UNIVERSITÄT LEIPZIG

OCTOBER 2025

CROSS-LINGUISTIC SYNTACTIC EVALUATION OF TRANSFORMERS VIA TREEBANK QUERYING

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE THESIS DIVISION
IN CANDIDACY FOR THE DEGREE OF
M.A. LINGUISTICS

DEPARTMENT OF LINGUISTICS

BY
DANIEL GALLAGHER

SUPERVISED BY
PROFESSOR DR. GREG KOBELE, DEPT. LINGUISTICS
PROFESSOR DR. GERHARD HEYER, DEPT. COMPUTER SCIENCE

TABLE OF CONTENTS

APPENDICES

ABSTRACT

This thesis contributes to existing research on Targeted Syntactic Evaluation (TSE) of transformer-based language models by presenting a novel method of generating minimal-pair syntactic tests using `Grew`, a graph-based query language, for the isolation of syntactic phenomena in Universal Dependency (UD) treebanks. We first introduce the relevant literature and discuss a number of important considerations when designing experiments for TSE research, such as next-token recovery and subsequence biases as well as the avoidance of sentence-level probability metrics. In order to examine model generalisation, we discuss a partial solution through the use of semantically implausible minimal pairs. The resulting pipeline handles the query-based isolation of syntactic phenomena from treebanks, their subsequent transformation into mask- or prompt-based minimal-pair syntactic tests, and the resulting Hugging Face model evaluation. The accuracy, average surprisal difference, and an entropic certainty score are automatically computed. Experiments are carried out for (1) encoder-only models on the genitive case in Polish, (2) decoder-only models on conditional auxiliary constructions in German, and (3) encoder-only models on split-ergative case alignment in Georgian. We find that models tend to perform better on semantically plausible over semantically implausible minimal pairs despite no difference in grammaticality over the original minimal pairs. In most cases however, models perform well on both types of datasets indicating partial generalisation. Rarer syntactic constructions are consistently underrepresented in model performance, indicating a possible sensitivity to frequency distribution for models with insufficient training data. We encourage a greater transparency in the proportion of such data used for each language and a push towards greater coverage of rarer syntactic phenomena and low-resource languages. Lastly, smaller monolingual models tend to perform better than bigger multilingual models, further corroborating previous work on the ‘curse of multilinguality’. These results show that this pipeline can be successfully used to identify syntactic performance gaps in language models.

In Memory of Dr. Brett Becker.

CHAPTER 1

INTRODUCTION

"Any fool can turn a blind eye, but who knows what the ostrich sees in the sand?"

— Samuel Beckett, Murphy, 1938

Although the meaning of the above quotation may not be immediately clear, we intuitively recognise that it is grammatically well-formed. Each word sits comfortably with the others without any strong sense of dissonance between them. We can recognise that *sees* depends on *the ostrich* in some way. That is, there is some form of dependency relation between the two; if it were *the ostriches* then the verb would need to be adjusted to *see*. By *dependency*, we therefore mean that if one word or morpheme changes then it causes the necessary change of another. Each word also has its own *features*. For instance, `number` and `gender` are common features for nouns, or `person`, `mood`, and `aspect` for verbs. These features, dependencies, and the rules governing their arrangement constitute what we call *syntax*. Many theoretical frameworks exist for explaining syntax, including *generative grammar*, *HPSG*, *LSG*, and *dependency grammar*. We focus here primarily on the latter, which can be seen as a combination of structural relations between words and the morphosyntactic features they carry.

Our primary goal is to explore a novel means of evaluating the *syntactic performance* of language models (henceforth LMs), as defined in Definition 1 [Newman et al., 2021]. Judgements of grammaticality can vary across speakers, contexts, and usages. Similarly, ones own grammatical *competence*, what a speaker knows about their internal grammar, does not always reflect one's grammatical *performance*, what a speaker actually says. We will take *grammaticality* to be the degree to which an utterance corresponds to the structural patterns of a language as represented by a speaker's *competence*, yet focus on evaluating a model's *performance*. Much work has been carried out in refining the standard approaches of evaluating the grammatical performance of LMs [Warstadt et al., 2023, Newman et al., 2021, Kulmizev and Nivre, 2021] as well as some work in the use of treebanks [Jumelet et al., 2025], however this is the first such method that allows one to define their own syntactic phenomena for evaluation through a query-based language. An overview of the pipeline is shown in Figure 1.1.

Definition 1. *Syntactic performance refers to the measurement of a model's systematic preference for grammatical over ungrammatical words or subwords at critical points in a sentence, reflecting its underlying syntactic competence.*

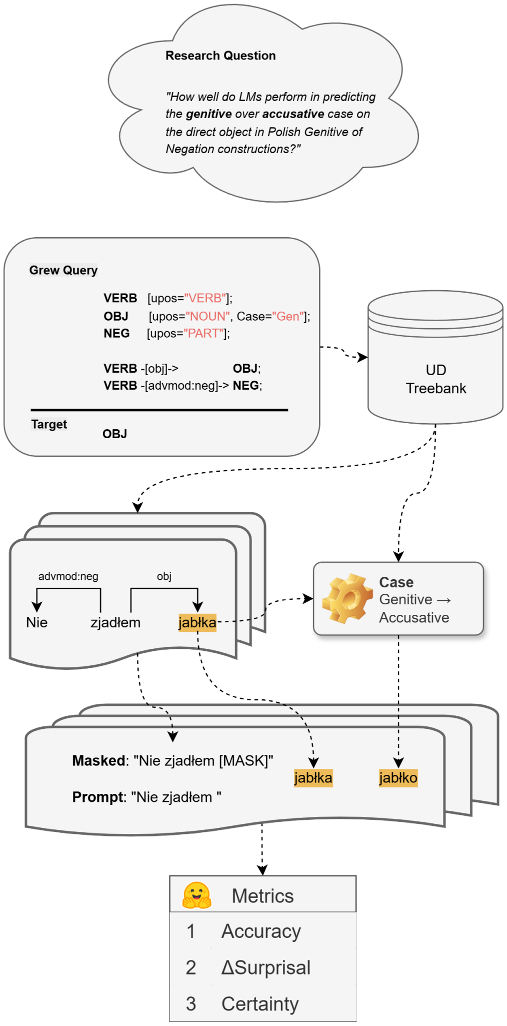


Figure 1.1: One example use case of the the GrewTSE pipeline for syntactic evaluation of Polish genitive of negation constructions.

1.1 Can language models tell us anything about syntax?

LMs have led to a paradigm shift in how we consider the learning of language by a machine due to their uncanny ability to comprehend and fluently produce non-trivial and ostensibly creative text. The uncanny factor of their output can be further refined through methods such as *reinforcement learning from human feedback* (RLHF), where models are 'taught' not just to predict most-likely continuations of their input but also to respond in a more human-like fashion [Bai et al., 2022]. Might these models have something to teach us about our *own* ability to learn syntax? The success of LMs raises a number of issues with the argument that language cannot be learned without domain-specific formal constraints [Futrell and Mahowald, 2025]. The *poverty of the stimulus* argument however, i.e. that the number of sentences a child hears is too limited to explain their rapid learning of language, still holds in the sense that LMs require *vastly* larger amounts of linguistic input in order to produce fluent text [Lappin and Shieber, 2007]. Chomsky claims that LMs have no potential for meaningful contribution in the field of linguistics by arguing that they are able to learn so-called 'impossible' languages just as well as real ones [Chomsky and Mirfakhraie, 2023], it has been repeatedly found however that LMs *do* struggle with such languages [Kallini et al., 2024, Xu et al., 2025a, Yang et al., 2025a]. Ziv et al. [2025] argues that LMs do not in any way explain human linguistic cognition themselves but instead may have a role to play in testing theories practically. For instance, a theory may lead to certain conclusions about syntactic difficulty that we would expect to be mirrored in a machine learning model. We believe that it is possible that these models open up new avenues for research in linguistics, but that we must first be able to evaluate models on a more complete set of syntactic phenomena cross-linguistically before we are able to meaningfully do so.

1.2 Can syntax tell us anything about language models?

Evaluations of the syntactic performance of language models have only been carried out on a small minority of the world's languages, however this disparity is beginning to be addressed through initiatives such as European Language Equality [Rehm and Way, 2023]. The heavy focus on this tiny subset, it is argued, further exacerbates the inability of *low-resource language* (LRL) speakers to access modern language technologies. Syntactic evaluation is a key method of studying which LMs are performing poorly and will be a primary focus of this work. We aim to improve the *granularity* i.e. level of detail with which one can carry out these types of evaluations. We focus on building a pipeline that allows linguists, computer scientists, or anyone with a curiosity for a language they are familiar with to run their own experiments on highly-specific syntactic constructions through grammatical/ungrammatical minimal-pair datasets. Many LMs released today are purportedly multilingual, however performance in any given language tends to rely heavily on that language's own representation in the training data. Furthermore, larger studies of the syntactic capabilities of multilingual LMs tend to examine simple agreement phenomena and avoid more complex or unique constructions. We believe that this leads to an underrepresentation of these models for LRLs and through this work we aim to provide an avenue for the rapid creation of such tests for user-defined syntactic phenomena across any of the over 150 languages annotated within the *Universal Dependencies* (UD) treebank project. With an ever-increasing number of represented languages, researchers can explore a broader variety of phenomena and more precisely identify where models succeed or fall short.

1.3 Research Questions

This work only proves useful if it can be helpful in contributing to research questions (RQs) pertaining to targeted syntactic evaluation of LMs. We therefore devise three RQs in this domain and aim to provide a small contribution to each, largely through experiments on syntactic phenomena in Polish, German, and Georgian. These are outlined below.

