge of complexity-related probes, but their performances highly increase after fine-tuning. Specifically, a noticeable improvement was obtained on features that were already better encoded in base pretrained representation, i.e. sentence length and related, suggesting that fine-tuning possibly accentuates only properties already well-known by the model, regardless of the target task. To verify that this isn't the case, the same probing tests were repeated on ALBERT

models fine-tuned on the smallest length-binned subset (i.e. 10 ± 1 tokens) presented in previous sections. The right side of Table 3.3 presents the resulting scores. From the length-binned correlation analysis of Section 3.2, PC scores were observed to be mostly uncorrelated with length phenomena, while ET scores remain significantly affected despite our controlling of sequence size. This observation also holds for length-binned probing task results, where the PC model seems to neglect length-related properties in favor of task-specific ones that were also highlighted in our fine-grained correlation analysis (e.g. word length, numbers, explicit subjects). The ET-trained model follows the same behavior, retaining strong but lower performances for length-related features.

In conclusion, although higher probing task performances after fine-tuning are not direct proof that the neural language model exploits newly-acquired morpho-syntactic and syntactic information, results suggest that training on tasks strongly connected with underlying linguistic structures triggers a change in model representations resulting in a better encoding of related linguistic properties.

3.5 Summary

In this chapter, the connection between eye-tracking metrics and the offline perception of sentence complexity was investigated from an experimental standpoint. An in-depth correlation analysis was performed between complexity scores and sentence linguistic properties at different granularity levels, highlighting the strong relationship between metrics and length-affine properties and revealing different behaviors when controlling for sentence length. Models using explicit linguistic features and unsupervised word embeddings were evaluated on complexity prediction, showing comparable performances across metrics. Finally, the encoding of linguistic properties in a neural language model's contextual representations was tested with probing tasks. This approach highlighted the natural emergence of task-related linguistic properties within the model's representations after the fine-tuning process. Thus, it can be conjectured that a relation subsists between the model's linguistic abilities during the training procedure and its downstream performances on morphosyntactically-related tasks and that linguistic probes may provide a reasonable estimate of the task-oriented quality of representations.

4 Representational Similarity in Models of Complexity

The experiments of this chapter aim to shed light on how the linguistic knowledge encoded in the contextual representations of complexity-trained neural language models varies across layers of abstraction and fine-tuning tasks. Two similarity approaches, Representational Similarity Analysis (RSA) and Projection-Weighted Canonical Correlation Analysis (PWCCA) are used to evaluate the relation subsisting between representations spanning different models and different layers of the same model. The outcomes are finally compared against a set of assumptions aimed at determining a model's generalization capabilities across language phenomena. Results provide empirical evidence about the inability of state-of-the-art language modeling approaches to effectively represent an abstract hierarchy of linguistic complexity phenomena.

Chapter 3 highlighted how the relation between online and offline complexity perspectives and linguistic phenomena diverge when considering same-length sentences and how those properties of language are adequately captured by a neural language model fine-tuned on complexity metrics. This chapter adopts a complementary perspective on the model-driven study of complexity. Instead of connecting learned representations to the input's structural properties, it explores how those representations change when the same model is exposed to different training objectives using similarity measures. This approach is used to gain insights on the underlying similarities across complexity metrics, using representations as proxies for the knowledge needed to correctly model various complexity phenomena under a minimal set of assumptions.

The same ALBERT (Lan et al., 2020) model introduced in Section 2.2 and used for the last section's probing task experiments is leveraged for this chapter's experiments.[1] The model is first taken as-is in its pre-trained version without fine-tuning (referred to as **Base**). Then, three instances of it are fine-tuned respectively on **Automatic Readability Assessment** (RA, Section 1.3.1), **Perceived Complexity Prediction** (PC, Section 1.3.2) and **Eye-tracking Metrics Prediction** (ET, Section 1.3.3) until convergence. The four models are evaluated in two settings: first, by comparing the similarity of same-layer representation across models (*inter-model similarity*), and then comparing the similarity across different layers of the same model (*intra-model similarity*). For each setting, two similarity metrics are used: Representational Similarity Analysis (RSA, Section 2.3.2) and Projection-Weighted Canonical Correlation

[1] The `albert-base-v2` checkpoint from 🤗 `transformers` (Wolf et al., 2020) is used.

Analysis (PWCCA, Section 2.3.3). RSA and PWCCA were selected since they provide different perspectives over the similarity of representations: if, on the one hand, RSA naively evaluates the similarity across input representations through correlation, PWCCA factors in the importance of sparsity patterns that characterize overparametrized neural networks using a projection operation. Both token and sentence-level representations are evaluated to obtain a fine-grained overview of representational similarity.

The models trained on perceived complexity and eye-tracking metrics are again the main subjects of this study, given the logical and empirical relation subsisting between the two complexity perspectives highlighted in previous chapters. The additional use of Base and readability-trained models allows us to verify whether ALBERT representations satisfy a minimal set of assumptions deemed necessary and sufficient for modeling an abstraction hierarchy of linguistic complexity phenomena in an interpretable fashion. Results produced by representational similarity experiments diverge significantly from the initial hypothesis, suggesting the prominence of surface structures and task setups over underlying general knowledge about the nature of the modeled phenomena in shaping representations during the training process.

Contributions While multiple works aimed at inspecting NLM representations by mean of similarity approaches already exist, this is the first work to the best of my knowledge that does so with the explicit purpose of evaluating the impact of linguistic complexity training. This work:

- Highlights similarity and differences in the representations of models trained on different complexity-related tasks to understand how neural network parameters capture different perspectives over linguistic complexity after the training process;

- Presents similarity and differences in the representations found at different layers of the same model to understand how knowledge is distributed hierarchically at various abstraction levels after training;

- Provide evidence about the inability of state-of-the-art NLP approaches to learning to effectively represent an abstract hierarchy of linguistic complexity phenomena in an unsupervised manner, relying solely on complexity-related annotations.[2]

[2]Code available at https://github.com/gsarti/interpreting-complexity

4.1 Knowledge-driven Requirements for Learning Models

At the beginning of Chapter 2 two prerequisites to any model-driven study were defined: that available annotated corpora should be informative about the underlying phenomena we are trying to model, and that sufficiently elaborate models should be able to represent knowledge to solve phenomena-related tasks after being trained on those corpora effectively. This section formalizes the two assumptions and builds upon them to define a set of fundamental requirements that should be satisfied by models capable of generalizing over unseen linguistic structures after undergoing a learning process. Let:

- $\mathcal{C}^{\phi}_{\alpha} = \Big[(x_1, \alpha_1) \dots (x_m, \alpha_m)\Big]$ be an annotated corpus containing some knowledge relative to an abstract phenomenon of interest ϕ encoded in its annotations α. x can represent any i-th linguistic structure or substructure (sentence, word, morpheme). This notation can be generalized to settings where annotations are not explicitly defined (e.g. in the context of language modeling, next structure $x_i + 1$ acts as an annotation for x_i) or when multiple annotations are present (e.g. if $\mathcal{C}$ has two sets of annotations α, β modeling the same phenomenon $\mathcal{K}$ is equivalent to two corpora $\mathcal{C}^{\phi}_{\alpha}, \mathcal{C}^{\phi}_{\beta}$ with shared x's).

- M be a model that, after being trained on $\mathcal{C}^{\phi}_{\alpha}$, learns representations (i.e. parameters) that allow him to map correctly linguistic structures to annotations

- $\mathcal{K}^{\phi}$ be a set containing all empirical knowledge that is specifically relevant to phenomenon ϕ. $\mathcal{K}^{\phi}_{\alpha}$ represents all knowledge relative to ϕ contained in a corpus $\mathcal{C}^{\phi}_{\alpha}$. Concretely, given a corpus $\mathcal{C}^{\phi}_{\alpha}$, we can logically infer from it some estimate knowledge $\tilde{\mathcal{K}}^{\phi}_{\alpha}$ such that $\tilde{\mathcal{K}}^{\phi}_{\alpha} \simeq \mathcal{K}^{\phi}_{\alpha} \subset \mathcal{K}^{\phi}$.

- $\varsigma^{\phi}_{\alpha,\beta}(x)$ be an idealized similarity function reflecting the similarity between two sets of representations in performance-driven terms relative to phenomenon ϕ, i.e. measuring their invariance in relation to all knowledge sets $\mathcal{K}^{\varphi}$, with $\phi \neq \varphi$ that are irrelevant to phenomenon ϕ.

For example, taking linguistic complexity as ϕ, and the GECO corpus as $\mathcal{C}^{\phi}_{\alpha}$ (with α being e.g. the total fixation duration annotations), we may have $\tilde{\mathcal{K}}^{\phi}_{\alpha}$ (i.e. our inferred knowledge about linguistic complexity) contains the observation $o =$ "longer structures are more complex" because longer words have longer total fixation durations on average. Note that the relation $o \in \mathcal{K}^{\phi}_{\alpha}$ can only be hypothesized whenever a corpus with different annotations $\mathcal{C}^{\phi}_{\beta}$ pertinent to the same phenomenon allows us to infer a $\tilde{\mathcal{K}}^{\phi}_{\beta}$ such that $o \in \tilde{\mathcal{K}}^{\phi}_{\alpha} \cap \tilde{\mathcal{K}}^{\phi}_{\beta}$ (e.g. longer sentences are also deemed more complex on average in the perceived complexity corpus, so length is probably related to complexity in general).

Chapter 2 assumptions can now be summarized in a single statement:

Assumption 4.1 (Learning-driven encodability) A learning process that trains a model M on a corpus $\mathcal{C}_\alpha^\phi$ up to a reasonable accuracy is equivalent to an encoding function that maps ϕ-relevant knowledge contained in $\mathcal{C}_\alpha^\phi$ to M's learned representations.

If Assumption 4.1 is verified, then annotations must be informative, and the model must be able to encode all knowledge present in the corpora relevant to the phenomena. On top of that foundational assumption, three further requirements that are sufficient and necessary for building interpretable learning models able to represent knowledge in a generalizable manner are defined:

Assumption 4.2 (Knowledge-similarity interrelation) Given two corpora $\mathcal{C}_\alpha^\phi, \mathcal{C}_\beta^\phi$ providing different and possibly complementary knowledge about the same phenomenon ϕ and representations R_α^M, R_β^M learned by a model M trained respectively on the two corpora, the more those representations are similar in relation to ϕ, the more ϕ-related shared knowledge is contained in the two corpora. When the two representations are perfectly ϕ-similar, the two corpora share the same ϕ-related knowledge.

Assumption 4.3 (Pertinence-based preponderance) The amount of knowledge $\mathcal{K}_\alpha^\phi$ related to phenomenon ϕ contained in a corpus $\mathcal{C}_\alpha^\phi$ that explicitly encodes some knowledge about ϕ is always larger than the amount of knowledge relative to ϕ contained in any corpus $\mathcal{C}_\beta^{\phi'}$ which explicitly covers a different phenomenon ϕ' by means of its annotations β.

Assumption 4.4 (Knowledge-similarity transitivity) Given three corpora $\mathcal{C}_\alpha^\phi, \mathcal{C}_\beta^\phi, \mathcal{C}_\gamma^\phi$ providing different views over the same phenomenon ϕ and representations $R_\alpha^M, R_\beta^M, R_\gamma^M$ learned by a model M trained on each one of them respectively, if a pair of those representations has higher ϕ-similarity than another, then the respective pair of corpora also have a larger amount of shared ϕ-related knowledge and vice versa.

The experimental section of this chapter is aimed at testing whether those requirements are satisfied by ALBERT. Assumption 4.2 enables us to use representational similarity measures to evaluate our corpora's latent knowledge related to linguistic complexity. In particular, RSA and PWCCA will be used respectively as naive and more advanced approximations of ς that evaluate representations' distance in the n-dimensional space across multiple linguistic structures.

The first step in this verification process involves comparing representations learned by ALBERT models trained on PC, ET, and RA against those of Base. Since the base model was exposed to a general MLM pre-training, without having access to any complexity-related annotation, it can be hypothesized that *the three complexity-trained models had access to more*

complexity-related information during training (Assumptions 4.1 and 4.3), *and thus learned representations that are closer together in similarity terms than those of Base* (Assumption 4.2). The other perspective involves evaluating how different views related to the same phenomenon are captured. While perceived complexity annotations and gaze metrics are at the antipodes of the processing spectrum (see Figure 1.1), they should logically contain more complexity-related shared information than readability categories since they are both related to the reader's viewpoint, while RA captures the writer's perspective. If Assumption 4.4 is verified, then it can be hypothesized that *ALBERT-PC and ALBERT-ET learned representations closer together in similarity terms than those of the ALBERT-RA model*.

Before moving to the experiments, two crucial aspects should be highlighted. First, corpus size was abstracted away from the verification process despite being commonly known to be an essential factor in shaping neural network training effectiveness. In particular, we should be aware that the size imbalance across available corpora can be a significant source of error in the evaluation process. Secondly, sentence-level training objectives are used for PC and RA tasks, while ALBERT-ET is trained on token-level annotations.[3] If, on the one hand, this difference in training approaches can act as an additional confounder when evaluating requirements, from another perspective, it can provide us with some information relative to the generalization abilities of ALBERT beyond task setup.

4.2 Experimentsl Evaluation

This section describes the similarity experiments that have been carried out over model representations across multiple training setups. First, Section 4.2.1 presents the data used to train ALBERT models and evaluate their representational similarity. Then, Section 4.2.2 focuses on validating the assumptions formulated at the beginning of this chapter by evaluating the intra-model similarity across all model pairs. Finally, Section 4.2.3 employs the same similarity approach in an intra-model setting, providing us with some evidence on how linguistic knowledge is encoded hierarchically across ALBERT layers during the training process.

4.2.1 Data

The experiments of this chapter leverage all corpora that were presented in Sections 1.3.1, 1.3.2 and 1.3.3 for fine-tuning the three complexity models whose representations were compared against each other and the Base pre-trained ALBERT. Specifically:

[3] More details on this procedure are provided in Appendix C.

Readability Assessment The OneStopEnglish corpus (Vajjala and Lučić, 2018) is leveraged by splitting each document into sentences and labeling those with the original reading level. A total of 7190 sentences equally distributed across the Elementary, Intermediate, and Advanced levels are used to fine-tune ALBERT-RA in a multiclass classification setting.

Perceived Complexity The English portion of the corpus by Brunato, De Mattei, et al. (2018) was again used to fine-tune ALBERT-PC, following the same preprocessing steps detailed in Section 3.1 of the previous chapter.

Eye-tracking The GECO (Cop et al., 2017), Dundee (Kennedy et al., 2003), ZuCo (Hollenstein, Rotsztejn, et al., 2018) and ZuCo 2.0 (Hollenstein, Troendle, et al., 2020) corpora were merged (Total column of Table 1.4) and used to train the ALBERT-ET model. As opposed to the previous section's sentence-level approach, ALBERT-ET is trained to predict gaze metrics *at token-level* to obtain a fine-grained perspective over the input's complexity and fully exploit the information available through gaze recordings.[4]

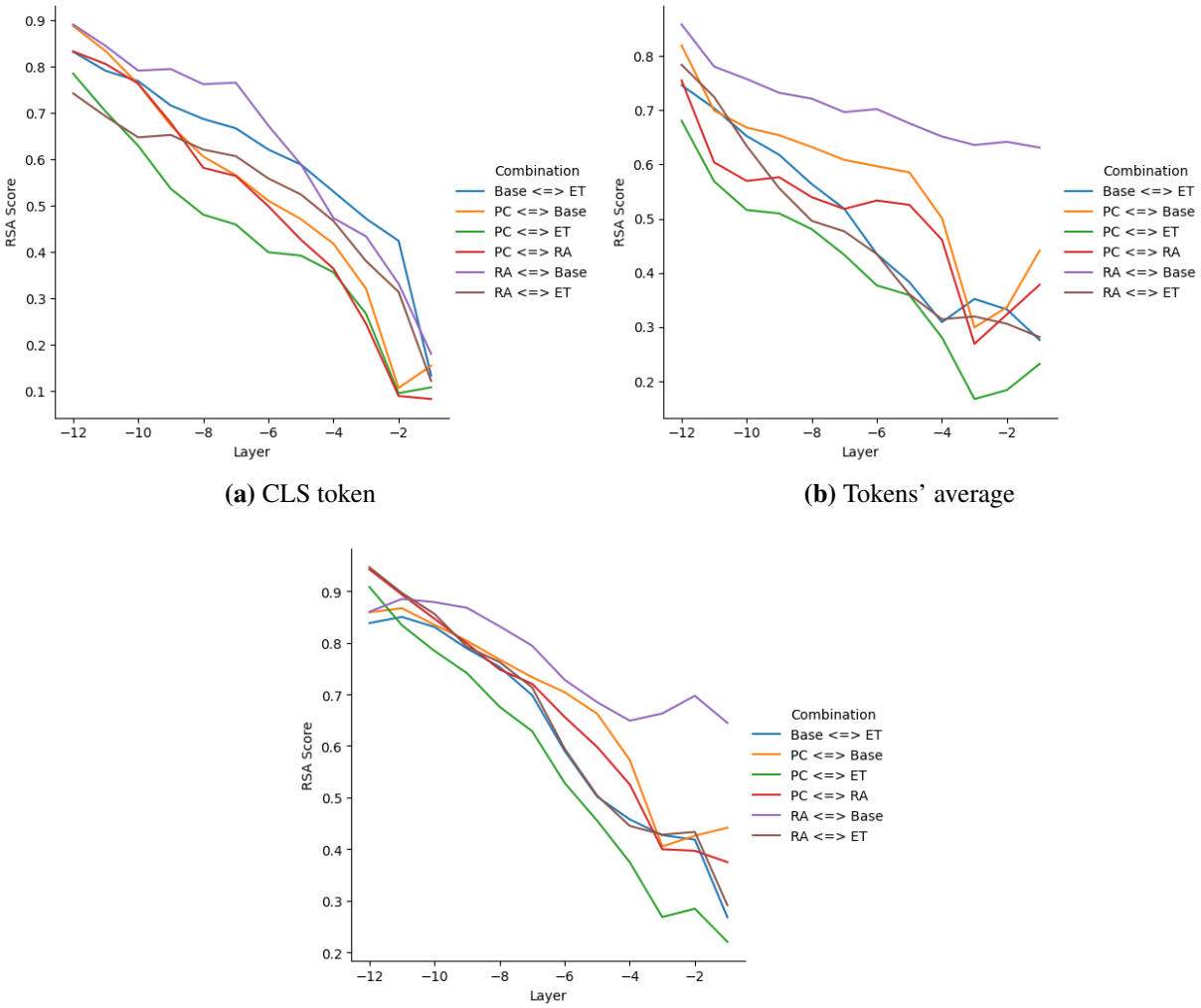
Evaluation All models are evaluated by measuring the similarity of their representations of the Stanford Sentiment Treebank (SST, Socher et al. (2013)). The version of the treebank leveraged for this study contained 11,855 sentences and was selected because the movie review genre is different from all textual genres encompassed by the available corpora (except ZuCo, which represent only a small fraction of the whole set of eye-tracking data used). Sentiment annotations were removed, and only sentences were considered.

4.2.2 Inter-model Representational Similarity

The inter-model similarity is evaluated by comparing layer-wise representations of models trained on different tasks using the same ALBERT architecture. Given the representations produced by two ALBERT models trained on different complexity-related annotations for all the sentences in the SST corpus, their similarity is evaluated using both RSA and PWCCA in three settings:

- **[CLS] token**: Only the sentence-level `[CLS]` initial embedding is considered when evaluating similarity at each layer for all sentences in the SST corpus.
- **Tokens' average**: A sentence-level embedding obtained by averaging all the individual subword embeddings produced by ALBERT is considered when evaluating similarity at each layer for all sentences in the SST corpus.

[4] See Appendix B for additional details on the preprocessing and merging of eye-tracking corpora.



(a) CLS token

(b) Tokens' average

(c) All tokens

Figure 4.1: Inter-model RSA scores across layers for all ALBERT models' combinations. Layer -1 corresponds to the last layer before prediction heads. Higher scores denote stronger inter-model similarity.

- **All tokens**: The subword embeddings produced by ALBERT for all SST sentences are considered when evaluating similarity at each layer, including `[CLS]`, `[SEP]` and regular token embeddings, for all sentences in the SST corpus. In practice, the number of considered embedding was set to a maximum of 50,000 to limit such an approach's computational costs.

Figure 4.1 presents inter-model RSA scores for all model combinations and layers, going from the input layer after initial embeddings (-12) to the last layer before prediction heads (-1).

Given the RSA similarity metric has range $[0, 1]$, it can be observed that representational similarity varies greatly across layers, ranging from very high (~ 0.9) across bottom layers of the models to very low (< 0.1) for top layers. This observation supports the widely accepted claim

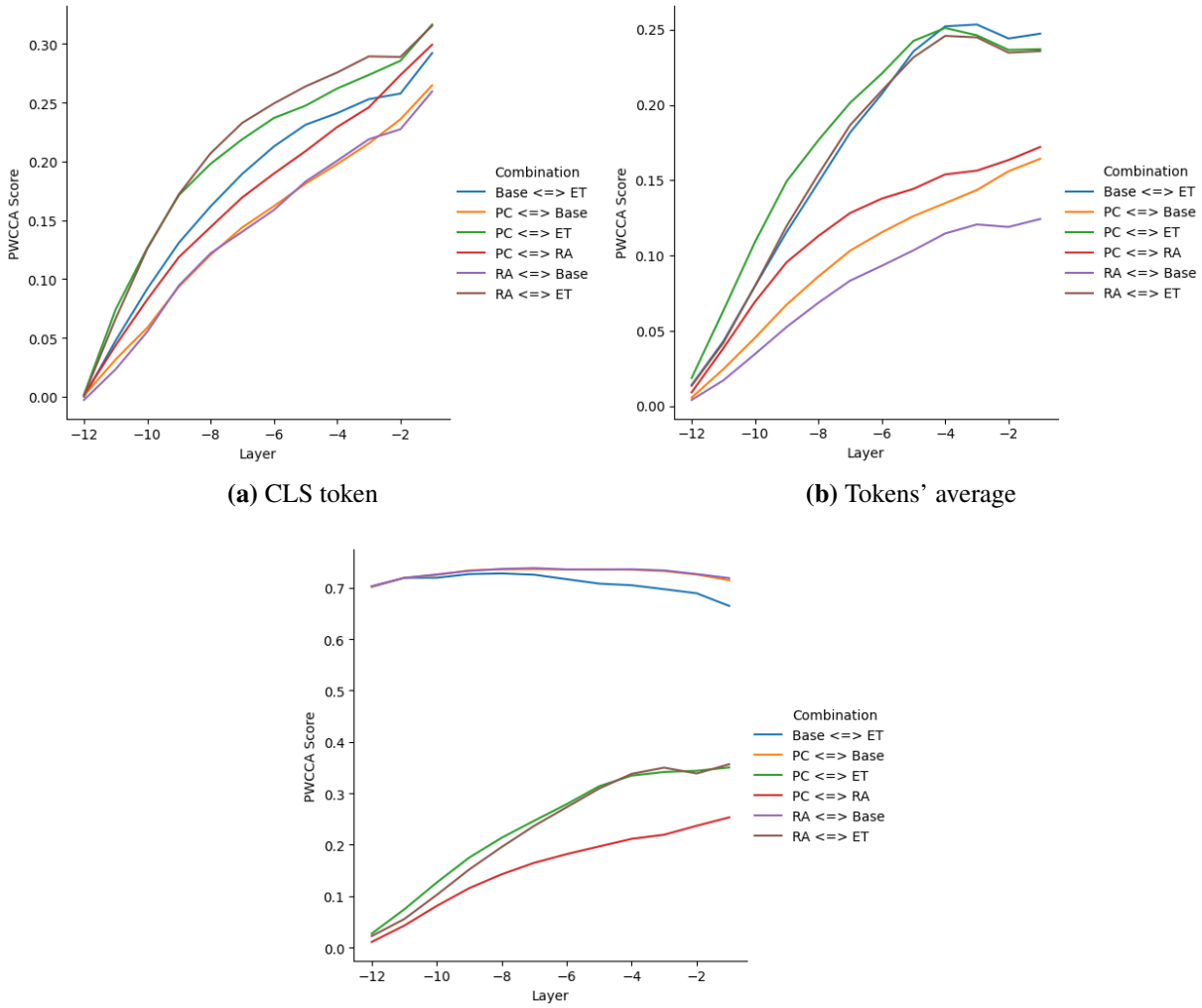
that layers closer to the input in NLMs are almost unaffected by task-specific fine-tuning since they encode low-level properties, while layers closer to prediction heads represent task-related abstract knowledge and tend to diverge rapidly during training.

In settings involving the PC-trained model (yellow, red, and green lines in Figure 4.1) no sharp decrease in similarity is observed across the top layer for all three variations. Conversely, spikes of decreasing similarity are observed for top layers of all other model pairs. While in terms of `[CLS]` all models behave comparably, there is a marked dissimilarity between PC and ET-trained models for top layers when considering all token representations, both with and without averaging (green line in Figures 4.1 a,b). Conversely, RA's `[CLS]` representations behave similarly to the ones of other models, but token representations stay very similar to Base even for top layers, i.e. are slightly affected by fine-tuning (purple line in Figures 4.1 b,c). It can be hypothesized that the RA-trained model cannot collect relevant token-level information since it misses the relative perspective that, as saw in Section 1.3.1, plays a key role for readability assessment. In this case, PC and ET-trained models are the only ones building relevant complexity-related knowledge, but they still tend to diverge in terms of representational similarity.

Figure 4.2 presents PWCCA scores in the exact same setup as Figure 4.1. It does not come as a surprise that scores, in this case, tend to increase while moving towards prediction heads since the PWCCA distance on the *y*-axis represents here a function of representational dissimilarity between different layers. Besides this difference, a sharp contrast in behavior is observed in relation to RSA scores, with generally smaller value ranges ($\sim$ 0.0 to 0.4).

In terms of `[CLS]` representations, (PC, Base) and (RA, Base) are the two closest pairs, while (PC, ET) and (RA, ET) are furthest. This relation can be rationalized if considering that PC and RA-trained models are trained using the `[CLS]` token representation for prediction and have relatively few annotations if compared to the token-level trained ET model. The contrast is even more pronounced when PWCCA distances are measured across token averages (Figure 4.2 b). Here, pairs containing the ET model quickly diverge from the common trend and settle to a shared PWCCA distance for top layers. Finally, the comparison of all individual token representation contradicts previous RSA trends by showing a remarkably consistent divergence from Base representations at all layers for all the three complexity-trained models.

All in all, both RSA and PWCCA suggest an abstraction hierarchy where the closeness of a representation layer to prediction heads is proportional to the magnitude of changes in parameter values during the training process. While RSA similarity highlights a markedly different behavior for the readability-trained model, the more advanced PWCCA method indicates that representations of models trained with similar objectives stay close in parameter space throughout training, regardless of the conceptual proximity phenomena modeled by their loss functions.



(a) CLS token

(b) Tokens' average

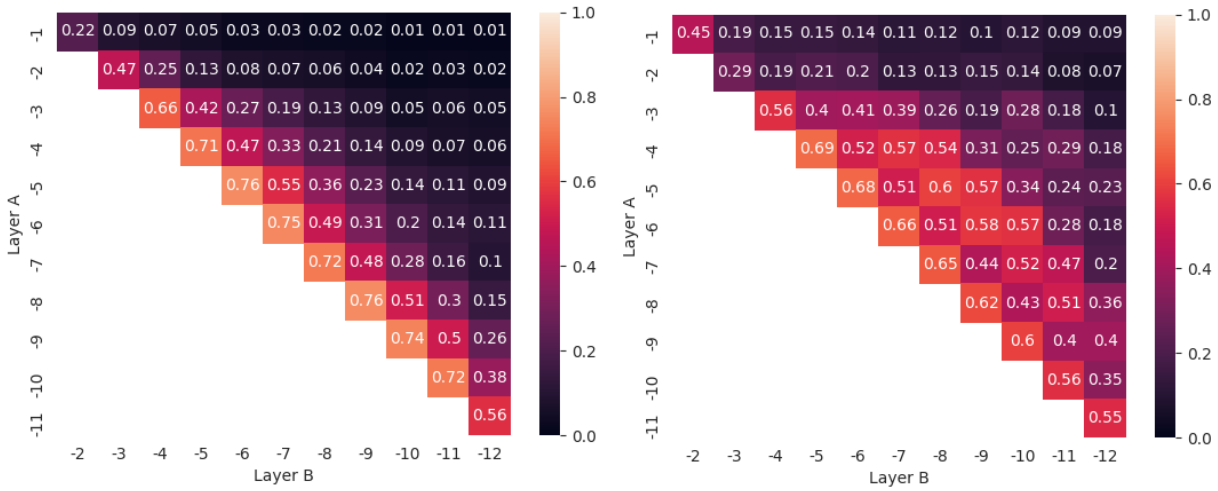
(c) All tokens

Figure 4.2: Inter-model PWCCA distances across layers for all ALBERT models' combinations. Layer -1 corresponds to the last layer before prediction heads. Higher values denote weaker inter-model similarity.

4.2.3 Intra-model Representational Similarity

The intra-model similarity is evaluated in the same setting of the previous section. However, instead of comparing the same layer across two different models, the representations learned by all layer pairs inside the same model are compared using RSA and PWCCA. Again, the three perspectives of `[CLS]`, token's average, and all tokens introduced in the previous chapter are evaluated to understand the shift in representations across layers at different levels of granularity (two sentence-level and one token-level).

Figure 4.3 presents intra-model RSA similarity scores for all layer pairs of the Base model, going from the input layer after initial embeddings (-12) to the last layer before prediction heads (-1). Only the Base model results are presented in this chapter since they are very similar to



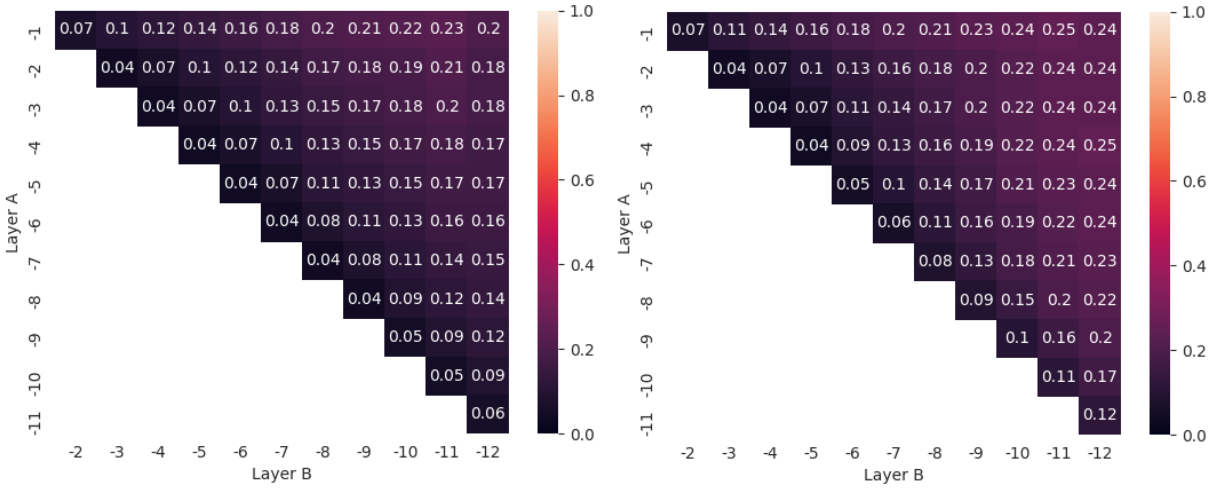
(a) CLS token

(b) Tokens' average

(c) All tokens

Figure 4.3: Intra-model RSA scores across layers' combinations for the pre-trained ALBERT model without fine-tuning (**Base**). Layer -1 corresponds to the last layer before prediction heads. Higher values denote stronger inter-layer similarity.

those produced by fine-tuned models. The latter can be found in Appendix D. The first insight relative to RSA intra-model results is that ALBERT layers tend to learn representations that are generally very similar to those of layers in their neighborhood, especially for layers found at the center and close to the input embeddings of the model. While in the case of `[CLS]` similarity scores fall sharply beyond the preceding/following layer for each layer, suggesting a significant variation in the information encoded across the model structure, the high-similarity range is much broader for tokens' average and all tokens representations. It is interesting to note that the top two layers (-1 and -2) are almost always very dissimilar in relation to the rest of the model, which is coherent with the spiking behavior around inter-model scores highlighted



(a) CLS token

(b) Tokens' average

(c) All tokens

Figure 4.4: Intra-model PWCCA distances across layers' combinations for the pre-trained ALBERT model without fine-tuning (**Base**). Layer -1 corresponds to the last layer before prediction heads. Higher values denote weaker inter-layer similarity.

in the previous section. Another interesting observation is that, while `[CLS]` and all tokens' representations are consistently decreasing, the tokens' average representation similarity follows an undulatory behavior across middle layers for all the tested models, with similarity scores dropping and raising while moving away from reference layer. This fact further supports the evidence that token's sentence-level average may better integrate language information from lower layers into high-level representations, as highlighted by Miaschi and Dell'Orletta (2020) in the context of morphosyntactic knowledge.

Figure 4.4 presents PWCCA scores in the exact same setup as Figure 4.3. As in the previous section, the inverse trend in scores here is due to PWCCA being a dissimilarity measure,

and the range of result scores is smaller than the one of RSA. Conversely to the previous setting, [CLS] representations stay closer across layers when their similarity is measured using PWCCA, and there are no significant spikes in score values. The latter finding is coherent with the effect of cross-layer parameter sharing adopted by ALBERT authors. Quoting Lan et al. (2020): "We observe that the transition from layer to layer [in terms of L2 distances and cosine similarity] are much smoother for ALBERT than for BERT. These results show that weight-sharing affects stabilizing network parameters". In the context of [CLS] representations, the lowest layer (-12) appears to be slightly closer to the top layers than the subsequent ones. This fact ultimately supports the intuition that ALBERT is heavily overparametrized, and first-level embeddings already capture much information.

Again for intra-model similarity, PWCCA highlights an abstraction hierarchy inside ALBERT with smoother and generally more reasonable transitions than those showed by RSA. There is no reason to believe that ALBERT adapts its representation hierarchy as a function of its objective since intra-model similarity scores stay approximately the same before and after fine-tuning for all complexity corpora.

4.3 Summary

In this chapter, the representations learned by a neural language model fine-tuned on multiple complexity-related tasks were compared using two widely-used representational similarity approaches. Token and sentence-level representations were compared both considering the same layer across models exposed to different training corpora and different layer pairs contained in the same model. In the first case, the absence of a preponderant similarity between complexity-trained models when compared to the pre-trained one suggests that those models learn their objective by overfitting annotations and without being able to recognize useful primitives that could be recycled throughout complexity tasks. This fact is highlighted in the comparison between perceived complexity and eye-tracking-trained models, where similarity scores of layers close to prediction heads are very different despite the close relationship between the two complexity perspectives. In conclusion, this work strongly supports the claim that representation learning in ALBERT and other neural language models is mainly driven by training biases like task granularity (token-level vs. sentence-level) that are unrelated to the nature of the task itself. This fact hinders their generalization performances, suggesting that much work still needs to be done beyond language modeling to drive generalizable, hierarchical, and compositional representation learning in models of language.