RQ1 *To what extent can transformer-based language models accurately predict key syntactic words or morphemes within syntactic structures across typologically diverse languages, and how does performance vary by syntactic phenomenon?*

RQ2 *How do differences in architecture, size, and tokenisation affect performance on the same syntactic phenomenon?*

RQ3 *What differences do we observe in model performance when we evaluate on syntactic minimal pairs that are semantically implausible, and what does this tell us about a model's generalisation of syntactic structure?*

RQ1 deals with how well our models have handled the syntactic tests we have devised in our experiments cross-linguistically. We focus here on the overall results as opposed to analysing individual languages or models. In RQ2, we will look at variations within each language across models tested for that language. There are many model hyperparameters that can be adjusted, such as number of weights, the tokeniser, languages trained on, number of steps in pre-training, as well as vocabulary size. This question has been less constrained by dataset creation efforts as we can use the same dataset across many models, however for LRLs there tends to be a more restricted variety of models available for testing. In RQ3, we aim to observe how performance is affected by the use of semantically implausible minimal pairs that aim to approach an out-of-distribution test. Due to the difficulty of accounting for the full dataset used in model pre-training and fine-tuning, it is difficult to know however when even semantically implausible sentences have been seen. As will be further explored in

Section 3, the increasing difficulty in controlling training data in language models has made it evermore difficult to assess whether a model has indeed *generalised* and not simply *rote learned.* Specifically, it is difficult to discern whether a model has learned syntactic patterns rather than simply memorising each item in the lexicon within its respective grammatical context. While we aim in this thesis to analyse performance across a wide variety of models, due to hardware constraints we avoid testing models that are greater than approximately 500 million parameters and leave this as a valuable route for future work.

1.4 Thesis Structure

We introduce the background and related work in Section 2, exploring representations of syntax, transformer-based language models, and previous work in syntactic evaluation. In Section 3 we expand on current research in this domain by discussing a number of important considerations in experiment design such as controlling for confounds, testing on data unseen in training, potential semantic violations that can cause a bias in results, as well as a critique of using sentence-level probabilities over token-level probabilities. The methodology of our pipeline is outlined in Section 4 and the resulting experiments are carried out in Section 5, where we examine Polish genitive of negation constructions (Section 5.1), German conditional auxiliaries (Section 5.2), and the Georgian split-ergative case alignment system (Section 5.3). Lastly, we conclude the thesis in Section 6 with an overview of our findings.

CHAPTER 2

BACKGROUND & RELATED WORK

We provide an overview of the background literature for the relevant aspects of syntax, transformer-based LMs, and evaluation of LM syntactic performance. In Section 2.1 we will cover various theories of natural language syntax such as *generative* and *dependency* grammars. Section 2.2 will concern itself with developments in the field of *natural language processing* (NLP) with respect to transformers and the subsequent laws of scaling that led to the ubiquitous large language model (LLM). We will discuss the origins of sequence-to-sequence models such as the Recurrent Neural Network (RNN) or its variant the Long Short-Term Network (LSTM). We additionally discuss the important concept of *attention* in improving performance on long-distance dependencies as well as greater contexts in general. Finally, Section 2.3 will build on these two domains to explore the evaluation of transformers on their syntactic capabilities when producing and comprehending natural language. We will discuss means of interpreting a model's internal state to search for knowledge of syntax and the alternative of analysing token probability distributions. The standard benchmarks and metrics for this domain are introduced along with a number of criticisms of the standard research methodology.

2.1 Representations of Syntax

We examine two primary traditions in the study of natural language syntax. Section 2.1.1 will revolve around *constituency* grammar and Section 2.1.2 around *dependency* grammar. There are many theories which have emerged as a result of the former, such as Lexical Functional Grammar [Kaplan and Bresnan, 1982], Head-Driven Phrase Structure Grammar [Pollard et al., 2002], and Categorial Grammar. While there is a lot that can be said about each, they will not be discussed deeply in this work.

2.1.1 Constituency Grammar

Constituents form the basis of much of syntactic theory and represent a word or phrase that functions as an atomic unit within the structure of a sentence. A central idea to *constituency grammar*, otherwise known as *phrase structure grammar*, is that sentences have a hierarchical as opposed to linear structure. Chomsky [1956] formalised this framework by introducing a model that defines the makeup of constituents as well as measurements of its computational complexity. Later works additionally introduced transformational rules as well as *deep* versus *surface* structure [Chomsky, 1957, 1965].

The Minimalist Program

Children are particularly adept at recognising, learning and producing the patterns observed in language. For any given syntactic construction C in language L, it has been observed that children learn C with such speed and consistency that their success is difficult to attribute solely to the linguistic input available to them. This observation underlies the *poverty of the stimulus* argument, which posits that the linguistic data children are exposed to is insufficient to account for their eventual grammatical competence. This additionally gave rise to the idea of an internal 'universal grammar' which is present in all humans and for which certain linguistic parameters are adjusted depending on the language or languages we are exposed to as a child. In its contemporary form, this is captured by Chomsky's *Minimalist Program* [Chomsky, 1993]. This research direction seeks to explain grammatical phenomena using the simplest possible mechanisms, such as a minimal set of operations known as *merge* and *move*, and aims to reduce grammatical theory to its most fundamental computational principles.

Minimalism is intended as a program as opposed to an individual theory. This allows the coexistence of differing or even incompatible theories within the same strand of research, such as Starke-style Nanosyntax and Chomsky's C-T-v-V clause structure [Kobele, 2021]. *Lexical decomposition* is a categorial framework within the minimalist program formalised by Kobele

[2018]. It performs operations on representations of grammars, known as 'Minimalist Grammars' (MGs), in order to find those representations which are optimally decomposed and generalised. This allows a quantitative approach to analysing syntactic theories, for instance [Ermolaeva, 2021] uses a metric which combines the number of phonological and syntactic symbols required to describe the grammar with the aim of minimising this description length.

2.1.2 Dependency Grammar

Dependency grammar, on the other hand, is built on the notion of *heads* and their *dependents*. *Heads* typically represent the most important word in respect to others e.g. nouns in a noun phrase such as *table* in *the big wooden table*. *Dependents* are those words which depend on heads e.g. the adjective *wooden* in *the big wooden table*. Within this framework, language structure consists of directed links from heads to dependents. It emphasises binary relations rather than hierarchical groupings.

One of the most common ways of representing dependency structures is through a dependency tree. Dependency trees are incredibly helpful for understanding both the structure and the syntactic features within a language, making them an important representation for modern language analysis. Such a tree can be viewed in Figure 2.1 with dependencies defining the subject (`nsubj`), object (`obj`), determiner (`det`), and relative clause (`acl:relcl`).

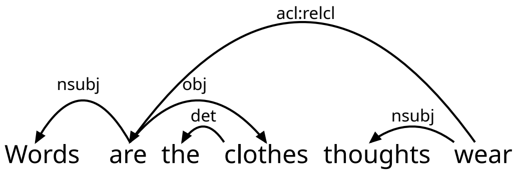


Figure 2.1: A basic dependency tree.

Dependency grammars are typically more robust cross-linguistically than constituency grammars for computational analysis and led to the development of *Universal Dependencies* (UD). This is an extension of the following projects: (1) *'Stanford dependencies'* [de Marneffe et al., 2006, 2014, 2021], (2) *'Google universal part-of-speech tags'* [Petrov et al., 2012], and (3) *'Interset interlingua for morphosyntactic tagsets'* [Zeman, 2008]. It is an open community initiative to create standardised cross-linguistic annotations of language structure within a *lexicalist* dependency-based framework, meaning that syntactic relations are always formed between words. Collections of UD dependency trees are created for languages and released as *dependency treebanks.* For instance, one such treebank is `UD-English-CHILDES` published by Yang et al. [2025b] and amounts to over 48,000 trees representing English speech from children. An example from this treebank for the sentence "I don't know what it is" is shown in Figure 2.2.

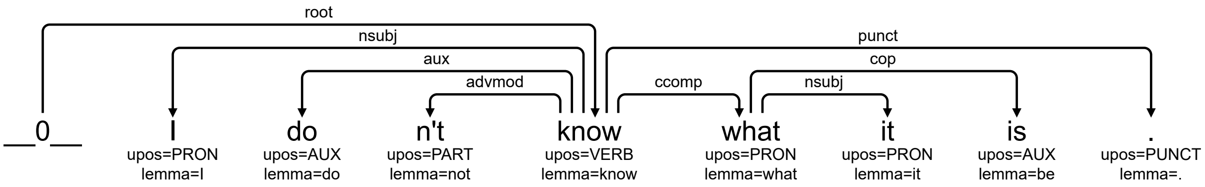


Figure 2.2: A dependency tree for the sentence "I don't know what it is" from the treebank UD English CHILDES.

The properties of a typical UD dependency tree are outlined by Nivre et al. [2020] as follows:

1. Word segmentation
2. Morphological layer comprising lemmas, universal part-of-speech tags, and standardised morphological features.
3. A syntactic layer focusing on syntactic relations between predicates, arguments and modifiers

Furthermore, each word form in a UD tree is further broken down into:

1. A lemma representing the base form.
2. A part-of-speech tag representing the grammatical category.
3. A set of features representing lexical and grammatical properties.

There are now over 290 UD treebanks published across more than 160 languages, a significant feat within the computational linguistics domain.

Dependency Tree Querying

Query languages allow for the extraction of data from a database or dataset with the specification of individual properties about said data. Such languages are crucial within the field of computational linguistics for efficiently exploring large annotated corpora and identifying patterns that correspond to syntactic phenomena. They have been particularly powerful when applied to UD treebanks. `Grew` is one such graph-based query language [Guillaume, 2021b, 2019] that treats each dependency tree as a graph with *nodes* (typically whole words) and *edges* (relations between words). Queries specify graph patterns that must be matched within the treebank, making it more flexible than traditional linear matching as it naturally lends itself to querying their relational structure. The querying of treebanks is important for research into specific phenomena as well as improving annotation consistency and error correction. For instance, the following `Grew` query can isolate transitive sentences:

```
pattern {
  V [upos="VERB"];
  SUBJ [upos="NOUN", case="Nom"];
  OBJ [upos="NOUN", case="Acc"];
  V -[nsubj]-> SUBJ;
  V -[obj]-> OBJ;
}
```

Extensions of this language such as `Arborator-Grew` [Guibon et al., 2020] allow for easier creating, updating, maintaining, and curating syntactic treebanks and semantic graph banks through an interactive web environment. Other such older query languages are `TigerSEARCH` [Lezius et al., 2002] and `Tregex` [Levy and Andrew, 2006]. These tools make it easier for researchers to collaboratively maintain large-scale treebanks and perform large-scale linguistic analyses.