5 | Gaze-informed Models for Cognitive Processing Prediction

This final experimental chapter aims to study the syntactic generalization capabilities of neural language models by evaluating their performances over atypical linguistic constructions. In particular, architectures pre-trained with masked and causal language modeling are evaluated in their ability to predict garden-path effects on three test suites taken from the SyntaxGym psycholinguistic benchmark. First, the results of previous studies using GPT-2 surprisal to predict garden-path effects are reproduced, and a conversion coefficient is used to evaluate GPT-2 surprisal in terms of human reading times delays. Two neural language models are fine-tuned over gaze metrics from multiple eye-tracking corpora in a multitask token-level setting. Gaze metric predictions on garden-path sentences are evaluated to see whether gaze data fine-tuning can improve garden-path effects prediction. Results highlight how GPT-2 surprisals overestimate the magnitude of MV/RR and NP/Z garden-path effects, and fine-tuning procedures on gaze metrics prediction over typical linguistic structures do not benefit the generalization capabilities of neural language models on out-of-distribution cases like garden-path sentences.

Human behavioral data collected during naturalistic reading can provide useful insights into the primary sources of processing difficulties during reading comprehension. Multiple cognitive processing theories were formulated to account for the sources of such difficulties (see Section 1.4). Notably, **surprisal theory** (Hale, 2001; Levy, 2008) suggests that processing during reading is the direct result of a single mechanism, that is, the shift in readers' probability distribution over all possible parses. To evaluate whether this perspective holds empirically, language models defining a probability distribution over a vocabulary given previous context (RNNs in Elman (1991) and Mikolov, Karafiát, et al. (2010), recently Transformers in Hu et al. (2020)) are commonly used to obtain accurate predictability estimates that can directly be compared to behavioral recordings (e.g. gaze metrics) acting as proxies of human cognitive processing.

A computational model that consistently mimics human processing behaviors would provide strong evidence of cognitive processing's underlying probabilistic-driven nature. For this reason, many studies in the fields of syntax and psycholinguistics have focused on probing the abilities of language models to highlight phenomena related to reading difficulties (Linzen, Dupoux, et al., 2016; Gulordava et al., 2018; Futrell, Wilcox, et al., 2019). Peculiar constructions like garden-path sentences are often used in this context to evaluate the generalization capabilities

of language models for two main reasons. First, garden-path sentences are rare in naturally-occurring text. As such, they represent out-of-distribution examples for any language model trained on conventional data and can be used to test the latter's generalization capabilities. Secondly, researchers nowadays have access to reasonably-sized literature describing the impact of garden-path effects on cognitive processing proxies such as gaze recordings, with articles being often released alongside publicly-available resources for reproducible evaluation (Prasad et al., 2019a; Prasad et al., 2019b) and recently even ad-hoc benchmarks (Gauthier, Hu, et al., 2020).

This final experimental chapter evaluates the ability of neural language models in predicting garden-path effects observed on human subjects, using language modeling surprisal and eye-tracking metrics elicited respectively before and after multitask token-level eye-tracking fine-tuning for garden-path effects prediction. Specifically, an autoregressive (GPT-2, Radford et al. (2019)) and a masked language model (ALBERT, Lan et al. (2020)) are first tested over three garden-path test suites that are part of the SyntaxGym benchmark to evaluate whether their language modeling surprisal before and after eye-tracking fine-tuning (ET) can be used to predict the presence and the magnitude of garden-path effects over disambiguating regions. In particular, GPT-2 and GPT-2 XL results presented in Hu et al. (2020) are reproduced. Finally, the same procedure is repeated using predicted eye-tracking scores predicted by models after fine-tuning instead of language modeling surprisal, following the intuition that an accurate model of gaze measurements should predict such phenomena correctly.

While the usage of surprisal is a common practice for garden-path effect prediction, leveraging eye-tracking scores predicted by a neural language model trained for this purpose is a novel research direction that is deemed interesting as a way to combine the predictive power of modern language models and the strong connection between cognitive processing and gaze metrics. While predicted gaze metrics for garden-path evaluation were used in concurrent studies (van Schijndel et al., 2020), the approach adopted by this work can be regarded as complementary evidence since eye-tracking metrics predictions are produced as results of an end-to-end supervised fine-tuning procedure involving a neural language model rather than being derived from surprisal values through a conversion coefficient. Findings suggest that, while surprisal scores from autoregressive models accurately reflect garden-path structures both before and after fine-tuning, gaze metrics predictions produced by fine-tuned models do not account for the temporary syntactic ambiguity that characterizes such sentences and makes them difficult to process.

Contributions This study validates the performances of standard and gaze-informed Transformed-based neural language models for garden-path effects prediction. In particular:

- It reproduces the GPT-2 performances on garden-path test suites reported by Gauthier, Hu, et al. (2020) and highlights how GPT-2 overestimates reading delays caused by garden-path effects on MV/RR and NP/Z constructions.

- It highlights masked language models' inability to consistently predict garden-path effects, using language modeling surprisal and gaze metrics predictions.

- It introduces a novel gaze metrics multitask token-level fine-tuning approach that, despite being accurate for predicting eye-tracking scores on standard constructions, does not improve models' performances on garden-path effects predictions.

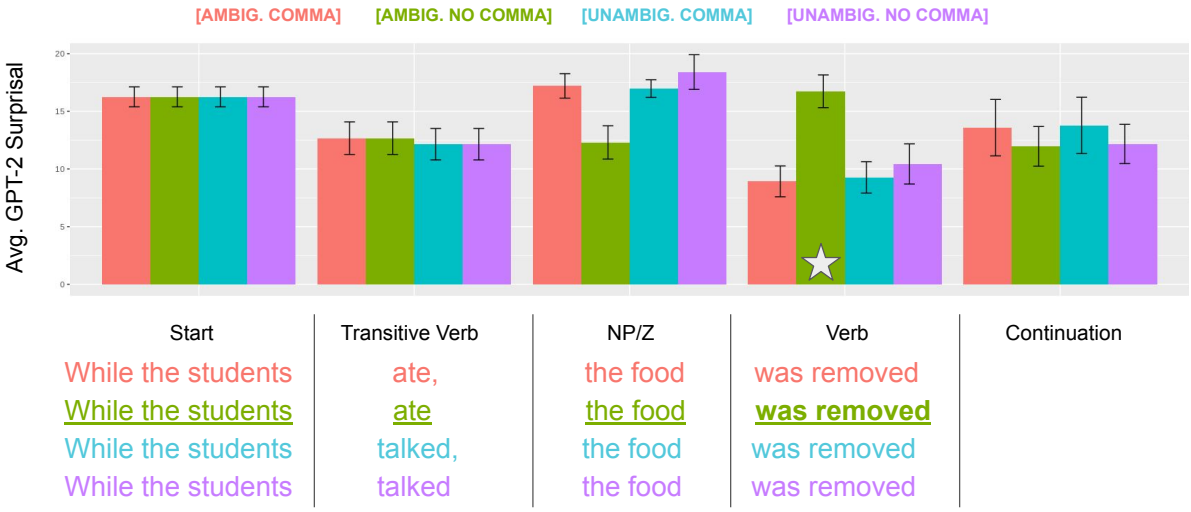
5.1 Experimental Setup

Fine-tuning data As for the gaze metrics model presented in the previous chapter, all eye-tracking datasets presented in Section 1.3.3 were merged and used to fine-tune neural language models using the multitask token-level approach described in Appendix C. Only the training variant without embedding concatenation (referred to as "surprisal" in the appendix) was evaluated on garden-path test suites given comparable modeling performances.

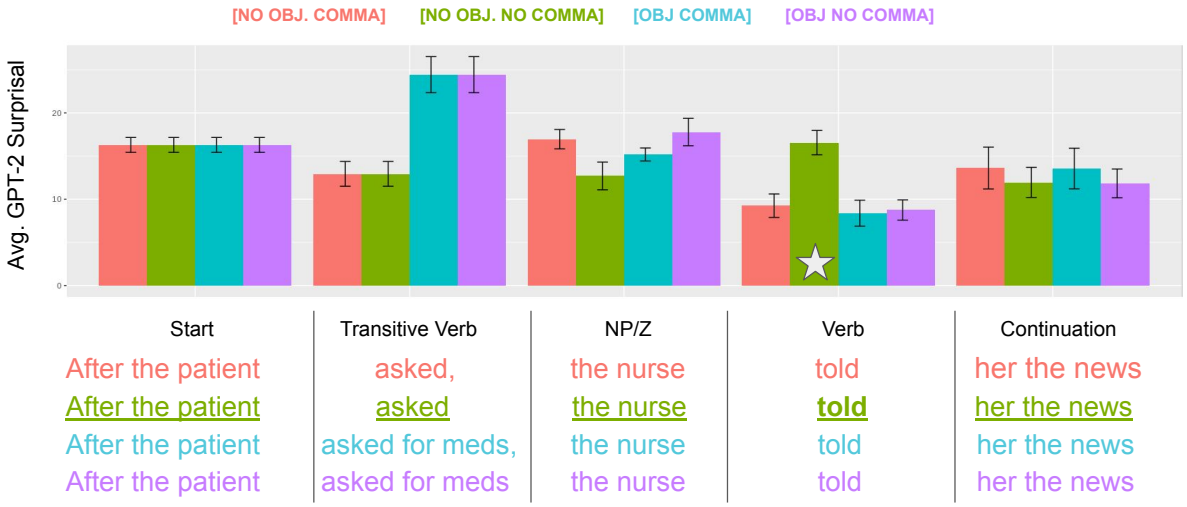
Models Two variants of GPT-2 having respectively 117 million and 1.5 billion parameters are evaluated in terms of surprisal-driven predictability, alongside an ALBERT model with 11 million parameters.[1] Only the small GPT-2 model and the ALBERT model were fine-tuned for gaze metric predictions due to limited computational resources.

Evaluation data SyntaxGym (Gauthier, Hu, et al., 2020) is a recently introduced online platform designed to make the targeted evaluation of language models on psycholinguistic test suites both accessible and reproducible. The MV/RR and NP/Z test suites containing garden paths from Futrell, Wilcox, et al. (2019) are used in the context of this work. The MV/RR test suite consists of 28 groups containing a sentence with a main verb/reduced relative ambiguity and its non-ambiguous rewritings. In comparison, the NP/Z test suites consist of 24 groups containing a sentence with a nominal/zero predicate ambiguity, produced either by a misinterpreted transitive use of a verb (Verb Transitivity) or the absence of an object for the main verb (Overt Object). Examples (3), (4), and (5) from Section 1.4 follow the format used in the three SyntaxGym test suites used in this work.

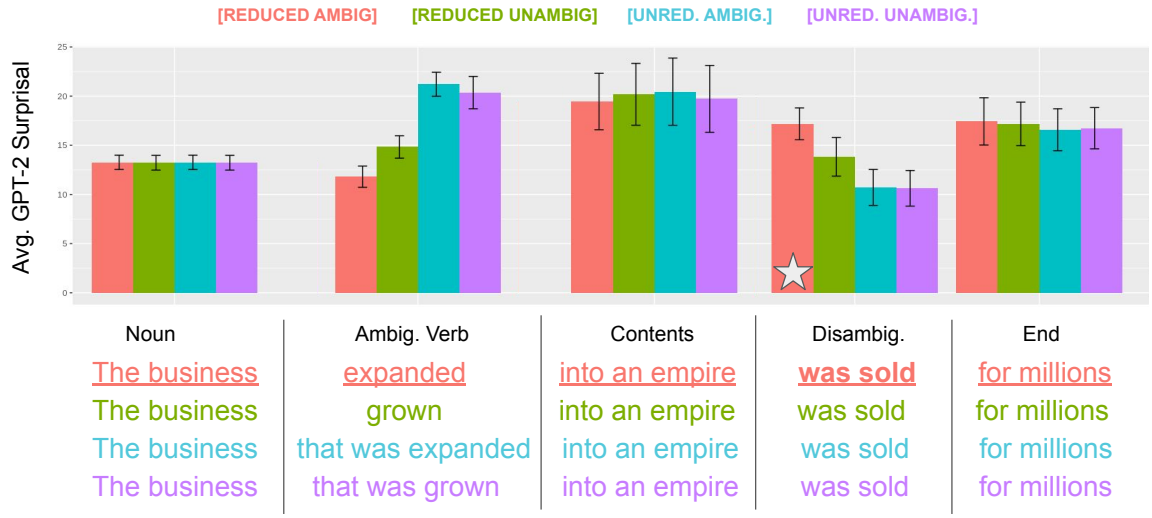
[1] The `gpt2`, `gpt2-xl` and `albert-base-v2` pre-trained models from 🤗 `transformers` (Wolf et al., 2020).



(a) NP/Z Ambiguity (Verb Transitivity)



(b) NP/Z Ambiguity (Overt Object)



(c) MV/RR Ambiguity

Figure 5.1: Average GPT-2 surprisal predictions and examples for the three SyntaxGym test suites. Star marks the garden-path disambiguator (bold in examples), and bars show 95% confidence intervals.

5.2 Experimental Evaluation

For the first part of the experiments, the smallest version of the model GPT-2 is used. Figure 5.1 reproduces the original setting tested by Hu et al. (2020), showing how predictability estimates produced by the model correctly individuate the presence of garden-path effects.[2] Surprisal values are computed using a pre-trained GPT-2 for all tokens in all sentences of the three test suites. Then, those values are aggregated by summing them across all tokens composing a sentence region. For example, for the NP/Z Ambiguity test suite entry shown in example (a) the region "Start" will be associated with the sum of surprisal estimates for all subword tokens in the sequence *While the students*. It is important to note that the four variants of the same sentence have only minimal variations, but only one of those (the underlined one in all examples) is a garden-path sentence. After computing GPT-2 surprisal scores for all regions of all sentences in the test sets, those are averaged region-wise across sentences belonging to the same test set to obtain the three plots presented in Figure 5.1. The star symbol is used to mark the disambiguating region of garden-path sentences, making evident how predictability estimates are significantly lower (i.e., higher surprisal values) for those and correctly predict the presence of a garden-path effect in most settings and for all the three garden-path variants.

5.2.1 Estimating Magnitudes of Garden-path Delays

An important part of evaluating model predictions over garden-path sentences is determining whether the increase in surprisal scores correctly captures the effect's magnitude. van Schijndel et al. (2020) perform this evaluation on RNN language models, finding that they vastly underestimate garden-path effects for MV/RR and NP/Z ambiguities. In their approach, van Schijndel et al. (2020) estimate the surprisal-to-reading-times conversion rate at 2ms per surprisal bit by fitting a linear mixed-effect model on relevant factors (surprisal, entropy, word length, among others) relative to a word and its three preceding words to account for spillover effects. The approach adopted in this work is different in that it stems from the empirical relation between surprisal scores produced by GPT-2 and reading times produced by eye-tracking experiments' participants. Figure 5.2 presents the median values over words for the ratio between gaze metrics recorded by participants and GPT-2 surprisal estimates, with the red cross indicating the average median surprisal-to-metric ratio $C^{\text{metric}}_{\text{corpus}}$ computed across all participants of a corpus. The following formula is used to produce the surprisal-to-reading-times conversion coefficient:

$$C_{S \to RT} = w_1 \cdot C^{\text{FPD}}_{\text{GECO}} + w_2 \cdot C^{\text{FPD}}_{\text{Dundee}} + w_3 \cdot C^{\text{FPD}}_{\text{ZuCo NR}} + w_4 \cdot C^{\text{FPD}}_{\text{ZuCo SR}} + w_5 \cdot C^{\text{FPD}}_{\text{ZuCo 2.0}} \tag{5.1}$$

[2] Similar plots are available on the SyntaxGym website: http://syntaxgym.org/viz/individual

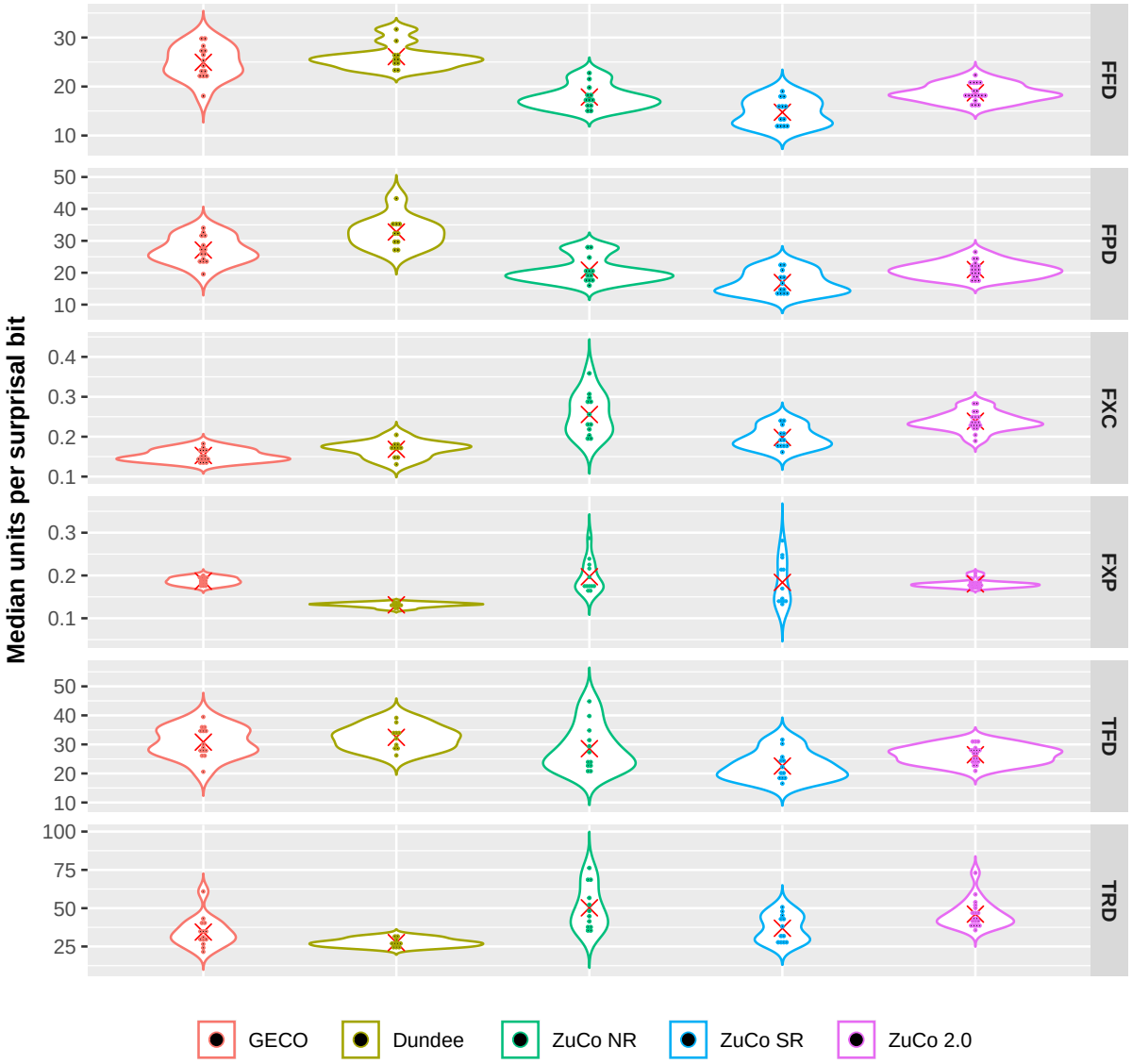


Figure 5.2: Median scores for the ratio between gaze metrics units and GPT-2 surprisal estimates across all participants of all eye-tracking datasets used in this study. The red cross shows the average across participants of a single dataset. Units are in ms for durations, % for FXP, and raw counts for FXC.

with $w = [.4, .45, .05, .05, .05]$ being the weighting coefficients representing the proportion of each corpus' tokens over the total amount of available gaze-annotated tokens.