2.1.3 Syntactic Complexity

One of the reasons that we represent natural language structure at all is in order to measure and compare its inherent complexity. The simplest measure of such complexity may be a simple-sentence length metric measured in number of words. However, it is generally agreed that a sentence such as *"the boy goes to school every morning with his friends"* is less syntactically complex than *"the boy that saw the bully quickly ran"*, despite the former having more words. In fact, *garden-path sentences* are examples of sentences that native speakers require significantly longer to process regardless of their word length e.g. *the old man the boat* where *man* is the verb.

The complexity of syntactic structures tends to be broadly agreed upon on across theoretical frameworks. The definition and quantification of this complexity varies significantly however. Subordinate clauses are typically considered more complex than simple noun or verb phrases, but the *reason* given for this will change. Within generative grammar, a greater number of derivational steps, i.e. move / merge operations, may be associated with a more complex construction. If we consider the process of lexical decomposition, we may use another metric to define complexity, such as *Minimum Description Length*, or MDL. The MDL of a grammar is the smallest possible representation of that grammar using this formalism that still has its full generative capacity. Within a dependency-based framework one typically looks towards dependency length or tree depth [Futrell et al., 2015]. The inherent "difficulty" of parsing UD treebanks themselves have also been used as such a complexity rating, such as through dataset cartography, ν-information, and minimum description length [Kulmizev and Nivre, 2023].

Measuring complexity from a psycholinguistic perspective involves tracking eye movement to measure parsing times and processing difficulty [Clifton et al., 2007, Rayner and Clifton, 2009]. A more difficult syntactic construction may, for instance, have a long dependency and therefore require more memory. The surprisal [Shannon, 1948] of a word w_t is a metric linked to the probability of its occurrence given a context C, where a high surprisal indicates that a word is less likely to appear and a low surprisal that a word is more likely to appear within that context. Garden-path sentences are defined by their high surprisal due to a present misleading word. Surprisal is an important concept for this work due to its relevance in an LM prediction confidence and will be further explored in Section 4.3.

2.2 Sequences, Attention & Transformers

In this section we introduce work that has led to major advances in LM performance and represents an important inflection point in the field of *natural language processing* (NLP). Section 2.2.2 gives an overview of the concept of *tokenisation* with a description of a number of common approaches and challenges, with *word embeddings* discussed in Section 2.2.3. We introduce the *sequence-to-sequence* architecture in Section 2.2.4 and expand on it with the *Transformer* and its variants in Sections 2.2.5, 2.2.6, and 2.2.7. Lastly, in Section 2.3 we discuss the current landscape of the evaluation of language models on syntactic tasks.

2.2.1 Connectionism

Connectionism, as it pertains to machine learning, describes a methodology that relies on the learning of the patterns of a set of data through what are known as *perceptrons* [Rosenblatt, 1958] or *neurons*, though this latter term is sometimes reserved for perceptrons that can be interpreted as models of a single data property e.g. the temperature in a weather report. Figure 2.3 shows a perceptron that consists of an *input*, *weights*, and summation and activation functions that result in the *output*. A group of a neurons can be termed a *layer*, and a set of layers makes up what we term a *neural network* (NN). A simple NN typically consists of an *input* layer, one or more *hidden* layers, and an *output* layer. A *feed-forward* neural network describes a network where information flows in a single direction and can be observed in Figure 2.4. This stems from work carried out in modelling the neuronal activity in the brain [McCulloch and Pitts, 1943] and was adapted to the machine learning domain through vital concepts such as the updating of weights through *backpropagation* [Rumelhart et al., 1986].

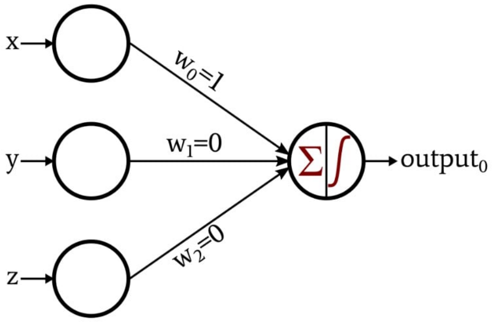


Figure 2.3: A perceptron.

Basic feed-forward NNs work well for modelling datasets of one or more continuous variables, such as the famous ‘Iris flower’ dataset [Unwin and Kleinman, 2021] that describes 50 samples of the multiple types of Iris flower (Iris setosa, Iris virginica and Iris versicolor). The length and width of the sepals and petals were measured in each case. One can train a simple neural network that learns to predict which type of Iris a given flower is when provided these four features as input. These models however are too simple to model the complexity of natural language. Furthermore, it is not obvious how one should pass a word or subword as input to a model as a continuous value. In the coming sections we will look at recent progress that has been made in this domain through *tokenisation*, *word embeddings*,

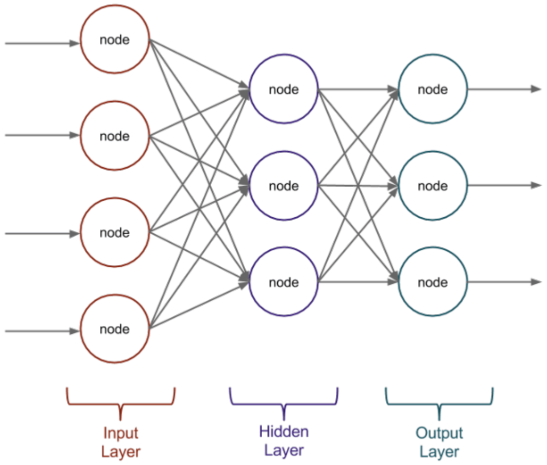


Figure 2.4: A Feed-Forward Neural Network

and the development of the *transformer* architecture.

2.2.2 Tokenisation

We now discuss *tokenisation*, that is, the splitting of text into *tokens* such as words or subwords. This is the first of a number of key differences when comparing classical machine-learning models that are designed to learn continuous variables such as temperature or categorial variables such as ethnicity with models for language comprehension and production. The granularity of the tokens can vary depending on the algorithm but typically correspond to linguistically meaningful units such as whole words or morphemes e.g. *car*,*be-*,*anti-*, or they can also correspond to substrings that don't necessarily represent a linguistically atomic unit e.g. the *nd* in *and*, *induction*, and *band*. Some tokenisers may even tokenise into individual characters. Early NLP systems used rule- or dictionary-based systems in order to carry out tokenisation. For instance, the Penn Treebank introduced a hand-crafted tokenisation scheme for English [Marcus et al., 1993] with a number of examples shown below:

`can't` → [`ca`, `n't`]

`he's` → [`he`, `'s`]

`they'll` → [`they`, `'ll`]

`co-operate` → [`co`, `-`, `operate`]

`John's` → [`John`, `'s`]

Stanford's tokeniser, on the other hand, used regular-expression rules for splitting based on spaces and punctuation with specific rules for specific cases such as abbreviations or parsing numbers. These tokenisers work well for individual languages but require a significant amount of manual effort and struggle with tokenising words that haven't been specified or accounted for.

Cross-linguistically this task can also present a number of challenges. In languages with spaces between words (e.g. English, German, Spanish) tokenisation is made relatively easy despite some complications such as compound words or punctuation. In languages without spaces (e.g. Chinese, Japanese, Thai) segmentation requires access to a lexicon or a statistical-based parser analyse frequently occurring patterns. Agglutinative languages, that is languages with long, complex word forms that continuously append morphemes such as Turkish or Kazakh, may require significant morphemic tokenisation for each word.

Statistical Tokenisation

Modern NLP models typically avoid using pre-determined rules or lexicons and instead rely on language-agnostic statistical algorithms such as Byte-Pair Encoding (BPE) [Sennrich et al., 2016], WordPiece [Schuster and Nakajima, 2012], or SentencePiece [Kudo and Richardson, 2018]. One of the most widely adopted tools for statistical tokenisation was `Moses` [Koehn et al., 2007], designed with the intention of aiding in statistical machine translation. However, this is not as commonly used in modern LMs. BPE was originally an algorithm for data compression that was adapted for tokenisation [Gage, 1994]. It begins with each character as an individual token and merge the most frequent co-occurring characters into larger tokens. This encourages more frequent words to be a single token and rarer words to be subdivided into multiple tokens, reducing the issue of tokenising words that have not been seen before. `WordPiece` follows a similar process but instead of merging frequently co-occurring tokens, it merges tokens that most improves a given language model. This requires more effort due to the training of an additional model, however it has been used very effectively for instance with `BERT` [Devlin et al., 2019]. The `WordPiece` tokeniser marks subwords with a ## symbol:

`playing` → [`play`, `##ing`]

`nationality` → [`nation`, `##al`, `##ity`]

Lastly, `SentencePiece` doesn't assume that text can be pre-divided through whitespace and is thus particularly suited for languages that don't make use of spaces such as Chinese or Japanese. It treats text as a stream of characters and whitespace is simply counted as another character to be represented, thus resulting in whitespace information being kept as below:

`playing` → [`_play`, `ing`]

`nationality` → [`_nation`, `al`, `ity`]

Each language model has some form of tokeniser that carries out the initial tokenisation step. We can see examples of the tokeniser for various the multilingual model `Multilingual BERT` [Devlin et al., 2018] as well as the monolingual Polish models `PolBERT` [Kłeczek, 2020], `SlavicBERT` [Arkhipov et al., 2019], `HerBERT` [Mroczkowski et al., 2021], `DistilBERT-PL` [Abdaoui et al., 2020], `RoBERTa-PL` [Dadas et al., 2020b], and `TrelBERT` [Szmyd et al., 2023] in Table 2.1.