The resulting value for the conversion coefficient is 27.7, i.e., *each surprisal bit predicted by GPT-2 accounts for roughly 27.7 milliseconds in first pass duration* (30.3ms using TFD). When applied to the average effects predicted by GPT-2 in Figure 5.1, it leads to an estimated delay of roughly 64ms for the MV/RR setting and 166ms and 194ms for the NP/Z Ambiguity and NP/Z Overt Object settings, respectively. These computed delays overestimate the literature's effects: Prasad et al. (2019a) and Prasad et al. (2019b), for example, report an average garden-path

effect of 22ms and 27ms for MV/RR and NP/Z variants, respectively. However, it should be mentioned that precedent studies found higher delays for NP/Z structures: Grodner et al. (2003) find a 64ms delay on disambiguating words, and Sturt et al. (1999)'s delays of 152ms per word are close to the estimates produced by GPT-2 surprisal predictions. Overall, using models' surprisal on gaze-annotated sentences to directly compute a conversion coefficient produces values that correctly identify delays on disambiguating regions and overestimate the magnitude of garden-path effects conversely to what was found by van Schijndel et al. (2020). Even with an adjustment of the conversion coefficient to match MV/RR estimates with Prasad et al. (2019a) findings, the NP/Z effect prediction would still be much larger than the empirically-observed values collected in comparable settings.

5.2.2 Predicting Delays with Surprisal and Gaze Metrics

The other perspective explored in this study is evaluating whether gaze metric predicted by models fine-tuned on eye-tracking corpora annotations can correctly estimate the presence and magnitude of garden-path effects and how they compare to surprisal-driven approaches. Table 5.1 presents the accuracy of multiple pre-trained Transformer-based language models in respecting a set of three conditions taken from Hu et al. (2020) for each SyntaxGym test suite, namely:

$$V_d(b) < V_d(a); \qquad V_d(c) < V_d(a); \qquad V_d(c) - V_d(d) < V_d(a) - V_d(b) \tag{5.2}$$

Where $V_d(a)$ corresponds to the value, either in terms of surprisal or gaze metrics, assigned by a model to the disambiguating region d of sentence a, and a, b, c, d are the same sentence's variants for each test suite presented in examples (3),(4) and (5) of Section 1.4. Accuracy is computed as the proportion of items in the test suite on which the language model's predictions conform to the respective criterion. The first three models (GPT-2, GPT-2 XL, and ALBERT) are the pre-trained variants of the three models presented in Table 5.1 without additional fine-tuning. Instead, the GPT-2 ET and ALBERT ET models correspond to the same GPT-2 and ALBERT models as before after a multitask token-level fine-tuning on gaze metrics for all the aggregated corpora. The top part of Table 5.1 shows the five models' performances while using region-aggregated surprisals as predictors. Focusing on the GPT-2 variants, it can be observed that they all achieve considerably high scores on all evaluated conditions. Conversely, ALBERT masked language models poorly fit the specified criteria. This fact can be intuitively explained by accounting for the different training and evaluation setup used for the two architectures. GPT-2 models are likely to produce high surprisal estimates for garden-path sentences since, processing the input autoregressively and having access only to previous tokens, they incur in the same syntactic ambiguities faced by human readers.

Table 5.1: Results of experiments using surprisal and gaze metrics as predictors for garden-path effects on the three SyntaxGym test suites.

		NP/Z Verb Transitivity			NP/Z Overt Object			MV/RR Ambiguity		
		Cond. 1 [1]	Cond. 2 [2]	Cond. 3 [3]	Cond. 1 [a]	Cond. 2 [b]	Cond. 3 [c]	Cond. 1 [*]	Cond. 2 [†]	Cond. 3 [‡]
Surprisal										
GPT-2		0.96	0.92	0.88	**0.96**	**1**	**1**	**1**	**0.89**	**0.82**
GPT-2 XL		**1**	**0.96**	**1**	**0.96**	**1**	**1**	0.93	0.75	0.75
ALBERT		0.21	0.63	0.58	0.21	0.54	0.46	0.61	0.54	0.38
GPT-2 ET		0.96	0.88	0.79	**0.96**	**1**	0.96	0.96	0.79	**0.82**
ALBERT ET		0.42	0.42	0.58	0.42	0.75	0.62	0.5	0.64	0.64
Eye-tracking metrics										
GPT-2 ET	FFD	0.29	0.38	0.46	0.29	0.54	0.42	0.86	0.57	0.5
	FPD	0.13	0.46	0.67	0.13	0.5	0.46	0.86	0.54	0.36
	FXP	0.38	0.5	0.42	0.42	0.41	0.42	0.71	0.43	0.57
	FXC	0.75	0.5	0.42	0.75	0.63	0.46	0.92	0.46	0.54
	TFD	0.5	0.33	0.46	0.5	0.58	0.75	0.79	0.43	0.39
	TRD	0.67	0.46	0.54	0.63	0.25	0.54	0.29	0.39	0.5
ALBERT ET	FFD	0.67	0.33	0.42	0.42	0.83	0.67	0.68	0.61	0.5
	FPD	0.54	0.41	0.33	0.38	0.79	0.75	0.75	0.57	0.46
	FXP	0.28	0.46	0.29	0.54	0.38	0.63	0.29	0.5	0.43
	FXC	0.63	0.46	0.5	0.38	0.67	0.71	0.86	0.43	0.39
	TFD	0.75	0.38	0.29	0.5	0.88	0.83	0.79	0.61	0.54
	TRD	0.96	0.42	0.42	0.63	0.75	0.5	0.79	0.5	0.57

Description of the evaluated conditions *NP/Z Verb Trans.:* [1] [Ambig. No Comma] > [Ambig. Comma]; [2] [Ambig. No Comma] > [Unambig. No Comma]; [3] [Ambig. No Comma] - [Ambig. Comma] > [Unambig. No Comma] - [Unambig. Comma] *NP/Z Overt Obj.:* [a] [No Obj. No Comma] > [No Obj. Comma]; [b] [No Obj. No Comma] > [Obj. No Comma]; [c] [No Obj. No Comma] - [No Obj. Comma] > [Obj. No Comma] - [Obj. Comma] *MV/RR Ambig.:* [*] [Reduced Ambig.] > [Unred. Ambig.]; [†] [Reduced Ambig.] > [Reduced Unambig.]; [‡] [Reduced Ambig.] - [Unred. Ambig.] > [Reduced Unambig.] - [Unred. Unambig.]

Conversely, ALBERT-like masked language models have access to bidirectional contexts and are not exposed to the ambiguity. It is interesting to observe that while the eye-tracking fine-tuning procedure appears to hamper GPT-2 surprisal performances, it generally improves the ALBERT model's accuracy. This phenomenon may be due to the sequential nature of reading that is being captured by gaze metrics and transferred to the bidirectional ALBERT model as a useful bias for sequential processing. The same procedure performs suboptimally, instead, when associated with an inherently autoregressive model like the GPT-2 decoder

The bottom part of Table 5.1 presents the two ET-trained models' accuracy in matching criteria using predicted gaze metrics. For both GPT-2 and ALBERT, it can be observed that gaze metrics vastly underperform in accuracy terms. We can conclude that, despite the conceptual relation between gaze metrics and predictability observed in humans, the predictions of fine-tuned model cannot generalize to unseen settings, and as such *eye-tracking predictions obtained after a fine-tuning on standard constructions do not appear useful to individuate or estimate the magnitude of garden-path effects*. This observation suggests that fine-tuned models stick to predicting gaze metric values that are the most likely for each specific token, regardless of the surrounding context's ambiguities. Plots in Appendix E present the region-aggregated average scores for all metrics predicted by GPT-2 ET in the same format as before and show how predictions on the disambiguator regions are unaffected by the presence of previous ambiguities.

5.3 Summary

This chapter focused on two perspectives related to the evaluation of neural language models for garden-path effects prediction. First, promising results from previous studies using GPT-2 surprisal to evaluate predictability are reproduced, and language modeling surprisal estimates are converted to reading times using a conversion coefficient. Resulting predictions vastly overestimate the magnitude of garden-path effects in all settings, suggesting the presence of additional mechanisms besides predictability in shaping cognitive processing in the presence of ambiguous constructions like garden-path sentences. This evidence is further supported by the second experimental perspective, in which reading times for garden-path sentences are predicted by models fine-tuned on eye-tracking annotations on corpora containing standard constructions. Results suggest that predicted gaze metrics poorly estimate the presence of garden-path effects over disambiguating regions, suggesting that fine-tuned models are once again incapable of out-of-the-box generalization beyond training settings.

Conclusion

This thesis work adopted a model-driven approach to investigate the relationship between different linguistic complexity perspectives for the English language and study how those are learned and encoded by deep learning models at various abstraction levels.

From the theoretical viewpoint of connecting different complexity perspectives using empirical annotations, Chapter 3 analysis highlighted the strong connection between online/offline complexity metrics and length-related linguistic properties of sentences. The relation was further investigated in length-controlled settings, obtaining similar results across online gaze measurements but different for offline perceived complexity annotations. The overall results identify syntagmatic complexity as the primary source of variation in both offline and online complexity perception for readers. However, they also show how the variety in parts and hierarchical structures contributes differently across different complexity perspectives when sentence length is controlled. Another theoretical aspect supported by Chapter 5 experimental results is the role played by cognitive mechanisms other than predictability in shaping human processing patterns on ambiguous constructions like garden-path sentences. In this context, a computational model that accurately predicts the presence or garden-path effects was used as a psycholinguistic subject to provide predictability annotations on standard and atypical constructions. A surprisal-to-reading-times conversion coefficient was then estimated from gaze annotations and surprisal scores on standard constructions. The resulting reading times were used to highlight how the model widely overestimated the magnitude of garden-path effects, following the methodology of van Schijndel et al. (2020). While results differ significantly from the latter study due to a much larger conversion coefficient, the presence of different accounts for cognitive processing is supported when considering how proportions in predicted magnitudes on different types of constructions do not match the ones reported in recent psycholinguistics literature.

Despite interesting theoretical findings, this work is mostly devoted to interpreting complexity phenomena from a modeling standpoint. Chapter 3 evaluates the encoding of linguistic properties inside neural language models' representations using probing tasks performed before and after model fine-tuning on complexity-related tasks. Results highlighted the emergence of task-related linguistic properties within the model's representations after the fine-tuning process, providing evidence for the relation between models' linguistic skills during training and their performances on morphosyntactically-related tasks. In light of these findings, it can be conjectured that linguistic probes may provide a reasonable estimate of the task-oriented quality of representations for those highly-syntactic tasks. In Chapter 4, the representations learned by neural language

models were compared across layers and fine-tuning tasks using representational similarity approaches. The absence of higher similarity scores between complexity-trained models compared to the pre-trained one suggests that training objectives are learned by overfitting annotations and that learned parameters hardly capture information that could be relevant for multiple complexity-related tasks.

Moreover, task framing and the annotation modalities were observed to play a much larger role in defining representational similarity scores rather than the conceptual similarity between tasks. This fact supports the claim that standard optimization procedures used in deep learning are not suitable for this type of concept-driven learning. Finally, Chapter 5 highlighted the inability of standard neural language models in leveraging syntactic cues to improve prediction in the context of garden-path effects. Models fine-tuned on gaze annotations were tested on garden-path test suites to evaluate whether reading time predictions can perform as well as surprisal in identifying garden-path triggers. Results highlight how models heavily overfit gaze annotation and cannot predict the increase in reading times observed in human subjects despite being exposed to the temporary syntactic ambiguity that characterizes garden-path constructions.

Recent trends in transfer learning have profoundly shaped the last few years of research in NLP, leading to astonishing improvements in almost all language-related tasks, including linguistic complexity prediction. Despite all the hype, the fundamental problem behind all computational linguistics research remains: even the most powerful deep learning models do not "understand" language, and their learned representations are "potentially useful, but incomplete, reflections of the actual meaning" they derive from structural training procedures (Bender et al., 2020). In support of this affirmation, all models leveraged in this study by following closely standard procedures were found lacking in generalization capabilities and hierarchical abstraction, despite their excellent performances on predicting in-domain observations. To conclude with a somewhat cliché affirmation, much work still needs to be done to drive generalizable, hierarchical, and compositional representation learning in language models, enabling proper human-level natural language understanding.

Broader Impact and Ethical Perspectives

The findings described in this thesis work are mostly meta-analytical, and as such, mostly intended to distill theoretical insights and evaluate recent efforts in the natural language processing community. This said, some of the models and procedures described in this work can be clearly beneficial to society. For example, using models trained to predict reading patterns may be used in educational settings to identify difficult passages that can be simplified, improving reading comprehension for students in a fully-personalizable way. This type of technology can

also be applied to domain-specific documents such as juridical or medical reports to identify critical areas that can be adapted to improve layman's understanding. However, it is essential to recognize the potentially malicious usage of such systems. The integration of eye-tracking systems in mobile devices, paired with predictive models presented in this work, could be used to build harmful surveillance systems and advertisement platforms using gaze predictions for extreme behavioral manipulation. Moreover, multiple individuals' gaze data could be leveraged by autonomous systems to enforce discriminatory practices towards neurodiverse subjects in hardly-detectable ways. In terms of research impact, the experiments presented in this work may provide useful insights into the behavior of neural language models for researchers working in the fields of interpretability in NLP and computational psycholinguistics.

Future Directions

In conclusion, multiple paths to improve and extend the scope of this work were identified during the experimental process, and will be left here as a final note for my future self and for anyone interested in pushing forward research in fields related to this thesis' topics.

- Self-training has recently proven to be very effective for compensating the lack of large labeled datasets in the context of acceptability and complexity prediction (Sarti, 2020). In light of these results, it would be interesting to evaluate whether self-training could also improve the performances and generalization of models used for gaze metrics prediction.

- Evaluate whether gaze-trained neural language models having undergone a *cloze distillation process* (Eisape et al., 2020), combining intuitions from masked language modeling and knowledge distillation (Hinton et al., 2015), would produce better results for modeling out-of-distribution garden-path phenomena compared to the somewhat naive approach adopted in this study.

- Incorporating gaze metrics prediction in the training objectives of learning models can be interesting to account for human cognitive biases during reading. The crucial aspect is how to get a sufficient amount of annotated data to make this idea scalable for modern language models' pre-training needs. In this regard, it could be interesting to test the approach by Hollenstein and Zhang (2019) where mean gaze scores are averaged for each type across annotators, effectively providing a way to label input sentences with robust gaze information in an unsupervised manner.

- Since eye-tracking metrics are complexity signals with free human supervision, it could be possible to leverage those for simplification and other related tasks in an iterative learning-from-human-feedback paradigm similar to the one described in Stiennon et al. (2020).

- It should in principle be possible to use human processing data as a replacement for the self-attention computation. The dot product critically bounds the computational efficiency of attention-based models, and fixed attention has been shown to have a limited negative impact on final results while making inference much faster (Tay et al., 2020). Fixing attention weights using human attention, as measured by eye-tracking metrics, can be an exciting perspective to explore in this context. This idea can be thought of as an application of human attention regularization of LSTM attentional networks for various tasks proposed in Barrett, Bingel, Hollenstein, et al. (2018) to Transformers networks.

- Would explicitly embedding complexity in the learning process of language models favor hierarchical abstraction? In this perspective, it would be exciting to evaluate whether a model trained on easy-to-hard sentences following language acquisition insights would encode different knowledge in terms of linguistic structures, concept abstraction, and allowances.

- Finding better ways to instill useful inductive biases into learning models, especially for syntax-heavy downstream tasks. Concrete examples following this direction may use parsing as a complementary task to keep top-level representations sensible to syntactic changes, as tested in Glavas et al. (2020) for natural language understanding, or use hybrid symbolic-neural models to represent syntax as in Zanzotto et al. (2020).