2.2.3 Word Embeddings

The problem still remains however, *how can we best represent these tokens in a neural network when they work solely with continuous or discrete numbers?* The answer comes in the form

PolBERT	SlavicBERT	Multilingual BERT	HerBERT	DistilBERT-PL	RoBERTa-PL	TrelBERT
WordPiece	Shared WordPiece	Shared WordPiece	CharBPE	WordPiece	SentencePiece BPE	CharBPE
ulot ##ek	ulo ##tek	ul ##ote ##k	ulotek</w>	ul ##ote ##k	ul otek	ulotek</w>
przeprowadzenia	przeprowadzenia	pr ##ze ##pro ##wadzenia	przeprowadzenia</w>	pr ##ze ##pro ##wadzenia	prze prowadzenia	przeprowadzenia</w>
zarzutów	zarzutów	za ##rz ##ut ##ów	zarzutów</w>	za ##rz ##ut ##ów	za rzut ów	zarzutów</w>
telefonu	telefonu	telefon ##u	telefonu</w>	telefon ##u	telefon u	telefonu</w>

Table 2.1: Tokenisation of sample Polish genitive nouns.

of an information-encoding vector. A vector can be thought of as an arrow within space that points in a particular direction and has a certain size, typically represented as $\vec{v}$. A vector may have any number of dimensions and represents an arrow within a n-dimensional plane. For instance, the below vector creates an arrow that is running only along the x axis:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{2.1}$$

while the following vector represents an arrow running along the y axis:

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.2}$$

Vectors are fundamental to neural language models as they are the only representation that a model can take as input. The model does not 'see' the word as we know it but rather sees what is known as a *word embedding.* These are vectors that can have any number of dimensions and are a vectorised representation of a given token. Ideally, each dimension encodes in its magnitude the strength of the connection between a given token and a given characteristic. In an idealised case where each dimension corresponds to a single interpretable feature, the relationships between embeddings can exhibit interesting algebraic properties. This is illustrated in Equation 2.3, where $E(t)$ represents the embedding vector for a token t.

$$E(Stalin) + E(Italy) - E(Russia) \approx E(Mussolini) \tag{2.3}$$

This can also be applied to syntactic components of language, for instance plurality as shown in Equation 2.4.

$$\vec{plur} \approx E(cats) - E(cat) \tag{2.4}$$

Word embeddings thus capture key aspects of both word *meaning* and *structure* in a numerical representation, a property that makes them suitable as inputs to a neural network. Furthermore, models may also learn to manipulate word embeddings so as to add in additional contextual information through adjusting particular values. For instance, a classic thorny issue in NLP is to distinguish between polysemous words such as ‘bank’ in the sense of a river or in the sense of money. Contextualised word embeddings thus allow the encoding of the properties of a word in its surrounding context. This will be discussed further in Section 2.2.4.

2.2.4 Sequence-to-Sequence

Encoder-decoder sequence transduction [Sutskever et al., 2014, Cho et al., 2014], or sequence-to-sequence (seq2seq) models, represented the first stage of rapid advancements in LMs with particular improvements in machine translation. A sequence of word embeddings is provided as input to a model, and these word embeddings undergo some arrangement of operations and transformations through the layers of a neural network, where finally another sequence of word embeddings is given as output. The original sequence is therefore made up of a matrix, as seen in Equation 2.5, with each row representing a different entire embedding and each column representing a particular dimension of all embeddings.

$$X = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,d} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ e_{T,1} & e_{T,2} & \cdots & e_{T,d} \end{bmatrix} \tag{2.5}$$

In some seq2seq models, the embeddings will be passed individually one-by-one and each next token represents a later "time" t in the process. In others however, they are processed simultaneously. One such model architecture that performed particularly well was that of Recurrent Neural Networks (RNN) introduced by Cho et al. [2014], the Long Short-Term Network [Sutskever et al., 2014] being a successful variant. In an RNN the embeddings are passed through the network one-by-one with the current embedding representing time step t and each new embedding representing $t + 1$. A hidden state vector h_t is stored which keeps track of all information seen up until time / embedding step t. Once the embeddings have been passed through the initial part of the model, known as the *encoder*, the final hidden state vector h_T should then have stored all of the "information" of the sentence and is passed then to the next stage of the model known as the *decoder* to create the final output sequence. A typical application of an RNN is in machine translation, where for instance a sequence of English word embeddings are translated into a corresponding sequence of German word embeddings. Note that in an LSTM architecture there are two states recorded: the hidden state vector h_t as well as the individual cell state vector c_t, representing short-term memory and long-term memory, respectively. The LSTM architecture is shown in Figure 2.5.

Attention

This passing of each embedding one-by-one led to poor retention of information from embeddings that were seen earlier in the sequence. This provided a significant bottleneck for the

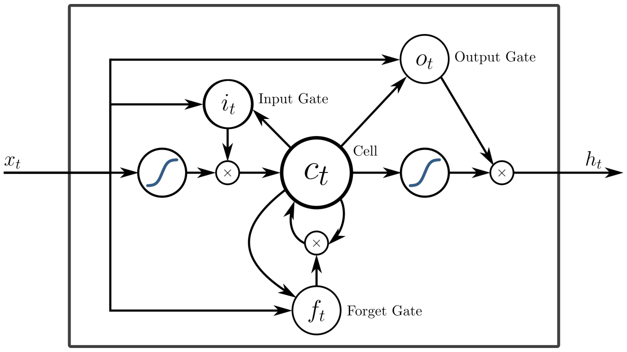


Figure 2.5: Long Short-Term Memory Architecture.

architecture and led to the development of an attention mechanism called *RNNSearch* [Bahdanau et al., 2016] that would allow the decoder to search back through the input sentence for relevant key words when predicting each word in the output sequence in a bi-directional type of mechanism. This bi-directional LSTM architecture would later be implemented for Google Translate [Wu et al., 2016]. This was the first time the concept of attention gained significant traction in the field and further architectures were quickly developed such as memory networks [Weston et al., 2015], visual attention [Gregor et al., 2015, Jaderberg et al., 2016], hierarchical attention [Yang et al., 2016], and attention for machine comprehension [Seo et al., 2018]. We will return to this concept later.

2.2.5 Transformers

The transformer architecture was introduced in a landmark research paper [Vaswani et al., 2017] that brought about seismic changes in the domain of text comprehension and generation. At its core, it comprises a set of floating point matrices and a pipeline of operations on those matrices. However, the way in which these are set up allow the model to take into account *context* in a much more robust manner than seen previously. The prior state-of-the-art of recurrence-based [Hochreiter and Schmidhuber, 1997, Cho et al., 2014, Sutskever et al., 2014] or convolutional-based [Kalchbrenner et al., 2017, Gehring et al., 2017] models struggled to integrate a more expanded window of context into their language comprehension and production, but later developed attention-based add-ons that somewhat eased this issue. The major innovations with the transformer architecture were (1) multi-head attention (explored in Section 2.2.5) being its *core* mechanism and (2) parallel computation as opposed to sequential. This led to significant speed-ups in computation time due to Graphics Processing Units, or GPUs, being aptly suited for this type of computation on large matrices. Soon after, the first report from OpenAI on the Generative Pre-trained Transformer (GPT) model was released [Radford and Narasimhan, 2018], where researchers focused on training the model on continuous next-word prediction given a previous context window that allowed the generation of texts for as long as necessary. The *neural scaling laws* were introduced by Kaplan et al. [2020] where they showed that model performance follows predictable power-law relationship with model size, dataset size, and amount of available processing power. This led to the beginning of a rapid expansion of model development and a paradigm shift in the field of LMs.

Architecture

The full encoder-decoder transformer architecture is shown in Figure 2.6. It encodes a sequence into an abstract representation producing from it a new sequence and is a standard architecture for modern machine translation. A number of popular encoder-decoder models are Google's T5 [Raffel et al., 2023], BART [Lewis et al., 2019], or MarianMT [Junczys-Dowmunt et al., 2018].

We can see a number of familiar components such as modules for input word embeddings as well as feed-forward neural networks. The most important novel module is the multi-head attention block which we will discuss next.

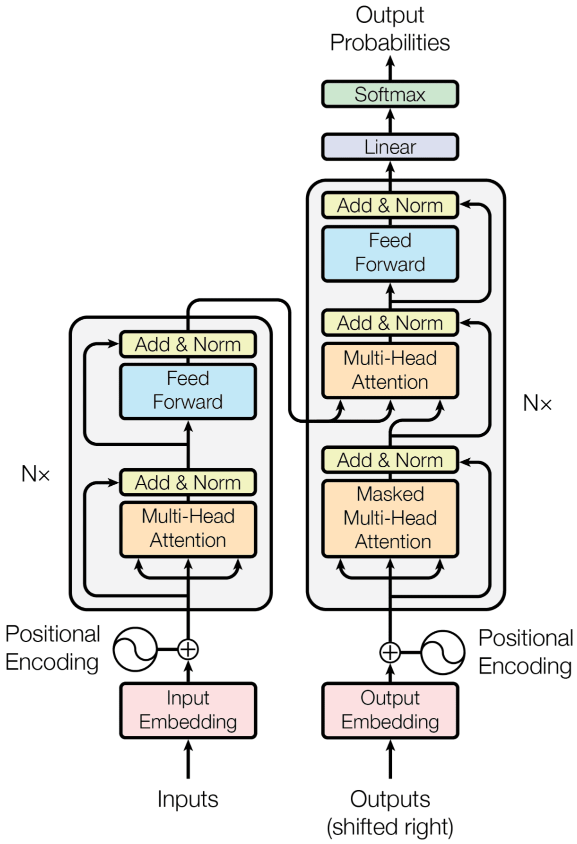


Figure 2.6: The encoder-decoder transformer architecture introduced by Vaswani et al. [2017].

Multi Head Attention

One of the key components of the transformer architecture is known as *multi-head attention*, visualised in Figure 2.7. It is made up of h attention 'heads'. These heads have two distinct advantages: computation of attention can be run in parallel and each head can focus on different aspects of the input. This introduces a component that specialises in incorporating various aspects of the context into its final answer and has led to significant improvements in the results of LMs. For instance, one head may focus on local verb agreement and another on long-distance dependencies.

The formula for a single one of the h attention heads can be seen in Equation 2.6 and results in a set of weights that ranks how important each word is in answering a particular

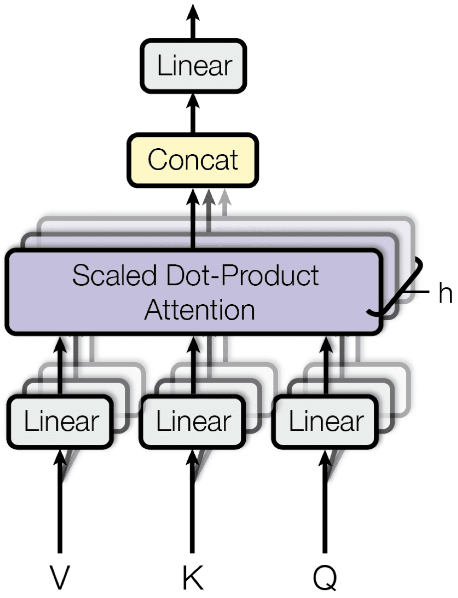


Figure 2.7: Multi-Head Attention Block.

query. We say that a query *attends* to the relevant keys in the input sequence.

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.6}$$

Q represents the *query* matrix indicating the type of question that is being asked by this head, K represents the *key* matrix representing the context that is available to be used in the answer, and V represents the *value* matrix which matches the inputs in K most valuable for answering the question in Q. *Softmax* is an activation function that converts the output vector of raw attention values to a probability distribution. That which any given attention head pays attention to is determined through the model pre-training process.

Hyperparameters

Hyperparameters refer to those parameters that can be adjusted prior to pre-training or fine-tuning. They will typically determine important features of both the training process and results. A number of important general hyperparameters which determine the model architecture itself are the number of layers, input embedding dimensions, and attention heads. We also may choose the dimension of the hidden layers. For pre-training, we make adjustments that affect the training process itself such as the learning rate, batch size, weight decay, and dropout. The tokeniser also plays an important role as we may choose its maximum size e.g. approximately 30,000 tokens in the case of BERT (see Section 2.2.6).

2.2.6 Encoder-only

The encoder-decoder transformer can be further subdivided into its components to form new architectures for different tasks. An *encoder-only* architecture comprises the left portion of the encoder-decoder diagram and can be observed in Figure 2.8 and specialises in converting input text into an abstract representation. This is typically used in order to create word embeddings (see Section 2.2.3) which include contextualised information within each embedding. The BERT model [Devlin et al., 2019], or the less graceful *Bidirectional*

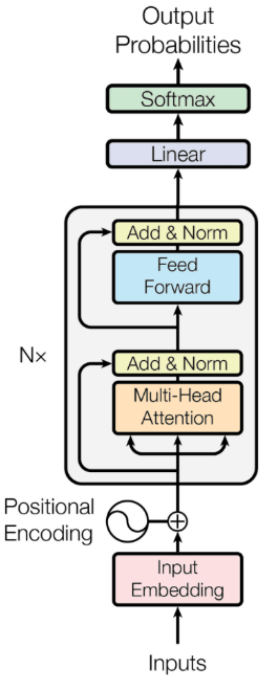


Figure 2.8: Bi-directional Encoder Representations from Transformers Architecture (BERT)

Encoder Representations from Transformers, is an encoder-only model that was trained for the task of *Masked Language Modelling* (MLM) and *Next-Sentence Prediction* (NSP). The MLM task comprises the removal of one or multiple tokens from a sentence (known as the masked tokens) and the prediction of either the most-likely token (token-level masking) or the most-likely word (word-level masking [Cui et al., 2021]) for that position in the sentence. This task can be observed for the `RoBERTa` model, a variant of the original `BERT` model, in Figure 2.9 [Weyssow et al., 2022].

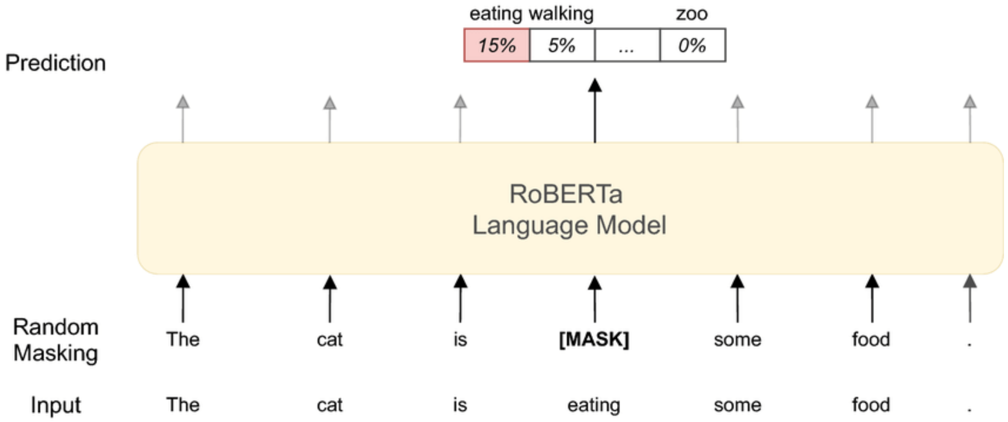


Figure 2.9: A RoBERTa encoder-only model, a variant of `BERT`, performing the task of MLM.

The entire unmasked input is provided as context for each prediction, allowing for more information to be taken into account than will be the case in next-token prediction tasks (more on this in Section 2.2.7). The NSP task, on the other hand, has the model determine whether or not one sentence naturally follows another. An example from the Wikipedia entry for the city of *Leipzig* is provided below:

`Input:` “Leipzig has been a trade city since at least the time of the Holy Roman Empire.”

`Test:` “Via Regia and the Via Imperii, two important medieval trade routes, intersected here, marking the city’s economic importance.”

`Desired Model Output: isNext = True`

The NSP task was later dropped for models such as `RoBERTa` due to findings that it was not particularly beneficial and sometimes even harmful to performance [Liu et al., 2019, Conneau et al., 2019].

2.2.7 Decoder-only

Decoder models take an input, typically known as a *prompt*, and generate some form of output sequentially. They are *autoregressive*, meaning that at each step n, they predict the next most-likely token given all input and outputs produced from the first step until $n-1$. We can see this below for some sequence of tokens $x_1, x_2, \ldots, x_n$.

$$P(x_1, x_2, \ldots, x_n) = P(x_1) \cdot P(x_2 \mid x_1) \cdot P(x_3 \mid x_1, x_2) \cdots P(x_n \mid x_1, x_2, \ldots, x_{n-1})$$

One of the most ubiquitous decoder models is known as a *Generative Pre-trained Transformer* (GPT). These are also autoregressive and thus attempt to find the most likely sequence like so:

$$P(\text{I love Käsespätzle}) = P(\text{I}) \cdot P(\text{love} \mid \text{I}) \cdot P(\text{Käsespätzle} \mid \text{I love})$$

The architecture for this is shown in Figure 2.10. These models are typically trained for the task of *Next-Token Prediction* (NTP). They will autoregressively generate new tokens that they deem the most likely given the input and any tokens previously generated. This model architecture makes up a large proportion of modern LLMs such as the GPT collection, Llama, and Gemini.

2.3 Syntactic Evaluation of Language Models

The question that naturally arises from the pairing of Section 2.1 and 2.2 is that of how well transformer-based architectures perform in the comprehension and production of the structure of language. Do they use the correct affixes in the right places? Do we observe the

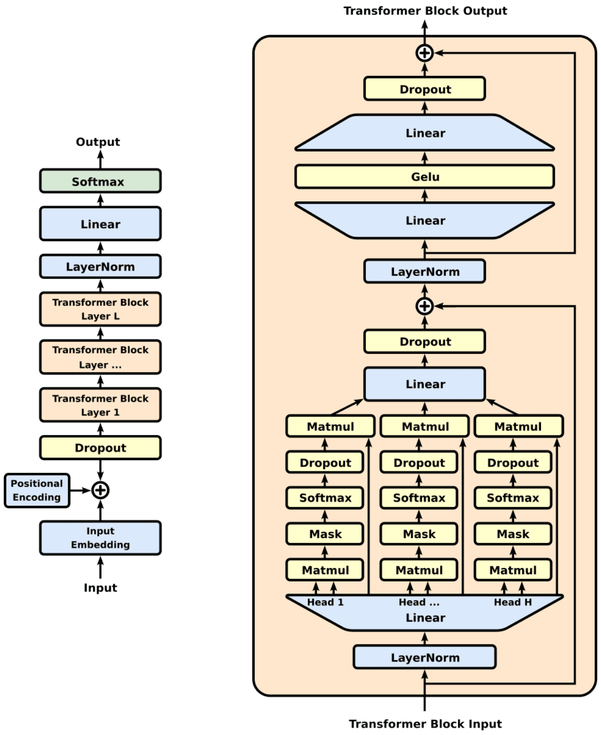


Figure 2.10: Generative Model Architecture (Decoder Only)

suitable assignment of case or order of words? Models that aim to comprehend and produce language naturally need to absorb an understanding, in whatever form that may take, of natural language syntax. The production of language appears to require some representation of its structure within the producer and language models are no exception. However, there is little consensus on what this representation looks like. We can imagine a scenario where models store each item in the lexicon separately with little to no generalisation or another where generalisation has taken place through some form of lexical decomposition. Progress has been made in both areas.