Appendices

A | Linguistic Features

The following list of features was used in the context of Chapter 3 experiments and is a summary of the full set of features presented in Brunato, Cimino, et al. (2020):

A.1 Raw Text Properties and Lexical Variety

- **Sentence length** (*n_tokens*): Length of the sentence in terms of number of tokens.
- **Word length** (*char_per_tok*): Average number of characters per word in a sentence, excluding punctuation.
- **Type/Token Ratio for forms and lemmas** (*ttr_form, ttr_lemma*): Ratio between the number of lexical types and the number of tokens within a sentence.

A.2 Morpho-syntacting Information

- **Distribution of grammatical categories** (*upos_dist_*, xpos_dist_**): Percentage distribution in the sentence of the 17 core part-of-speech categories present in the Universal POS tagset (adjective, adverb, interjection, noun, proper noun, verb, adposition, auxiliary, coordinating conjunction, determiner, numeral, particle, pronoun and subordinating conjunction, punctuation, and symbols).
- **Lexical density** (*lexical_density*): Ratio of content words (verbs, nouns, adjectives, and adverbs) over the total number of words in a sentence.
- **Inflectional morphology** (*aux_mood_*, aux_tense_*): Percentage distribution in the sentence of a set of inflectional features (*Mood, Number, Person, Tense and Verbal Form*) over lexical verbs and auxiliaries of each sentence.

A.3 Verbal Predicate Structure

- **Distribution of verbal heads** (*vb_head_per_sent*): Number of verbal heads in the sentence, corresponding to the number of main or subordinate clauses co-occurring in it.

- **Distribution of verbal roots** (*dep_dist_root*): Percentage of verbal roots out of the total sentence roots.
- **Verb arity** (*verb_arity*): Average number of dependency links sharing the same verbal head per sentence, excluding punctuation and copula dependencies.

A.4 Global and Local Parsed Tree Structures

- **Syntactic tree depth** (*parse_depth*): Maximum syntactic tree depth extracted for the sentence, i.e., the longest path in terms of dependency links from the root of the dependency tree to some leaf.
- **Average and maximum length of dependency links** (*avg_links_len, max_links_len*)
- **Number and average length of prepositional chains** (*n_prep_chains, prep_chain_len*), with the latter expressed in number of tokens.
- **Subject-object ordering** (*subj_pre, subj_post, obj_pre, obj_post*): Relative order of the subject and object arguments with respect to the verbal root of the clause in the sentence.

A.5 Syntactic Relations

- **Distribution of dependency relations** (*dep_dist_**): Percentage distribution of the 37 universal relations in the UD dependency annotation scheme.

A.6 Subordination Phenomena

- **Distribution of main and subordinate clauses** (*princ_prop_dist, sub_prop_dist*): Percentage distribution of main vs subordinate clauses in the sentence.
- **Relative ordering of subordinates** (*sub_pre, sub_post*): As for subjects and objects, whether the subordinate occurs in pre-verbal or post-verbal position in the sentence.
- **Average length of embedded subordinates** (*sub_chain_len*): Average length of subordinate clauses recursively embedded into each other to form a subordinate chain.

Readers are referred to the original paper by Brunato, Cimino, et al. (2020) and the Profiling-UD webpage[1] for additional details on linguistic features.

[1] http://linguistic-profiling.italianlp.it

B | Precisions on Eye-tracking Metrics and Preprocessing

Table B.1: Eye-tracking mappings from dataset-specific fields to the shared set of metrics.

Metrics	Dundee	GECO	ZuCo 1 & 2
First fix. dur. (FFD)	First_fix_dur	FIRST_FIXATION_DURATION	FFD
First pass dur. (FPD)	First_pass_dur	GAZE_DURATION	GD
Fix. prob. (FXP)	Fix_prob	¬ WORD_SKIP	FXC > 0
Fix. count (FXC)	nFix	FIXATION_COUNT	FXC
Tot. fix. Dur. (TFD)	Tot_fix_dur	TOT_READ_TIME	TRT
Tot. Regres. Dur. (TRD)	Tot_regres_from_dur	GO_PAST - SEL._GO_PAST	GPT - GD

Univocal gaze metrics conversion Table B.1 present the conversion scheme used to obtain a unified set of eye-tracking metrics from different corpora annotations. This method follows closely the approach adopted by Hollenstein and Zhang (2019). While the mapping is straightforward for shared metrics, the TRD metric needs to be computed for GECO and ZuCo. For GECO, the difference between go-past time (i.e. total time elapsed between the first access of a word boundary and the first access of subsequent words, including regressions) and its selective variant (i.e. go-past time only relative to the specific word, without accounting for regressions) gives an exact conversion to regression duration. Instead, in the ZuCo case, an approximate conversion using gaze duration (i.e. first pass duration) instead of selective go-past time is used since selective go-past time is not provided. ZuCo's TRD estimate should be deemed an upper bound for regressions' duration since gaze duration is always smaller than the selective go-past time when regressions are present and is precisely equal to it in the complete absence of regressions.

Averaging across participants Gaze metrics are averaged across participants for all experiments of this thesis work. Metrics missing for some participants due to skipping are replaced with the lowest recorded value across participants for that word before averaging. This procedure is preferred to zero-filling missing values since the latter produces significant drops in metrics associated with tokens skipped by multiple participants, making averaged values inconsistent with empirical observations.

C | Multi-task Token-level Regression for Gaze Metrics Prediction

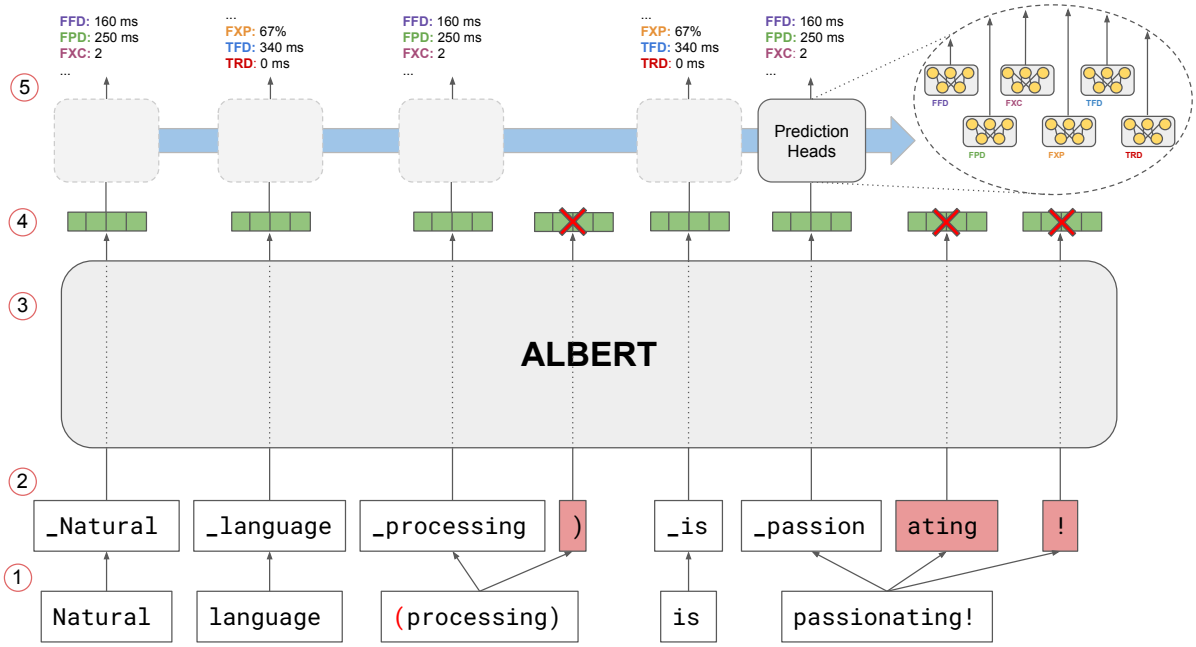


Figure C.1: Multi-task token-level regression on eye-tracking annotations. Preceding punctuation is removed (1), and the sentence is tokenized while keeping track of non-initial tokens (2). Embeddings are fed to the ALBERT model (3), and non-initial representations are masked to ensure a one-to-one mapping between labels and predictions (4). Finally, task-specific prediction heads are used to predict gaze metrics in a multitask setting with hard parameter sharing (5).

A multitask token-level regression fine-tuning approach was adopted throughout this study to predict eye-tracking metrics using neural language models. This novel approach's choice stems from the fact that the regression task of predicting gaze metrics is inherently word-based given the granularity of eye-tracking annotations and that different gaze metrics provide complementary viewpoints over multiple stages of cognitive processing and can as such be modeled more precisely in a multitask learning setting. Figure C.1 presents the model's training and inference procedure, closely matching other approaches used to train neural language models for sequence tagging tasks like POS tagging and named entity recognition.

The most defining detail in the procedure is the need to preserve an exact one-to-one mapping between input words and gaze metrics annotations, which is non-trivial in light of subword tokenization approaches that represent nowadays the *de facto* standard for training modern neural language models. To enforce such mapping, two steps are taken. First, all initial

punctuation (e.g. the open parenthesis before *processing* in Figure C.1 example) is removed to make the initial subword token for that word (i.e. the one preceded by whitespace) equal to the word's first characters. Then, all non-initial subword tokens are identified in step (2), and their respective embeddings are masked in step (4) before passing the remaining initial embeddings (one per whitespace-tokenized word at this point, as for gaze metrics) to the set of prediction heads responsible for inferring individual gaze metrics. While this procedure can be regarded as suboptimal since not all learned representations are used for prediction, it is essential to remember that all the embeddings produced by attention-based neural language models are contextualized and encode information about the entire sentence and surrounding context to some extent. In this sense, initial token embeddings can be trained in this setting to predict gaze metrics relative to the whole word, effectively bypassing the issues about information loss raised by the masking procedure.

Another important detail in the training and inference procedure is the standardization of metrics, which plays a key role in this setup due to the different ranges of different metrics (e.g. fixation probability is always defined in the interval $[0, 1]$, while gaze durations are integers in the scale of hundreds/thousands of milliseconds). Specifically, considering the set X of values assumed by a specific metric for all tokens in the eye-tracking datasets, the average μ_X and standard deviation σ_X of those values are computed, and each value is transformed as:

$$X'_i = \frac{X_i - \mu_X}{\sigma_X} \tag{C.1}$$

to produce a new range X' with average equal to 0 and standard deviation equal to 1. Predicted values are then reconverted to the original scale as $X_i = (X'_i \cdot \sigma_X) + \mu_X$ when performing inference, and training and testing metrics are computed on each metric's original scale.

Spillover concatenation Cognitive processing literature reports evidence of reading times for a word being shaped not only by the predictability of the word itself but also by the predictability of the words that precede it (Smith et al., 2013) in what is commonly referred to as the *spillover effect* (Mitchell, 1984). The existence of spillover has important implications in the context of this gaze metrics prediction approach since the embeddings for a single word may not contain enough information to predict the influence of preceding tokens in shaping reading behaviors. Notably, van Schijndel et al. (2020) include the surprisal of the three previous words in a mixed-effect model used to estimate a surprisal-to-reading-times conversion coefficient. While it can be hypothesized that in this approach, the usage of contextualized word embeddings can automatically account for this type of interaction, the effect of leveraging preceding tokens for the current token's metric prediction is assessed to confirm this hypothesis. A new procedure defined as *spillover concatenation* is introduced for this purpose, in which token embeddings

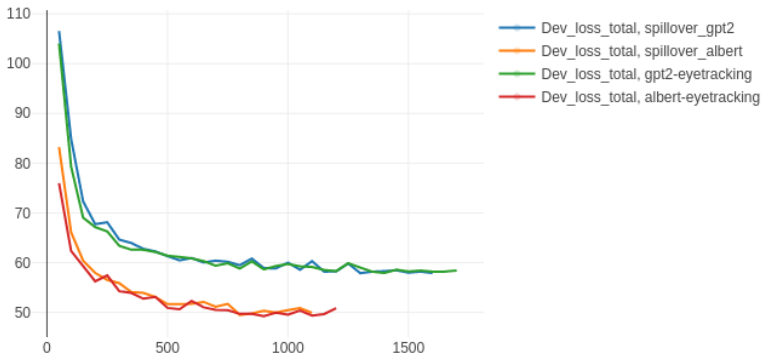


Figure C.2: Validation total loss for GPT-2 and ALBERT over a split of the eye-tracking merged corpora with and without spillover concatenation. Model predictive performances were comparable across training and testing for the two models.

are augmented by performing a rolling concatenation of the n preceding embeddings before feeding the final representation to prediction heads. Initial tokens are padded with 0 vectors to match the fixed size defined by embedding size and the n parameter. For example, using spillover concatenation with $n = 3$ within a BERT model with a hidden size of 768 involves having prediction heads taking input size of $768 \cdot (3+1) = 3072$, the size of the token embedding for which gaze metrics should be predicted plus the size of the three preceding token embeddings. In this way, information about preceding tokens is explicitly included at prediction time.

Figure C.2 shows the validation losses during training for the two models used in the experiments of Chapter 5 with their counterparts using spillover concatenation. Model performances are not positively influenced by introducing the concatenation technique and remain very similar for both architectures.

Model performances Table C.1 presents the test performances of ALBERT and GPT-2 models trained with and without the spillover concatenation approach on the merge of all eye-tracking corpora. The top two rows present descriptive statistics about extreme values, the mean and standard deviation in annotations averaged across participants for each metric. It is interesting to observe that the maximum value observed for first pass duration (FPD) is higher than the one for total fixation duration (TFD). While this situation would not be possible in practice due to first pass duration being included in total reading times, it reminds us about the approximate nature of our filling-and-averaging procedure described in Appendix B. Comparing results to those of Table 3.2, where gaze metrics were modeled at the sentence level, we observe much

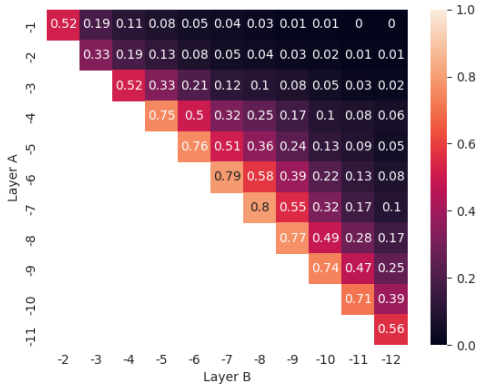
worse results in terms of explained variance for both models: while fixations and first pass duration (FXC, FXP, FPD) are generally well modeled, worse results are obtained for first and total fixation durations (FFD, TFD), and in particular for the duration of regression (TRD). These results can be attributed to the merging of different corpora that, being annotated by different participants, present very different properties, as shown in Table 1.4 and Figure 5.2. While on the one hand, this choice harms modeling performances, on the other hand, it provides us with more representative results for the general setting.

Table C.1: Descriptive statistics and model performances for the merged eye-tracking training corpus. Model scores are in format $\text{RMSE}_{\text{MAX}}|R^2$, where RMSE is the root-mean-squared error and MAX is the max error for model predictions.

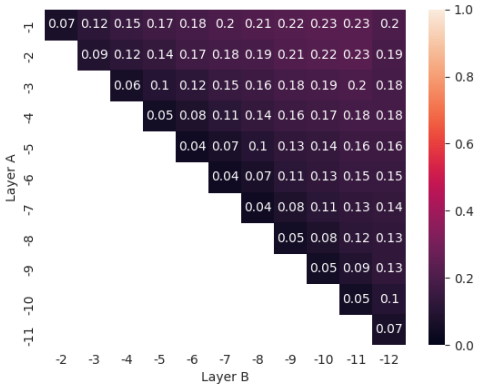
	FFD	**FPD**	**FXP**	**FXC**	**TFD**	**TRD**
min-max value	$0-986$	$0-2327$	$0-1$	$0-8.18$	$0-1804$	$0-4055$
$\mu\|\sigma$ statistics	$162\|50$	$188\|86$	$.56\|.27$	$.85\|.53$	$206\|87$	$90\|122$
ALBERT	$41_{78}\|.33$	$61_{121}\|.50$	$.17_{.32}\|.60$	$.31_{.62}\|.66$	$65_{132}\|.44$	$110_{207}\|.19$
ALBERT Spillover	$41_{78}\|.33$	$61_{122}\|.50$	$.17_{.33}\|.60$	$.31_{.62}\|.66$	$65_{132}\|.44$	$110_{208}\|.19$
GPT-2	$44_{83}\|.23$	$68_{136}\|.37$	$.18_{.35}\|.56$	$.36_{.70}\|.54$	$74_{149}\|.28$	$115_{222}\|.11$
GPT-2 Spillover	$43_{83}\|.26$	$68_{135}\|.37$	$.19_{.35}\|.50$	$.36_{.70}\|.54$	$73_{146}\|.30$	$116_{220}\|.10$

In general, better performances are observed for the masked language model ALBERT, suggesting the importance of having access to bidirectional context for gaze metrics prediction. Results present additional evidence supporting the superfluity of the spillover concatenation procedure, which was henceforth dropped in the context of Chapters 4 and 5's experiments. Although good scores in terms of average and maximal errors are observed for all metrics, the relatively low R^2 seem to suggest that large margins of improvement are still available in the context of gaze metrics predictions with neural language models.