Interpreting Internal States

The internal state tells us about the *efficiency* of a model's acquisition of syntax. We can, for instance, examine whether there is a specific group of neurons that lights up in response to a particular syntactic feature and whether those same neurons light up for other features. This internal approach gets to the core of analysing LM syntactic representations, however the difficulty of interpreting even the most primitive of deep neural networks makes this an exceedingly difficult task. Local syntactic agreement has been found to be at least partially modelled within the initial model layers [Mueller et al., 2022] and more complex syntactic constructions in the middle layers [Jawahar et al., 2019, Lin et al., 2019]. Hewitt and Manning [2019] find evidence of dependency trees modelled deeply in the vector geometry of such models. This corroborates findings that suggest that `BERT` models learn more complex syntactic generalisations later in the training process [Warstadt et al., 2020b], typically reaching a consistent point where they undergo a phase transition to learning deeper language structure [Chen et al., 2025]. *Polysemanticity*, the compression of multiple features into individual neurons in a neural network, has been found to potentially be the reason that LLMs are so uninterpretable [Anthropic, 2025]. The usage of *sparse autoencoders* has brought progress in unravelling these features and allowing the creation of tools such as `Neuronpedia` that allow exploration of the internal representations of LLMs [Lin, 2023]. It is easy to see how this may lead to a greater understanding of the types of syntactic representations these models may be storing, however this type of exploration is still in its early stages.

Interpreting Probability Distributions

On the other hand, we can simply look towards the performance of a model on syntactic tasks and not concern ourselves with the internal representation. In this work we will focus primarily on this type of evaluation for syntax, with little attempt to attest to any contributions in the previous domain. That is, the understanding of the internal workings of neural networks. We will focus rather on the final probability distributions that are produced by these models, assigning a likelihood to each word or subword in a model's vocabulary given a particular context. However, to some extent there are means here to test the internal generalisation of these models through out-of-distribution tests, a topic that will be explored in

Section 3. We would like to know, firstly, whether a model chooses a pre-defined grammatical over ungrammatical word given a context, secondly, whether a model tends to distinguish grammatical over ungrammatical constructions generally, and finally, how much confidence it has in its own predictions about grammaticality. Through the probability distribution alone, we can shed light on all three key questions. However, there are a great deal of caveats that come with this that will be explored in the coming sections. Semantic as opposed to syntactic benchmarks have traditionally reigned here such as well-established the `GLUE` benchmark [Wang et al., 2019]. This is a common test of primarily semantic understanding that includes grammatical acceptability judgements, sentiment analysis, paraphrase identification, sentence similarity, various forms of logical entailment, and pronoun resolution. Leaderboards for these benchmarks give rankings of many models' performance and some insight into the types of models that seem to have absorbed semantic and/or syntactic properties "better". We will focus however purely on syntactic benchmarks and not delve deeper into the semantic performance of the models in question.

2.3.1 Targeted Syntactic Evaluation

Targeted Syntactic Evaluation, or TSE, was defined by Marvin and Linzen [2018] and introduced the focus on *grammatical* versus *ungrammatical* minimal pairs into the space of syntactic evaluation of language models. They focused on a number of phenomena such as subject-verb agreement and reflexive anaphora and evaluated the state-of-the-art at the time: Long Short-Term Memory (LSTM) models. For instance, Example 1 shows agreement across a subject relative clause defined as a grammatical versus ungrammatical minimal pair.

(1) **Original:** The officers that chased the thief run.

Test: The officers that chased the thief _.

Model assigns p(*run* | Test) > p(*runs* | Test) ✓

We can define this pair as the pair (ℓ_+, ℓ_-) representing the grammatical and ungrammatical form given the context and having a single lemma ℓ differing only in its syntactic features 3PL and 3SG. Newman et al. [2021] refined the goals of TSE further by focusing not just on how models fared with the most likely lemma for a given sentence, but also expanded to include a set of lemmas $\ell_1, \ell_2, ..., \ell_n$. For instance, it was observed that a model assigned the correct conjugation for the most-likely verb but the incorrect conjugation for a less likely verb, as seen in Example 2.

(2) **Test**: The keys to the cabinet $_o$n the table.
Model assigns p(*are* | Test) > p(*is* | Test) ✓
Model assigns p(*exist* | Test) > p(*exists* | Test) ✓

This disparity shows that while a model may perform well in predicting the correct form for *some* lemmas, this does not prove that it has fully generalised the syntactic construction across *all* lemmas. A primary additional finding was that the model architecture appears to be much more important than dataset size. The establishment of minimal pairs as an important test of a language model's syntactic capabilities was a key focal point leading to the creation of standardised test suites for specific phenomena. Recent work by Someya et al. [2024] has investigated how well LSTM, transformer-based and Stack-RNN models recognise linguistic phenomena situated at different levels on the Chomsky hierarchy. That is, *regular*, *context-free*, and *(mildly) context-sensitive constructions*. They found that despite the models being able to generate complex syntactic constructions they performed poorly in recognising nested or cross-serial dependencies, suggesting that these models may not have an understanding of syntactic dependencies. Hu et al. [2020] expanded on this to look at a greater number of models and constructions with an integration of theoretical syntax classifications, analysing how well language models generalise in English and what determines improved generalisation. There is however a lack of work in this area for the vast majority of low-resource languages (henceforth LRLs), as they rely largely on manual human annotation.

2.3.2 Benchmarks of Linguistic Minimal Pairs

The *Benchmark of Linguistic Minimal Pairs*, or BLiMP [Warstadt et al., 2020a], is a set of 67 datasets of 1000 sentences each that evaluates model performance on different syntactic phenomena in English. It still remains one of the most important datasets in the TSE domain, as it provided the first large dataset for English minimal pairs across the syntactic domain. Similar evaluation sets have been compiled since this contribution for other majority languages such as German [Zaczynska et al., 2020a], Spanish [Bel et al., 2024], Chinese [Song et al., 2022, Wang et al., 2021], Russian [Taktasheva et al., 2024a], and Japanese [Someya and Oseki, 2023]. Additionally, recent efforts have been made to expand this research to languages that have been typically less studied such as Turkish [Başar et al., 2025], Swedish [Volodina et al., 2021], Icelandic [Zhang et al., 2024], and Dutch [Suijkerbuijk et al., 2025].

MultiBLiMP 1.0

There is still only a small number of studies carried out that evaluate the syntax of LRLs. A significant contribution in this domain is that of `MuiltiBLiMP 1.0` [Jumelet et al., 2025]. This is a collection of datasets covering 101 languages and over 128,000 minimal pairs for cross-linguistic syntactic evaluation. They contain tests to evaluate `subject-verb` and `subject-participle` agreement across `number`, `person`, and `gender`. 42 decoder-only language models were benchmarked and ranked by their performance. The authors found that model size is particularly important with a larger model typically improving performance when controlled for other variables, as well as how well a language is represented within the pre-training corpora. Furthermore, monolingual models tend to perform better for LRLs than multilingual models. Such a large collection of datasets was created through the use of UD treebanks as we use in this work. However, they do not use a query language in order to isolate particular syntactic constructions but rather perform this programmatically, resulting in more difficulty in generating datasets dynamically for different syntactic constructions. This additionally makes it more difficult for linguists to specify particular constructions or generate their own datasets for evaluation.

2.3.3 Metrics

Here we will introduce key metrics in the TSE literature. We should particularly note that the *accuracy* metric is defined differently than in other fields, as we will discuss further in the coming section. We will additionally look to information theory in order to expand on previous work by arguing for the incorporation of surprisal and entropic metrics in our evaluation of not just a model's *performance* but also its *confidence* with regards to its predictions.

Accuracy

The most basic unit of measuring performance is simply *accuracy*, defined as the proportion of answers that a model gets correct in a test set. Typically, the notion of accuracy in machine learning relates to discrete or categorical values. The value with the highest probability is taken to be the final prediction $\hat{y}_i$ and that is compared with the ground truth y_i. We can see this defined in Equation 2.7.

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left[\hat{y}_i = y_i\right] \tag{2.7}$$

However, in TSE, we are not comparing one label with another but rather continuous variables i.e. probabilities [Marvin and Linzen, 2018]. We determine therefore that the model is correct when, for a minimal pair (w_{i_a}, w_{i_b}) and context C_i, the model has assigned a higher probability to w_{i_a} than to w_{i_b}. We can see this formally defined in Equation 2.8.

$$\text{Accuracy}_{\text{TSE}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1} \Big[p(w_{a_i} | C_i) > p(w_{b_i} | C_i) \Big] \tag{2.8}$$

Measuring accuracy like this may not take into account the highest-probability token from a model inference as is typical in other domains. Rather, it compares two tokens that have been pre-defined for the minimal-pair syntactic test. This is a method that may deserve criticism [Newman et al., 2021], however we will not explore this topic deeply in this work.

Surprisal

In the mid-1900s, American mathematician Claude Shannon introduced the fundamental unit of information theory as *surprisal* [Shannon, 1948]. If an event has a high probability of occurring, then we can say that our 'surprise' at that outcome should be low. Conversely, an event with a low probability of occurring should be more surprising. There is therefore a strong link between the *probability* of it occurring and our *surprise* at hearing that it will happen. In fact, it is simply this probability translated to a logarithmic scale of base b, as defined in Equation 2.9. The base is typically 2 to convert the measurement to bits.

$$I(x) = -\log_b P(x) \tag{2.9}$$

For each increment in the surprisal, the probability assigned is halved. For instance, a surprisal of 0 indicates a probability of 1, a surprisal of 1 equals a probability of $\frac{1}{2}$, a surprisal of 2 equals a probability of $\frac{1}{4}$, and so on. Therefore, if a model has a surprisal of x for w_1 and $x{+}1$ for w_2, it considers w_2 half as likely to be a suitable choice of word given a context.

The *Average Surprisal Difference*, or ASD, measures a model's surprisal value for the pre-determined ungrammatical syntactic feature in comparison with that of the grammatical syntactic feature and averages this over all predictions. This can be seen in Equation 2.10.

$$ASD = \frac{1}{N}\sum_{i=1}^{N}\left(I^{(i)}_{\text{Ungrammatical}} - I^{(i)}_{\text{Grammatical}}\right) \tag{2.10}$$

A higher value indicates that a model, on average, shows stronger preferences for the grammatical item than the ungrammatical item. However, this metric can be subject to biases due to outliers. An ASD value of 1, for instance, means that the model tends to consider the grammatical word twice as likely to be correct as the ungrammatical word, and an ASD value of 2 indicates that it is four times as likely. Conversely, a value of -1 indicates the opposite relationship. It is therefore desirable to achieve a higher ASD. An ASD score is most suitable for single-model comparison rather than cross-model comparison. This can be visualised as in Figure 2.11.

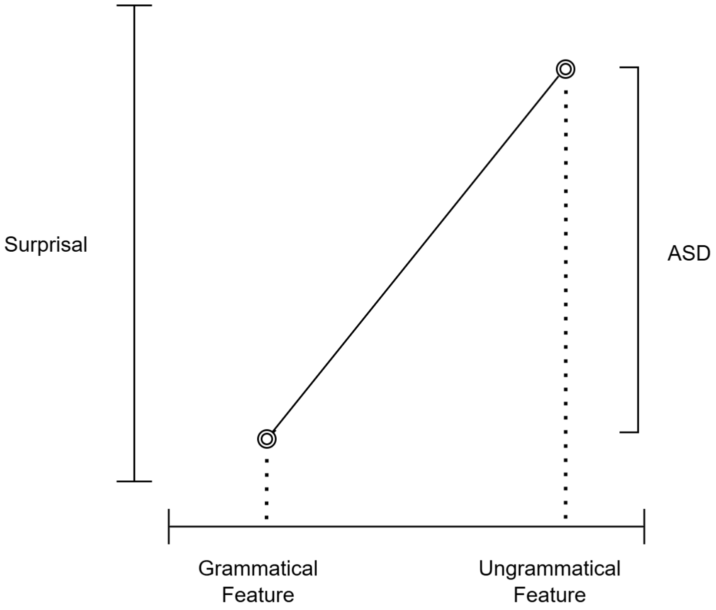


Figure 2.11: ASD can be visualised using a grammatical-ungrammatical feature comparison.

Entropy & Certainty

The building block of surprisal led to the notion of the *entropy* of a system. Given a set of events that can occur in a given system, this measures the amount of uncertainty around which events will occur in said system. We can think of each event as having its own probability of occurring, and the set of these probabilities as a *probability distribution.* If this probability distribution is *flat* - that is, each event has about the same likelihood of occurring - then there is *high* entropy in the system due to a high *uncertainty* about what's going to happen. If the distribution has some high probabilities and a lot of lows, then this is a *low* entropy system. The formula for entropy over a probability distribution p can be seen in Equation 2.11, keeping to the traditional H identifier. For each prediction that a language model makes, a probability distribution is created representing the probabilities of any given token being the 'best' given the context. An autoregressive model such as Anthropic's Claude or OpenAI's GPT range takes all tokens prior to the one being predicted as context and predicts the next, while a bi-directional encoder model such as BERT takes all tokens in a sentence except one masked token located at any position and predicts this masked token. The entropy of these token distributions tells us how *certain* or *uncertain* a model is about a given prediction given this aforementioned context. A probability distribution with high entropy is saying 'well, it could be any of these tokens...', while one with low entropy might be saying 'it's just gotta be this one!'.

$$H[p] = -\sum_{i=1}^{n} p_i \log p_i \tag{2.11}$$

It has previously been the case in syntactic evaluation research to report simply the *accuracy* of the model on a masked, minimal-pair task. However, we argue that throwing out this entropic dimension removes valuable information about the model's *certainty* in its predictions and hence its own syntactic capabilities. A similar logic has been applied to statistical classifiers [Tornetta, 2021], from which we've taken the notion of an *entropy score.* This is a normalised measure of model certainty in the domain of [0,1] as can be seen in Equation 2.12, where higher represents a higher certainty and lower represents a lower certainty. In this case, the value of n would be the model vocabulary size i.e the number of tokens that it can choose from. The name *entropy score* is slightly misleading, as a higher score means lower entropy for a prediction, therefore it will henceforth be referred to as the *certainty* of the model.

$$h = 1 - \frac{E[H[p]]}{\log n} \tag{2.12}$$

2.3.4 Criticism

There are two primary criticisms of TSE research: (1) there has been a lack of clarity with respect to applicable research questions as well as the hypotheses and conclusions that researchers form [Abdou et al., 2022, Kulmizev, 2023], and (2) there has been limited research on LRLs.

The first issue has led to what has been described by Kulmizev and Nivre [2021] as a *Schrödinger's Cat*-type situation, where syntactic structure has been both *found* and *not found* in LMs due to a lack of consistency about how and why conclusions are reached in this domain. They warn against using model performance measurements on individual coding properties (e.g the *-s* marker for 3SG in English) to make conclusions about syntactic performance and advocate instead for a typologically-driven approach. This could examine, for instance, syntactic performance on agreement mechanisms cross-linguistically to make any type of conclusion. They additionally note the vast number of variables that can affect syntactic performance, including *architecture*, *pre-training task*, *dataset domain* and *size*, *model size*, and the many hyper-parameters that can be adjusted. One particularly important aspect of TSE research design is to avoid aggregate metrics as these can lose relevant detail and cause an overgeneralised analysis. These types of metrics are indeed necessary, but must be thoughtfully selected depending on research question. For instance, the evaluation of a syntactic phenomenon such as subject-verb agreement should use a *macro-average* over verbs of all frequencies as opposed to only a *micro-average* over a smaller class of high-frequency verbs. This can tell us whether the model has actually made an inference about the syntactic structure itself as opposed to rote memorisation akin to a person learning off a list of verb forms for a language that they don't speak.

For the second issue, challenges with LRLs stem mainly from data scarcity, a lack of available models, and the dominance of majority languages. Many languages additionally lack existing research into the types of syntactic constructions that would be of interest to evaluate. Progress is made here through the `MultiBLiMP 1.0` for cross-linguistic evaluation, however this collection of datasets is thus far limited to simple agreement phenomena and this makes it difficult to isolate language-specific phenomena. More research is additionally required into how models handle unusual or complex constructions cross-linguistically. Kryvosheieva and Levy [2024] carried out such experiments for typologically difficult constructions resulting in the evaluation of Swahili noun-class agreement, Hindi split ergativity, and Basque verb agreement. They found that performance correlates well with how well represented a language is in the training data for multilingual models, with Hindi ranking highest likely due to its strong representation. They concluded that multilingual benchmarks can be misleading due to the wide variation in performance for languages and sometimes the complete lack of particular syntactic properties being encoded for LRLs. Improving representation of these languages in TSE research will not only give us a stronger understanding of model performance for underrepresented languages, but also indicate the types of constructions that need to be included in the pre-training or fine-tuning data for improving future model performance.

CHAPTER 3

EXTENSIONS AND CRITIQUES

Controlled experiment design is important due to the large number of confounding variables that can influence our results and conclusions. For instance, the balancing of sentence length, word-order variation, and lexical item frequency are all important to consider. In this chapter we discuss a number of additions for the improvement of experiment design for TSE research. In Section 3.1 we outline the difficulty of evaluating how well LMs generalise syntactic structure due to unfettered training data and partially address this using semantic implausibility. In Sections 3.2 and 3.3 we formalise a number of biases that were experienced in our own trial-and-error process of evaluation, namely those of *next-token recovery* and *subsequences.* Lastly, we criticise a common method of syntactic evaluation in LMs that uses *sentence-level* over *token-level* probabilities in Section 3.4.

3.1 Testing Models on Unseen Data

Traditionally, we have been able to test whether a model has generalised well through a process of strictly controlling the *training*, *validation*, and *test* datasets. These are three distinct "splits" of the full set of data that you are working with. The training split is intended to be used for pre-training the model and allowing it to learn the patterns inherent within the dataset, the validation split for comparing multiple models and adjusting choice of hyper-parameters, and finally the test split for the calculation of the final metrics such as *accuracy*, *sensitivity*, *specificity*, and so on. The test split, by nature of being separated from the pre-training data, is known as an *out-of-distribution* (OOD) dataset. This training-validation-test split of the data has been vital to the evaluation as to whether models have generalised the data well, as a model by definition cannot have generalised appropriately from its *training set* if it does not perform well on new examples from the *test set*; data that was exposed to the model in pre-training is no longer useful for generalisation testing. This tradition of data separation has unfortunately fallen to the wayside as the objective of modern LLM development has shifted towards feeding in the maximum amount of data possible in the pre-training phase, and then fine-tuning on a particular task or post-training to improve performance and ethical alignment. This has led to a scenario where we can no longer fully assess the ability of these models to generalise well from their performance on tasks alone. Any syntactic tests that we carry out run the risk of having been already observed in the pre-training data and therefore the matter of whether particular samples are out-of-distribution or not has lost its verifiability.

Finding Unseen Data

The evaluation of *syntactic* performance however has a distinct advantage in this respect over evaluation of *semantic* performance: we can assume that the vast majority of data fed to these models is either (1) grammatical and semantically meaningful, or (2) ungrammatical but semantically meaningful. Less likely however are sentences that are grammatical but semantically implausible. For instance, take the sentence *"our baby oversees the restaurant"*. Given our knowledge of the world, it is safe to assume that this sentence is more unlikely to have appeared in the training data than the sentence *"our baby likes the restaurant"*. Neither sentence commits any *syntactic* violation, however the former can be thought of as committing a certain *semantic* violation due to the impossibility of a child overseeing anything other than the destruction of your family living room. We may imagine that these types of sentences can be seen as at least closer to "out-of-distribution" in a world of unfettered training data.