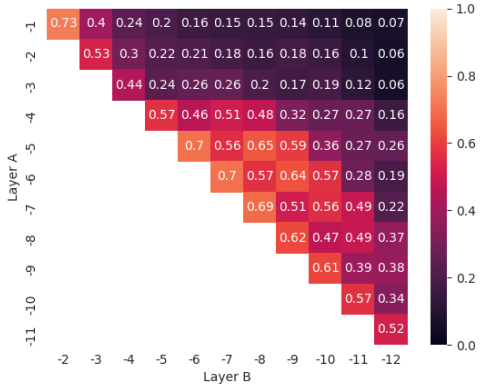
D | Intra-model Similarity for All Models



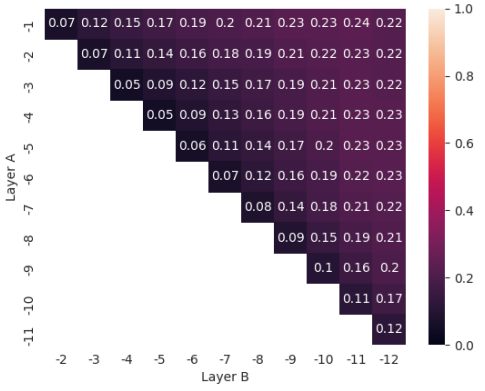
(a) RSA score, CLS token



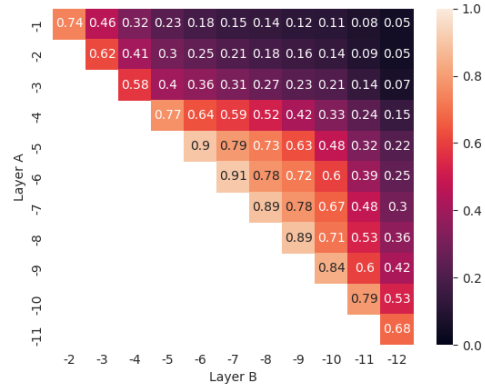
(b) PWCCA distance, CLS token



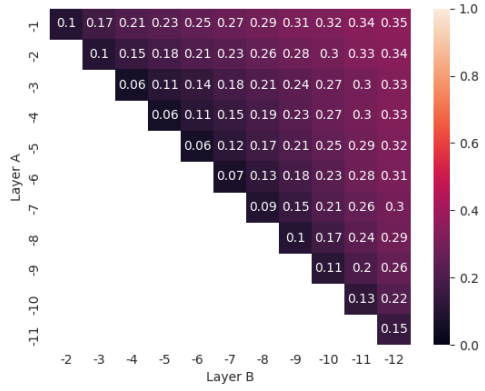
(c) RSA score, tokens' average



(d) PWCCA distance, tokens' average

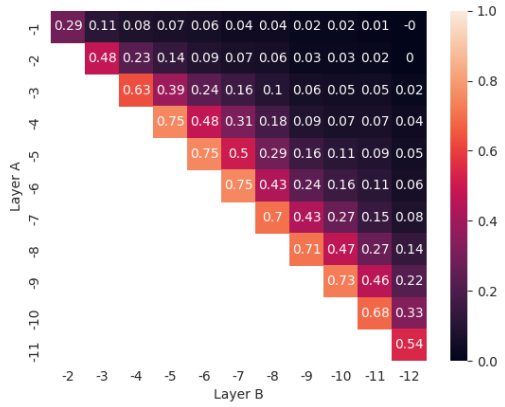


(e) RSA score, all tokens

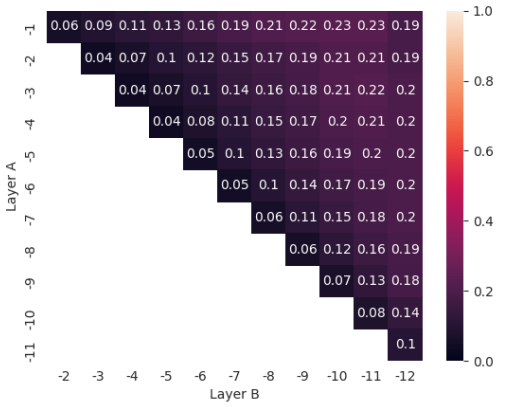


(f) PWCCA distance, all tokens

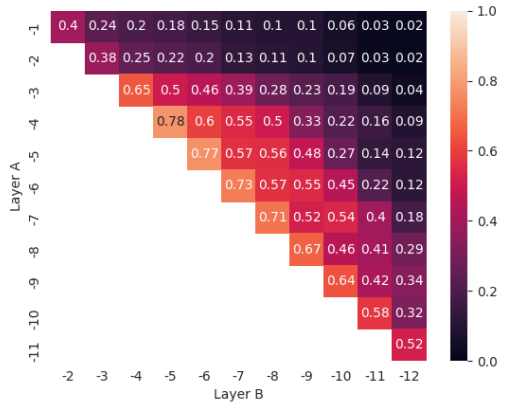
Figure D.1: Intra-model RSA and PWCCA scores across layers' combinations for the ALBERT model fine-tuned on perceived complexity (**PC**). Layer -1 is the last layer before prediction heads.



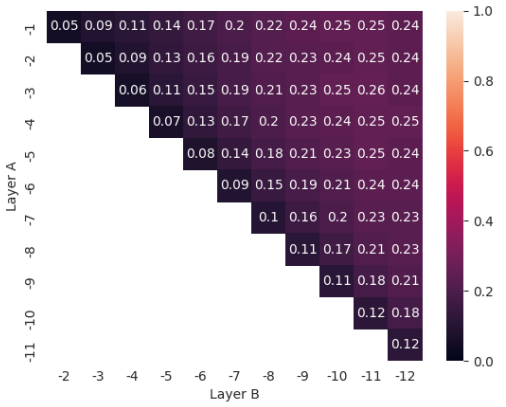
(a) RSA score, CLS token



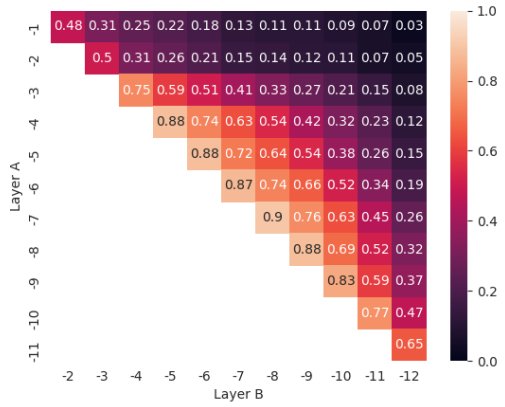
(b) PWCCA distance, CLS token



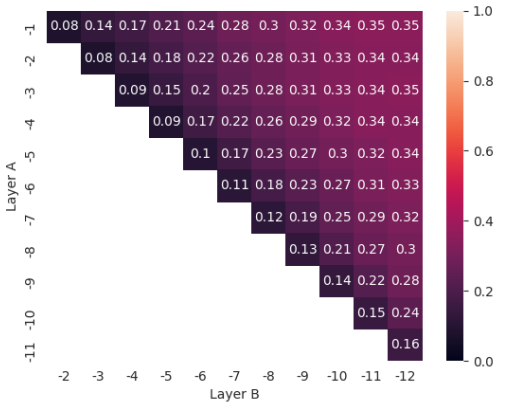
(c) RSA score, tokens' average



(d) PWCCA distance, tokens' average

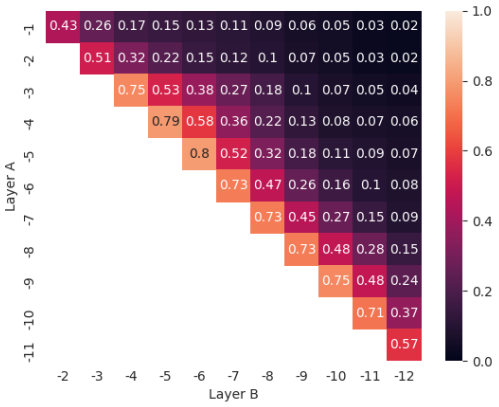


(e) RSA score, all tokens

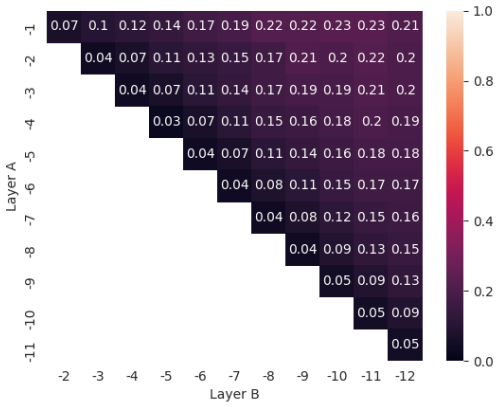


(f) PWCCA distance, all tokens

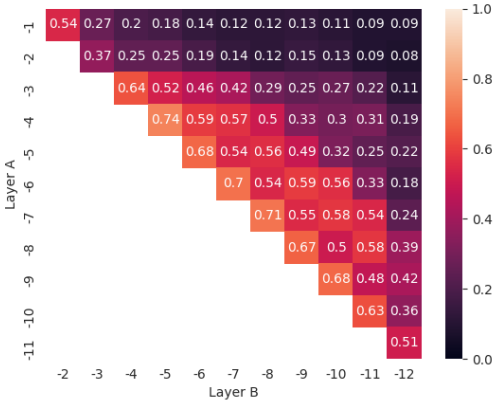
Figure D.2: Intra-model RSA and PWCCA scores across layers' combinations for the ALBERT model fine-tuned in parallel on gaze metrics (**ET**). Layer -1 corresponds to the last layer before prediction heads.



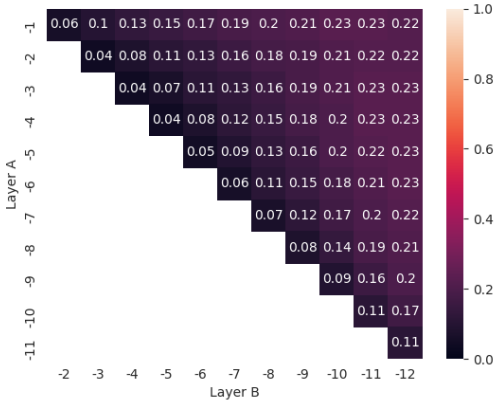
(a) RSA score, CLS token



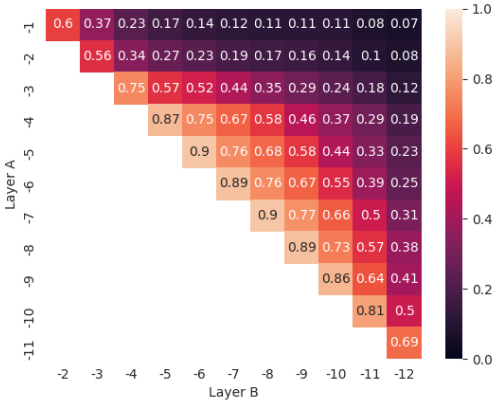
(b) PWCCA distance, CLS token



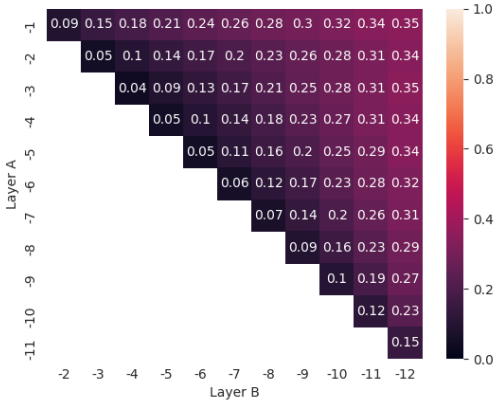
(c) RSA score, tokens' average



(d) PWCCA distance, tokens' average



(e) RSA score, all tokens



(f) PWCCA distance, all tokens

Figure D.3: Intra-model RSA and PWCCA scores across layers' combinations for the ALBERT model fine-tuned on readability assessment annotations (**RA**). Layer -1 corresponds to the last layer before prediction heads.

E | Gaze Metrics Predictions for Garden Path Sentences

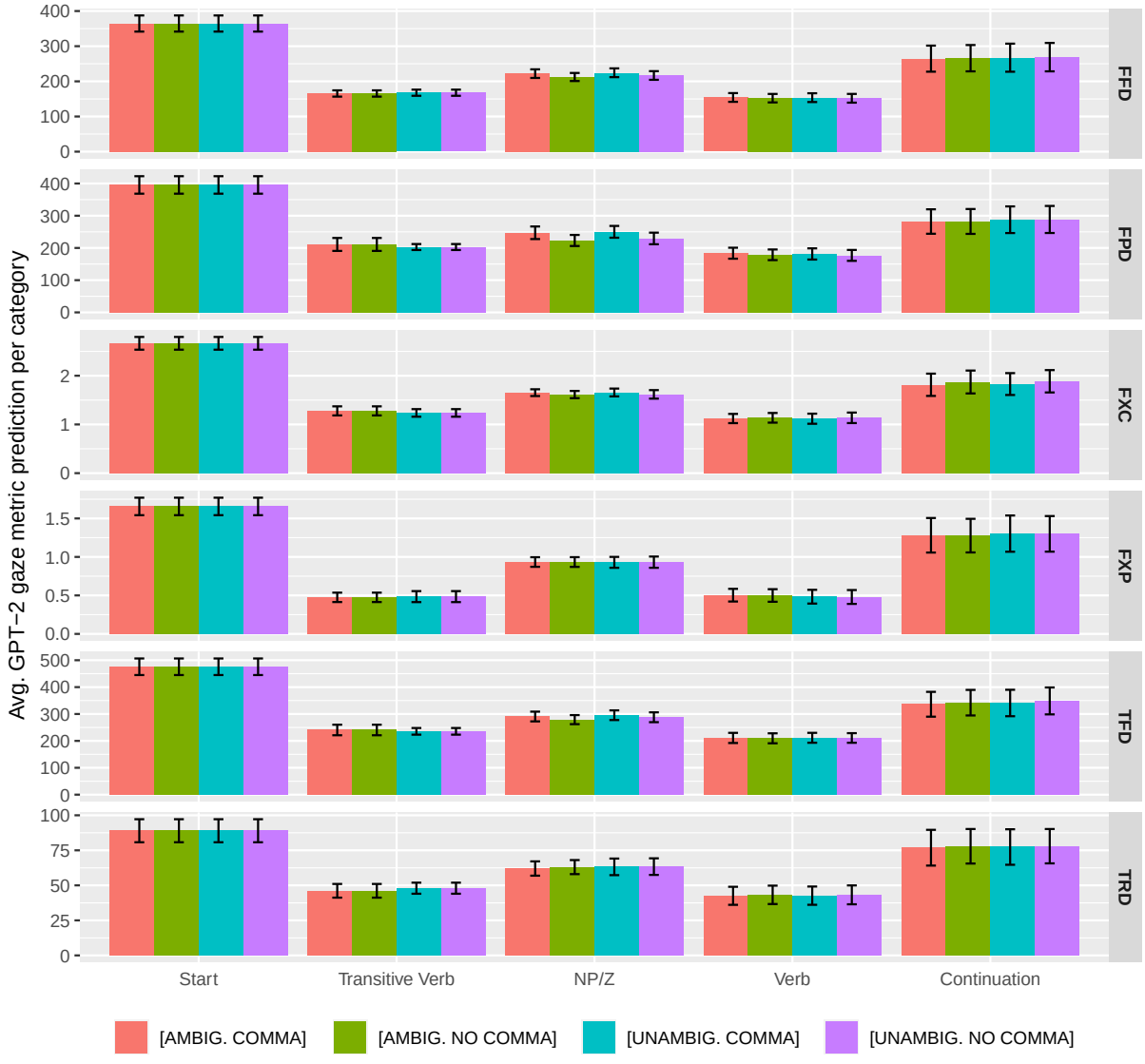


Figure E.1: Average GPT2-ET gaze metrics predictions for the "NP/Z Ambiguity with Verb Transitivity" SyntaxGym test suite. Bars show 95% confidence intervals. Units are in ms for durations, % for FXP, and raw counts for FXC.

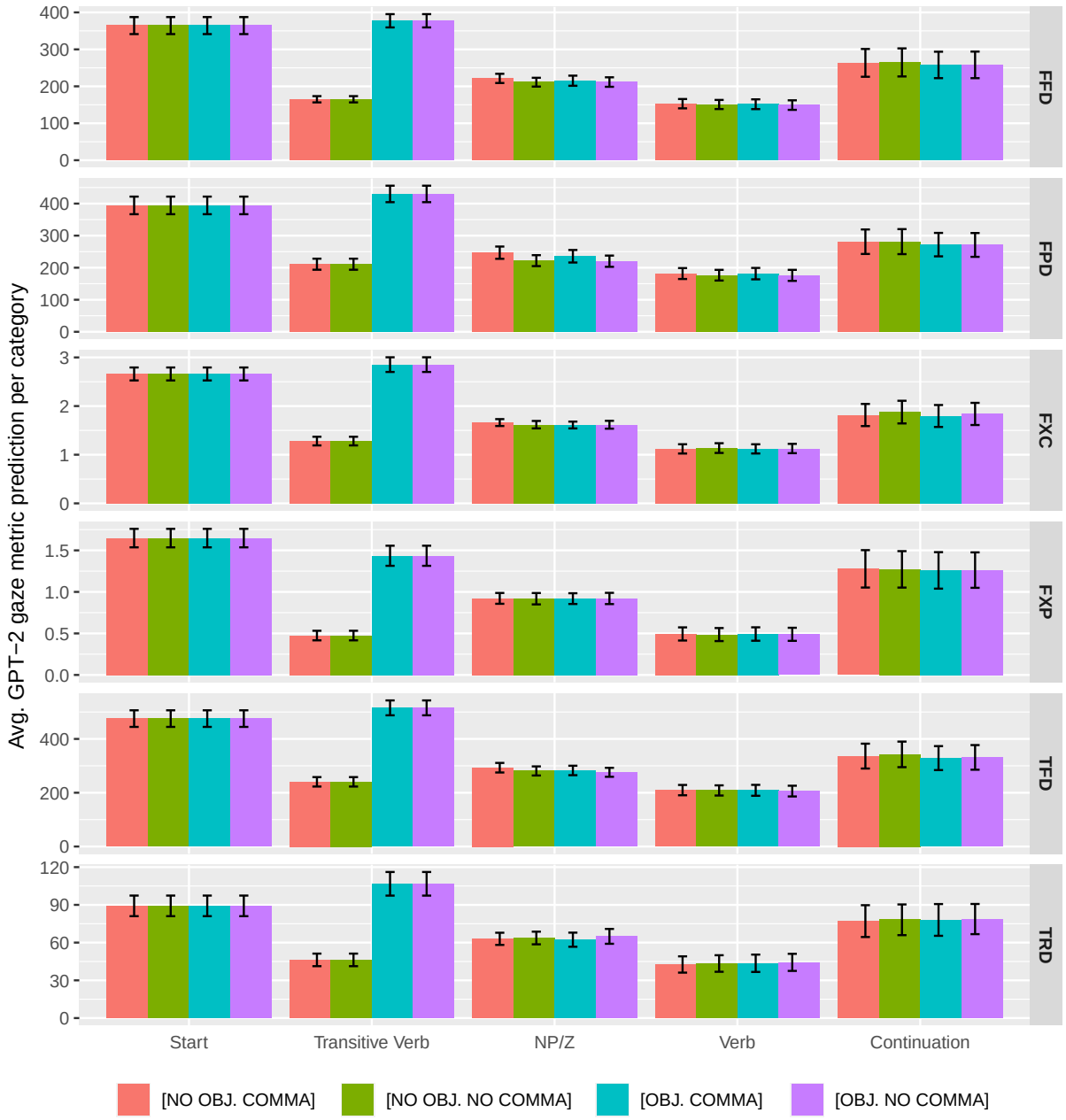


Figure E.2: Average GPT2-ET gaze metrics predictions for the "NP/Z Ambiguity with Overt Object" SyntaxGym test suite. Bars show 95% confidence intervals. Units are in ms for durations, % for FXP, and raw counts for FXC.

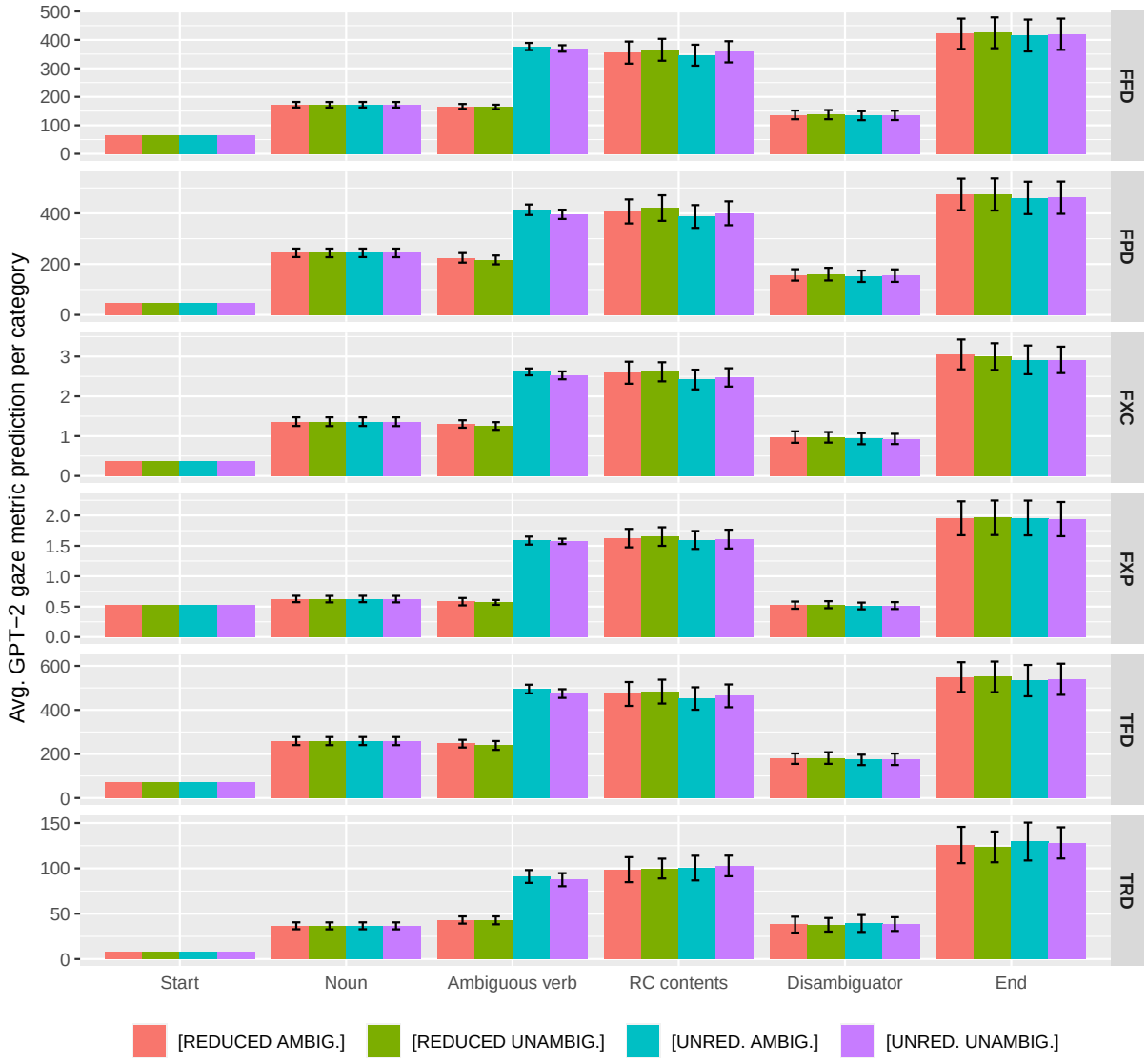


Figure E.3: Average GPT2-ET gaze metrics predictions for the "MV/RR Ambiguity" SyntaxGym test suite. Bars show 95% confidence intervals. Units are in ms for durations, % for FXP, and raw counts for FXC.

F | Reproducibility and Environmental Impact

Table F.1: Variable training parameters used in the experiments of this study. MTL stands for multitask learning.

	Chapter 3			**Chapter 4**			**Chapter 5**	
	PC	ET	Probes	PC	ET	RA	ALBERT	GPT-2
fine-tuning	standard	MTL	MTL	standard	MTL	standard	MTL	MTL
granularity	sent.	sent.	sent.	sent.	word	sent.	word	word
freeze LM w	✗	✗	✓	✗	✗	✗	✗	✗
weighted loss	-	✓	✗	-	✗	-	✗	✗
CV folds	5	5	5	-	-	-	-	-
early stopping	✓	✓	✗	✓	✓	✓	✓	✓
training epochs	15	15	5	15	15	15	15	15
patience	5	5	-	5	5	5	5	5
evaluation steps	20	40	-	20	100	80	100	100

Tools Experiments were executed on a Ubuntu 18.04 LTS server, using a NVIDIA K40 GPU with 12GB RAM and CUDA 10.1. Relevant Python libraries used throughout the study with their respective versions are: 🤗 `transformers 2.11.0` for accessing pre-trained Transformer language models, `farm 0.4.5` for multitask learning, `torch 1.3.0` as a backed for deep learning, and `syntaxgym 0.5.3` for Chapter 5 experiments. Python 3.6.3 was used for all training scripts. A custom adaptation of the Oxforddown template was used for this thesis.[1] Code for reproducibility purposes is available at the address https://github.com/gsarti/interpreting-complexity.

Model Training Table F.1 present the set of variable training parameters used in all the experiments of this study. Besides those, a set of fixed parameters was also used: all experiments were performed using a batch size of 32 observations, a maximum sequence length of 128 tokens, a linear training schedule with one-tenth of total steps used as warmup steps, the *AdamW* optimizer (Loshchilov et al., 2019) with weight decay equal to 0.01, and a learning rate of 10^{-5}. No hyperparameter search was performed due to time limitations.

[1] https://github.com/AI-Student-Society/thesisdown-it

Tokenization All tokenizers used in the experiments used cased text and were based respectively on the SentencePiece approach (Kudo et al., 2018) for ALBERT and a custom version of Byte-Pair Encoding tokenization (Sennrich et al., 2016) with token-like whitespaces for GPT-2. Default `AlbertTokenizer` and `GPT2Tokenizer` classes available in the 🤗 `transformers` library with pretrained tokenizers were used for this purpose. The vocabulary used by those had size 30'000 for ALBERT and 50'257 for GPT-2, including special tokens.

Architecture The default parameters for the 🤗 `transformers` checkpoints of ALBERT and GPT-2 (specifically, `albert-base-v2` and `gpt2` in the Model Hub) were used for this study. Concretely, this means embeddings and hidden sizes of 128 and 3072 for ALBERT and tied embedding-hidden size of 768 for GPT-2, 12 transformer blocks using 12 heads for multi-head self-attention each, and a smoothed variant of the Gaussian Error Linear Unit (GELU) as nonlinearity (Hendrycks et al., 2016). GPT-2 has an embedding and attention dropout rate of 0.1 and a layer normalization (Ba et al., 2016) epsilon of 10^{-5}, while ALBERT employs a classifier dropout rate of 0.1 and a layer normalization epsilon of 10^{-12}.

CO2 Emissions Related to Experiments Experiments were conducted using the private infrastructure of the ItaliaNLP Lab[2] at the Institute for Computational Linguistics "A. Zampolli" (ILC-CNR) in Pisa, which has an estimated carbon efficiency of 0.321 kgCO_2eq/kWh (Moro et al., 2018). A cumulative of roughly 100 hours of computation was performed on a Tesla K40 GPU (TDP of 245W). Total emissions are estimated to be 7.86 kgCO_2eq. Estimations were conducted using the Machine Learning Impact Calculator[3] presented in Lacoste et al. (2019).

In-detail reports of all experimental runsre produced automatically using the MLFlow[4] tool and are available at the following address: `https://public-mlflow.deepset.ai/#/experiments/99`.

[2] `https://www.italianlp.it`
[3] `https://mlco2.github.io/impact#compute`
[4] `https://mlflow.org/`

References

Abdou, Mostafa, Artur Kulmizev, Felix Hill, Daniel M. Low, and Anders Søgaard (Nov. 2019). "Higher-order Comparisons of Sentence Encoder Representations". 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 5838–5845.

Abnar, Samira (2020). "Visualizing Model Comparison". In: *Blog post*.

Abnar, Samira, Lisa Beinborn, Rochelle Choenni, and Willem Zuidema (Aug. 2019). "Blackbox Meets Blackbox: Representational Similarity & Stability Analysis of Neural Language Models and Brains". In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, pp. 191–203.

Abnar, Samira, Mostafa Dehghani, and Willem Zuidema (2020). "Transferring Inductive Biases through Knowledge Distillation". In: *ArXiv Pre-print* 2006.00555.

Alain, Guillaume and Yoshua Bengio (2016). "Understanding intermediate layers using linear classifier probes". In: *ArXiv Pre-print* 1610.01644.

Alammar, Jay (2018a). "The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)". In: *Blog post*.

— (2018b). "The Illustrated GPT-2". In: *Blog post*.

Ambati, Bharat Ram, Siva Reddy, and Mark Steedman (June 2016). "Assessing Relative Sentence Complexity using an Incremental CCG Parser". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 1051–1057.

Andreas, Jacob and Dan Klein (June 2014). "How much do word embeddings encode about syntax?" In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 822–827.

Ba, Jimmy, J. Kiros, and Geoffrey E. Hinton (2016). "Layer Normalization". In: *ArXiv Pre-print* 1607.06450.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *Proceeding of the 3rd International Conference on Learning Representations (ICLR'15)*.

Barrett, Maria, Joachim Bingel, Nora Hollenstein, Marek Rei, and Anders Søgaard (Oct. 2018). "Sequence Classification with Human Attention". In: *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Brussels, Belgium: Association for Computational Linguistics, pp. 302–312.

Barrett, Maria, Joachim Bingel, Frank Keller, and Anders Søgaard (Aug. 2016). "Weakly Supervised Part-of-speech Tagging Using Eye-tracking Data". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 579–584.

Belinkov, Yonatan, Sebastian Gehrmann, and Ellie Pavlick (July 2020). "Interpretability and Analysis in Neural NLP". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Online: Association for Computational Linguistics, pp. 1–5.

Belinkov, Yonatan and James Glass (2019). "Analysis Methods in Neural Language Processing: A Survey". In: *Transactions of the Association for Computational Linguistics (TACL)* 7, pp. 49–72.

Bender, Emily M. and Alexander Koller (July 2020). "Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 5185–5198.

Berruto, Gaetano and Massimo Simone Cerruti (2011). "La linguistica. Un corso introduttivo". De Agostini.

Berzak, Yevgeni, Boris Katz, and Roger Levy (June 2018). "Assessing Language Proficiency from Eye Movements in Reading". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 1986–1996.

Bever, Thomas G (1970). "The cognitive basis for linguistic structures". In: *Cognition and the development of language*.

Box, George EP (1976). "Science and statistics". In: *Journal of the American Statistical Association* 71.356, pp. 791–799.

Brunato, Dominique, Andrea Cimino, Felice Dell'Orletta, Giulia Venturi, and Simonetta Montemagni (May 2020). "Profiling-UD: a Tool for Linguistic Profiling of Texts". In: *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 7145–7151.

Brunato, Dominique, Lorenzo De Mattei, Felice Dell'Orletta, Benedetta Iavarone, and Giulia Venturi (Oct. 2018). "Is this Sentence Difficult? Do you Agree?" In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 2690–2699.

Cangelosi, Angelo and Huck Turner (2002). "L'emergere del linguaggio". In: *Scienze della Mente*.

Carr, Jon W, Valentina N Pescuma, Michele Furlan, Maria Ktori, and Davide Crepaldi (June 2020). "Algorithms for the automated correction of vertical drift in eye tracking data". In: *OSF Preprints*.

Caruana, Rich (1997). "Multitask Learning". In: *Machine Learning* 28, pp. 41–75.

Christie, Agatha (2003). "The mysterious affair at Styles: a detective story". Modern Library.

Collins-Thompson, Kevyn (2014). "Computational assessment of text readability: A survey of current and future research". In: *ITL-International Journal of Applied Linguistics* 165.2, pp. 97–135.

Conneau, Alexis, German Kruszewski, Guillaume Lample, Loıc Barrault, and Marco Baroni (July 2018). "What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 2126–2136.

Cop, Uschi, Nicolas Dirix, Denis Drieghe, and Wouter Duyck (2017). "Presenting GECO: An eyetracking corpus of monolingual and bilingual sentence reading". In: *Behavior research methods* 49.2, pp. 602–615.

Culotta, Aron, 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*. New York City, USA: Association for Computational Linguistics, pp. 296–303.

Day, Matthew (2004). "Religion, off-line cognition and the extended mind". In: *Journal of cognition and Culture* 4.1, pp. 101–121.

Demberg, Vera and Frank Keller (2008). "Data from eye-tracking corpora as evidence for theories of syntactic processing complexity". In: *Cognition* 109.2, pp. 193–210.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". 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). Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186.

Edmonds, Bruce M. (1999). "Syntactic measures of complexity". PhD thesis. University of Manchester Manchester, UK.

Eisape, Tiwalayo, Noga Zaslavsky, and Roger Levy (Nov. 2020). "Cloze Distillation Improves Psychometric Predictive Power". In: *Proceedings of the 24th Conference on Computational Natural Language Learning*. Online: Association for Computational Linguistics, pp. 609–619.

Eisenstein, Jacob (2019). "Introduction to natural language processing". MIT press.

Elman, Jeffrey L (1991). "Distributed representations, simple recurrent networks, and grammatical structure". In: *Machine learning* 7.2-3, pp. 195–225.

Ettinger, Allyson (2020). "What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models". In: *Transactions of the Association for Computational Linguistics* 8, pp. 34–48.

Fine, Alex B, T Florian Jaeger, Thomas A Farmer, and Ting Qian (2013). "Rapid expectation adaptation during syntactic comprehension". In: *PloS one* 8.10, e77661.

Frankle, Jonathan and Michael Carbin (2018). "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks". In: *Proceedings of the 8th International Conference on Learning Representations (ICLR'18)*.

Frazier, Lyn (1978). "On comprehending sentences: Syntactic parsing strategies". PhD thesis. University of Connecticut.

Frazier, Lyn and Janet Dean Fodor (1978). "The sausage machine: A new two-stage parsing model". In: *Cognition* 6.4, pp. 291–325.

Futrell, Richard, Edward Gibson, and Roger P Levy (2020). "Lossy-context surprisal: An information-theoretic model of memory effects in sentence processing". In: *Cognitive science* 44.3, e12814.

Futrell, Richard, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy (June 2019). "Neural language models as psycholinguistic subjects: Representations of syntactic state". 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 32–42.

Gauthier, Jon, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy (July 2020). "SyntaxGym: An Online Platform for Targeted Evaluation of Language Models". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics, pp. 70–76.

Gauthier, Jon and Roger Levy (Nov. 2019). "Linking artificial and human neural representations of language". 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 529–539.

Gibson, Edward (1991). "A computational theory of human linguistic processing: Memory limitations and processing breakdown". PhD thesis. Pittsburgh, PA: Carnegie Mellon University.

— (1998). "Linguistic complexity: Locality of syntactic dependencies". In: *Cognition* 68.1, pp. 1–76.

— (2000). "The dependency locality theory: A distance-based theory of linguistic complexity". In: *Image, language, brain* 2000, pp. 95–126.

Glavas, Goran and Ivan Vulic (2020). "Is Supervised Syntactic Parsing Beneficial for Language Understanding? An Empirical Investigation". In: *ArXiv Pre-print* 2008.06788.

González-Garduño, Ana Valeria and Anders Søgaard (2018). "Learning to Predict Readability Using Eye-Movement Data From Natives and Learners". In: *AAAI Conference on Artificial Intelligence 2018*. AAAI Conference on Artificial Intelligence.

Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio (2016). "Deep learning". MIT Press Cambridge.

Goodman, Joshua (2001). "A bit of progress in language modeling". In: *arXiv preprint cs/0108005*.

Grodner, Daniel, Edward Gibson, Vered Argaman, and Maria Babyonyshev (2003). "Against repair-based reanalysis in sentence comprehension". In: *Journal of Psycholinguistic Research* 32.2, pp. 141–166.

Gulordava, Kristina, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni (June 2018). "Colorless Green Recurrent Networks Dream Hierarchically". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 1195–1205.

Hale, John (2001). "A probabilistic Earley parser as a psycholinguistic model". In: *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

— (2016). "Information-theoretical complexity metrics". In: *Language and Linguistics Compass* 10.9, pp. 397–412.

Hauser, Marc D, Noam Chomsky, and W Tecumseh Fitch (2002). "The faculty of language: what is it, who has it, and how did it evolve?" In: *Science* 298.5598, pp. 1569–1579.

Hendrycks, Dan and Kevin Gimpel (2016). "Gaussian Error Linear Units (GELUs)." In: *ArXiv Pre-print* 1606.08415.

Hewitt, John and Percy Liang (Nov. 2019). "Designing and Interpreting Probes with Control Tasks". 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 2733–2743.

Hewitt, John 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4129–4138.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean (2015). "Distilling the Knowledge in a Neural Network". In: *ArXiv Pre-print* 1503.02531.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Hollenstein, Nora, Maria Barrett, and Lisa Beinborn (May 2020). "Towards Best Practices for Leveraging Human Language Processing Signals for Natural Language Processing". In: *Proceedings of the Second Workshop on Linguistic and Neurocognitive Resources*. Marseille, France: European Language Resources Association, pp. 15–27.

Hollenstein, Nora, Antonio de la Torre, Nicolas Langer, and Ce Zhang (Nov. 2019). "CogniVal: A Framework for Cognitive Word Embedding Evaluation". In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, pp. 538–549.

Hollenstein, Nora, Jonathan Rotsztejn, Marius Troendle, Andreas Pedroni, Ce Zhang, and Nicolas Langer (2018). "ZuCo, a simultaneous EEG and eye-tracking resource for natural sentence reading". In: *Scientific data* 5.1, pp. 1–13.

Hollenstein, Nora, Marius Troendle, Ce Zhang, and Nicolas Langer (May 2020). "ZuCo 2.0: A Dataset of Physiological Recordings During Natural Reading and Annotation". English. In: *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 138–146.

Hollenstein, Nora and Ce Zhang (June 2019). "Entity Recognition at First Sight: Improving NER with Eye Movement Information". 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). Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1–10.

Howard, Jeremy and Sebastian Ruder (July 2018). "Universal Language Model Fine-tuning for Text Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 328–339.

Hu, Jennifer, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy (July 2020). "A Systematic Assessment of Syntactic Generalization in Neural Language Models". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 1725–1744.

Iverson, Jana M and Esther Thelen (1999). "Hand, mouth and brain. The dynamic emergence of speech and gesture". In: *Journal of Consciousness Studies* 6.11-12, pp. 19–40.

Jawahar, Ganesh, Benoit Sagot, and Djamé Seddah (July 2019). "What Does BERT Learn about the Structure of Language?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3651–3657.

Jurafsky, Daniel (1996). "A probabilistic model of lexical and syntactic access and disambiguation". In: *Cognitive science* 20.2, pp. 137–194.

Kennedy, Alan, Robin Hill, and Joël Pynte (2003). "The dundee corpus". In: *Proceedings of the 12th European conference on eye movement*.

Kriegeskorte, N., M. Mur, and P. Bandettini (2008). "Representational Similarity Analysis – Connecting the Branches of Systems Neuroscience". In: *Frontiers in Systems Neuroscience* 2.

Kudo, Taku and John Richardson (Nov. 2018). "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, pp. 66–71.

Kusters, Wouter (2003). "Linguistic complexity". PhD thesis. Netherlands Graduate School of Linguistics.

— (2008). "Complexity in linguistic theory, language learning and language change". In: *Language complexity: Typology, contact, change*. John Benjamins Amsterdam, The Netherlands, pp. 3–22.

Laakso, Aarre and Garrison Cottrell (2000). "Content and cluster analysis: assessing representational similarity in neural systems". In: *Philosophical psychology* 13.1, pp. 47–76.

Lacoste, Alexandre, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres (2019). "Quantifying the Carbon Emissions of Machine Learning". In: *ArXiv Pre-print* 1910.09700.

Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut (2020). "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations". In: *International Conference on Learning Representations*.

Levy, Roger (2008). "Expectation-based syntactic comprehension". In: *Cognition* 106.3, pp. 1126–1177.

Lin, Yongjie, Yi Chern Tan, and Robert Frank (Aug. 2019). "Open Sesame: Getting inside BERT's Linguistic Knowledge". In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, pp. 241–253.

Linzen, Tal and Marco Baroni (2021). "Syntactic Structure from Deep Learning". In: *Annual Review of Linguistics* 7.1, null.

Linzen, Tal, Emmanuel Dupoux, and Yoav Goldberg (2016). "Assessing the ability of LSTMs to learn syntax-sensitive dependencies". In: *Transactions of the Association for Computational Linguistics* 4, pp. 521–535.

Liu, Nelson F., Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith (June 2019). "Linguistic Knowledge and Transferability of Contextual 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1073–1094.

Loshchilov, I. and F. Hutter (2019). "Decoupled Weight Decay Regularization". In: *Proceeding of the 7th International Conference on Learning Representations (ICLR'19)*.

Martinc, Matej, S. Pollak, and M. Robnik-Sikonja (2019). "Supervised and unsupervised neural approaches to text readability". In: *ArXiv Pre-print* 1907.11779.

McDonald, Ryan, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee (Aug. 2013). "Universal Dependency Annotation for Multilingual Parsing". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 92–97.

McWhorter, John H (2001). "The worlds simplest grammars are creole grammars". In: *Linguistic typology* 5.2-3, pp. 125–166.

Meyer, Bonnie JF and G Elizabeth Rice (1992). "12 Prose processing in adulthood: The text, the reader, and the task". In: *Everyday cognition in adulthood and late life*, p. 157.

Miaschi, Alessio, Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi (Dec. 2020). "Linguistic Profiling of a Neural Language Model". In: *Proceedings of the 28th Conference on Computational Linguistics (COLING)*. Online: Association for Computational Linguistics.

Miaschi, Alessio and Felice Dell'Orletta (July 2020). "Contextual and Non-Contextual Word Embeddings: an in-depth Linguistic Investigation". In: *Proceedings of the 5th Workshop on Representation Learning for NLP*. Online: Association for Computational Linguistics, pp. 110–119.

Miestamo, Matti (2004). "On the Feasibility of Complexity Metrics". In: *FinEst linguistics, Proceedings of the Annual Finnish and Estonian Conference of Linguistics*. Tallin, Finland, pp. 11–26.

— (2008). "Grammatical complexity in a cross-linguistic perspective". In: *Language complexity: Typology, contact, change*. John Benjamins Amsterdam, The Netherlands, p. 41.

Mikolov, Tomas, Kai Chen, G. S. Corrado, and J. Dean (2013). "Efficient Estimation of Word Representations in Vector Space". In: *CoRR* abs/1301.3781.

Mikolov, Tomas, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur (2010). "Recurrent neural network based language model". In: *INTERSPEECH*.

Mishra, Abhijit, Kuntal Dey, and Pushpak Bhattacharyya (July 2017). "Learning Cognitive Features from Gaze Data for Sentiment and Sarcasm Classification using Convolutional Neural Network". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 377–387.

Mitchell, Don C (1984). "An evaluation of subject-paced reading tasks and other methods for investigating immediate processes in reading". In: *New methods in reading comprehension research*, pp. 69–89.

Morcos, Ari, Maithra Raghu, and Samy Bengio (2018). "Insights on representational similarity in neural networks with canonical correlation". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Curran Associates, Inc., pp. 5727–5736.

Moro, Alberto and Laura Lonza (2018). "Electricity carbon intensity in European Member States: Impacts on GHG emissions of electric vehicles". In: *Transportation Research Part D: Transport and Environment* 64, pp. 5–14.

Munro, Robert, Steven Bethard, Victor Kuperman, Vicky Tzuyin Lai, Robin Melnick, Christopher Potts, Tyler Schnoebelen, and Harry Tily (June 2010). "Crowdsourcing and language studies: the new generation of linguistic data". In: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. Los Angeles: Association for Computational Linguistics, pp. 122–130.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman (May 2016). "Universal Dependencies v1: A Multilingual Treebank Collection". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), pp. 1659–1666.

Pascanu, R., Tomas Mikolov, and Yoshua Bengio (2013). "On the difficulty of training recurrent neural networks". In: *Proceedings of the 30th International Conference on Machine Learning (ICML'13)*.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (Oct. 2014). "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543.

Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (June 2018). "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 2227–2237.

Pimentel, Tiago, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell (July 2020). "Information-Theoretic Probing for Linguistic Structure". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4609–4622.

Prasad, Grusha and Tal Linzen (2019a). "Do self-paced reading studies provide evidence for rapid syntactic adaptation?" In: *PsyArXiv Pre-print*.

— (2019b). "How much harder are hard garden-path sentences than easy ones?" In: *OSF Preprint* syh3j.

Radford, A., Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). "Language Models are Unsupervised Multitask Learners". In: *OpenAI Blog*.

Raghu, Maithra, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein (2017). "SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., pp. 6076–6085.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang (Nov. 2016). "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 2383–2392.

Rayner, Keith (1998). "Eye movements in reading and information processing: 20 years of research." In: *Psychological bulletin* 124.3, p. 372.

Rello, Luz, Susana Bautista, Ricardo Baeza-Yates, Pablo Gervás, Raquel Hervás, and Horacio Saggion (2013). "One Half or 50%? An Eye-Tracking Study of Number Representation Readability". In: *Human-Computer Interaction – INTERACT 2013*. Ed. by Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 229–245.

Rogers, Anna, O. Kovaleva, and Anna Rumshisky (2020). "A Primer in BERTology: What we know about how BERT works". In: *ArXiv Pre-print* 2002.12327.

Ruder, Sebastian (2017). "An Overview of Multi-Task Learning in Deep Neural Networks". In: *ArXiv Pre-print* 1706.05098.

— (2020). "Why You Should Do NLP Beyond English". In: *Blog post*.

Samek, W., Grégoire Montavon, A. Vedaldi, L. Hansen, and K. Müller (2019). "Explainable AI: Interpreting, Explaining and Visualizing Deep Learning". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*.

Sanguinetti, Manuela and Cristina Bosco (2015). "PartTUT: The Turin University Parallel Treebank". In: *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*. Ed. by Roberto Basili, Cristina Bosco, Rodolfo Delmonte, Alessandro Moschitti, and Maria Simi. Cham: Springer International Publishing, pp. 51–69.

Saphra, Naomi and Adam Lopez (June 2019). "Understanding Learning Dynamics Of Language Models with SVCCA". 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 3257–3267.

Sapir, Edward (1921). "Language". In:

Sarti, Gabriele (2020). "UmBERTo-MTSA @ AcCompl-It: Improving Complexity and Acceptability Prediction with Multi-task Learning on Self-Supervised Annotations". In: *ArXiv Pre-print* 2011.05197.

Schotter, Elizabeth R (2018). "Reading ahead by hedging our bets on seeing the future: Eye tracking and electrophysiology evidence for parafoveal lexical processing and saccadic control by partial word recognition". In: *Psychology of Learning and Motivation*. Vol. 68. Elsevier, pp. 263–298.

— (2020). "Eye Tracking for Cognitive Science". SISSA Course.

Schotter, Elizabeth R, Bernhard Angele, and Keith Rayner (2012). "Parafoveal processing in reading". In: *Attention, Perception, & Psychophysics* 74.1, pp. 5–35.

Sennrich, Rico, 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)*. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725.

Shwartz-Ziv, Ravid and Naftali Tishby (2017). "Opening the Black Box of Deep Neural Networks via Information". In: *ArXiv Pre-print* 1703.00810.

Silveira, Natalia, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning (May 2014). "A Gold Standard Dependency Corpus for English". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pp. 2897–2904.

Simi, Maria, Cristina Bosco, and Simonetta Montemagni (May 2014). "Less is More? Towards a Reduced Inventory of Categories for Training a Parser for the Italian Stanford Dependencies". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pp. 83–90.

Singh, Abhinav Deep, Poojan Mehta, Samar Husain, and Rajkumar Rajakrishnan (Dec. 2016). "Quantifying sentence complexity based on eye-tracking measures". In: *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 202–212.

Sinnemäki, Kaius (2011). "Language universals and linguistic complexity: Three case studies in core argument marking". PhD thesis. University of Helsinki.

Smith, Nathaniel J and Roger Levy (2013). "The effect of word predictability on reading time is logarithmic". In: *Cognition* 128.3, pp. 302–319.

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts (Oct. 2013). "Recursive Deep Models for Semantic Compositionality Over a

Sentiment Treebank". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1631–1642.

Stiennon, Nisan, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano (2020). "Learning to summarize from human feedback". In: *ArXiv Pre-print* 2009.01325.

Straka, Milan, Jan Hajič, and Jana Straková (May 2016). "UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), pp. 4290–4297.

Strzyz, Michalina, David Vilares, and Carlos Gómez-Rodríguez (Nov. 2019). "Towards Making a Dependency Parser See". 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 1500–1506.

Sturt, Patrick, Martin J Pickering, and Matthew W Crocker (1999). "Structural change and reanalysis difficulty in language comprehension". In: *Journal of Memory and Language* 40.1, pp. 136–150.

Sussillo, David, Mark M Churchland, Matthew T Kaufman, and Krishna V Shenoy (2015). "A neural network that finds a naturalistic solution for the production of muscle activity". In: *Nature neuroscience* 18.7, pp. 1025–1033.

Tay, Yi, Dara Bahri, Donald Metzler, D. Juan, Zhe Zhao, and Che Zheng (2020). "Synthesizer: Rethinking Self-Attention in Transformer Models". In: *ArXiv Pre-print* 2005.00743.

Taylor, Wilson L (1953). ""Cloze procedure": A new tool for measuring readability". In: *Journalism quarterly* 30.4, pp. 415–433.

Tenney, Ian, 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*. Florence, Italy: Association for Computational Linguistics, pp. 4593–4601.

Thompson, Bruce (1984). "Canonical correlation analysis: Uses and interpretation". 47. Sage.

Turian, Joseph, Lev-Arie Ratinov, and Yoshua Bengio (July 2010). "Word Representations: A Simple and General Method for Semi-Supervised Learning". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, pp. 384–394.

Vajjala, Sowmya and Ivana Lucic (Aug. 2019). "On Understanding the Relation between Expert Annotations of Text Readability and Target Reader Comprehension". In: *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Florence, Italy: Association for Computational Linguistics, pp. 349–359.

Vajjala, Sowmya and Ivana Lučić (June 2018). "OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification". In: *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 297–304.

Van Schijndel, Marten and Tal Linzen (2018). "Modeling garden path effects without explicit hierarchical syntax." In: *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, pp. 2600–2605.

Van Schijndel, Marten and Tal Linzen (2020). "Single-stage prediction models do not explain the magnitude of syntactic disambiguation difficulty". In: *PsyArXiv Pre-print* sgbqy.

Vapnik, Vladimir (1998). "The Support Vector Method of Function Estimation". In: *Nonlinear Modeling: Advanced Black-Box Techniques*. Ed. by Johan A. K. Suykens and Joos Vandewalle. Boston, MA: Springer US, pp. 55–85.

Vasishth, Shravan, Titus von der Malsburg, and Felix Engelmann (2013). "What eye movements can tell us about sentence comprehension". In: *Cognitive science* 4 2, pp. 125–134.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., pp. 5998–6008.

Voghera, Miriam (2001). "Riflessioni su semplificazione, complessità e modalità di trasmissione: sintassi e semantica". In: *Scritto e parlato. Metodi, testi e contesti*, pp. 65–78.

Voita, Elena, Rico Sennrich, and Ivan Titov (Nov. 2019). "The Bottom-up Evolution of Representations in the Transformer: A Study with Machine Translation and Language Modeling Objectives". 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 4396–4406.

Wallace, Eric, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner (Nov. 2019). "Do NLP Models Know Numbers? Probing Numeracy in Embeddings". 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 5307–5315.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (Nov. 2018). "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, pp. 353–355.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush (Oct. 2020). "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 38–45.

Wu, Y., Mike Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, M. Krikun, Yuan Cao, Q. Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, Taku Kudo, H. Kazawa, K. Stevens, G. Kurian, Nishant Patil, et al. (2016). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". In: *ArXiv Pre-print* 1609.08144.

Xu, Wei, Chris Callison-Burch, and Courtney Napoles (2015). "Problems in Current Text Simplification Research: New Data Can Help". In: *Transactions of the Association for Computational Linguistics* 3, pp. 283–297.

Zanzotto, Fabio Massimo, Andrea Santilli, Leonardo Ranaldi, Dario Onorati, Pierfrancesco Tommasino, and Francesca Fallucchi (Nov. 2020). "KERMIT: Complementing Transformer Architectures with Encoders of Explicit Syntactic Interpretations". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 256–267.

Zeldes, Amir (2017). "The GUM corpus: creating multilayer resources in the classroom". In: *Language Resources and Evaluation* 51, pp. 581–612.

Zhang, Kelly and Samuel Bowman (Nov. 2018). "Language Modeling Teaches You More than Translation Does: Lessons Learned Through Auxiliary Syntactic Task Analysis". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, pp. 359–361.