Minimal Pairs for Semantically Implausible Sentences

We can therefore *partially* test model generalisation finding minimal pairs (w_1, w_2) for contexts that are semantically implausible but syntactically grammatical when w_1 is chosen and ungrammatical when w_2 is chosen:

> "Nowadays our baby _ this restaurant". → oversees ✓ / oversee ✗
>
> "Our refrigerator _ in the kitchen". → debates ✓ / debate ✗
>
> "The bicycle _ on the street". → flirted ✓ / flirt ✗

Counterintuitively, these *grammatical* but increasingly *nonsensical* minimal pairs are likely to strengthen our own capacity to measure how well a model has acquired a particular syntactic construction. We can therefore define the the goal of minimal-pair evaluation of syntactic generalisation in LLMs as in 3.1. The value of such generated minimal-pair sets is shown in Figure 3.1.

> **Goal:** To partially evaluate the syntactic generalisation of LMs trained on uncontrolled training data, we compare performance on grammatical-ungrammatical minimal pairs (w_1, w_2) given a context C in semantically meaningless or unlikely sentences, where the choice of w_1 results in a sentence committing a set of semantic violations $\mathbb{V}_{sem}$ but no syntactic violations, and the choice of w_2 results in a sentence committing a set of semantic violations $\mathbb{V}_{sem}$ and a single syntactic violation v_{syn}.

We say *partially* here due to a number of factors that reduce the efficacy of this approach. For one, there is no guarantee that these semantically implausible sentences have not been observed in the model training data. However, it may at least be of interest to see what sort of effect they have on model performance. A second issue is that of non-additive semantic violation effects swaying any measure of generalisation, as we will discuss in the next section.

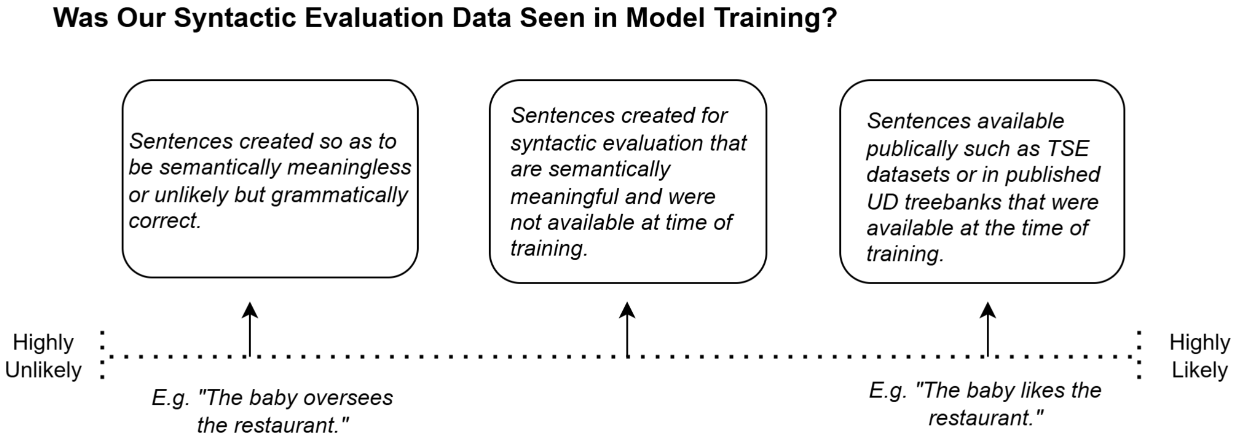

Was Our Syntactic Evaluation Data Seen in Model Training?
Sentences created so as to be semantically meaningless or unlikely but grammatically correct.
Sentences created for syntactic evaluation that are semantically meaningful and were not available at time of training.
Sentences available publically such as TSE datasets or in published UD treebanks that were available at the time of training.
Highly Unlikely
Highly Likely
E.g. "The baby oversees the restaurant."
E.g. "The baby likes the restaurant."

Figure 3.1: Difference in likelihood of different data being seen during model training.

Semantic-Syntactic Interactions

When evaluating syntactic generalisation, it is often assumed that semantic and syntactic factors contribute independently to a model's judgements. However, in practice, these may interact in *non-additive* ways. That is, their contribution to the model's token preference is not simply the sum of the semantic effect and the syntactic effect. We can imagine a function f that determines the effect the set of semantic and syntactic violations has on the model's final probabilities. As shown in Equation 3.1, a non-independent interaction of any semantic issues (represented by a set $\mathbb{V}_{sem}$) and the single grammatical issue in a minimal pair (represented by v_{syn}) may result in an effect that is different from that which we might expect if they were independent.

$$f(v_{syn}, \mathbb{V}_{sem}) \neq f(v_{syn}) + f(\mathbb{V}_{sem}) \tag{3.1}$$

The consequence of this insight is that a model performing worse on out-of-distribution minimal pairs does *not* necessarily mean that the model has not generalised. For us to make such a conclusion, we would need to determine whether syntactic and semantic issues are modelled independently. What we can do however is attempt to show that f is *monotonic*, that is, f only ever increases as more constraints are added. Therefore no matter the interaction of the syntactic and semantic constraints, the additional violation of a syntactic constraint v_{syn} when compared with the same function without the violation of v_{syn} always results in the increase of f. This is shown in Example 3.2.

$$
\begin{aligned}
f(v_{syn}) &> f(\varnothing) \\
f(v_{syn}, v_{sem}) &> f(v_{syn}) \\
f(v_{syn}, \mathbb{V}_{sem}) &> f(\mathbb{V}_{sem})
\end{aligned}
\tag{3.2}
$$

If we observe this monotonic effect then we can say that the model does appear to be at least *tracking* the target syntactic feature for *this* construction within *this* sample. That is, if the model's preference towards the grammatical item rather than the ungrammatical is *less* for the semantically implausible minimal pairs, this does not necessarily mean that the model hasn't generalised syntactic structure. Showing that a *preference* towards the grammatical item simply exists for both sets of data shows that the model at least appears to be tracking a syntactic feature and may have generalised, even if it has done so in ways we don't fully understand. While there is a lot that could be said about what the 'ideal' generalisation for syntax in LMs might be, we will not explore this topic too deeply in this work.

3.2 Next-Token Recovery Bias

This bias is defined by the ambiguities introduced at a grammatical inflection point when there is the possibility that the ungrammatical item may become grammatical depending on the further development of the sentence.

For instance, we may wish to design an experiment evaluating number agreement for the default main-clause word order in German, subject-verb-object (SVO), against for the default embedded-clause word order subject-object-verb (SOV). Singular verbs are strictly inflected with a suffix *-t*, while plural verbs with a suffix *-en*. The verb stem may undergo additional changes depending on if it is *strong* or *weak*. Strong verb stems will typically change for some variant of number and person agreement, such as *essen* "to eat" ⇒ *isst* "he eats" or *singen* "sing" ⇒ *sang* "he sang". A number of examples of subject-verb number agreement and the respective suffixes are shown in Example 3.

(3) Ich **ess-e** den Apfel
I.NOM eat.PRES-1SG the.ACC apple
'I'm eating the apple.'

Er **iss-t** den Apfel
he.NOM eat.PRES-3SG the.ACC apple

'He's eating the apple.'

Sie **ess-en** den Apfel
they.NOM eat.PRES-3PL the.ACC apple

'They're eating the apple.'

An SOV word order typically appears in an embedded clause such as a relative clause. In Example 4 we observe a number of SVO examples, converted from Example 3 to an embedded clause using the complementiser *dass* "that".

(4) Ich denke, dass ich den Apfel **ess-e**
I think.1SG that I.NOM the.ACC apple eat.PRES-1SG
'I think that I'm eating the apple.'

Ich denke, dass er den Apfel **iss-t**
I think.1SG that he.NOM the.ACC apple eat.PRES-3SG

'I think that he's eating the apple.'

Ich denke, dass sie den Apfel **ess-en**
I think.1SG that they.NOM the.ACC apple eat.PRES-3PL

'I think that they're eating the apple.'

While this may appear to be a reasonable experiment setup, a token-level evaluation of models trained on NTP tasks would likely yield inherently biased results. The issue here inherently comes down to (1) the model having no access to any context subsequent to the target token, and (2) inherent ambiguities within German syntax. Given the limited context window of models performing next-token prediction, ambiguity becomes much more prevalent of an issue. We can see examples of this in Example 5, firstly for the minimal pair "sie liest" *she reads* and "lesen" *they read*, and secondly for the pair "er...liest" *he reads* and "lesen" *they read*. The context that would be available to a decoder model is indicated in square brackets.

(5) *Issue 1: Determination of syntactic feature is made after the target.*

[Sie] _____ alle zusammen das Buch. → *lesen* ✓ / *liest* ✗
she/they/you(formal) ? all together the.ACC book

'They all *read/reads* the book together.'

Issue 2: Sentence continuation can lead ungrammatical token to become grammatical.

[Ich glaube, dass er das Buch] _____. → *liest* ✓ / *lesen muss* ✓
I believe.1SG that he.NOM the.ACC book ?

'I believe that he *reads/must read* the book.'

For the first issue, the pronoun used is the same in form but can represent either the 3rd-person singular or 3rd-person plural. The correct feature is only determined subsequently with the word *zusammen* "together". In the second example we observe a bias due to *verb chaining*. That is, the model is skewed by the fact that the sentence could be equally *sie..lesen muss* 'she..has to read' as *sie liest* 'she reads'. This renders the minimal pair *liest/lesen* largely unusable for this type of model due to the polysemous nature of 'lesen'. We may formalise a well-formedness condition for the minimal pairs as in Definition 2.

Definition 2. ***Next-Token Recovery Bias****: For a model M evaluated on a grammatical/ungrammatical minimal pair* (w_1, w_2) *given a prior context* T_{prior}*, there must not be any possible set of subsequent tokens* T_{post} *that allows the grammaticality of* w_2 *in* $T_{prior} \cdot w_2 \cdot T_{post}$*.*

One implication of this is that some level of grammatical knowledge of a language is desirable when carrying out syntactic tests and it is advisable to check the syntactic tests that are created for any unexpected such ambiguities. This key point is not just the case for decoder models where only T_{prior} is available, but also for encoder models where T contains the entire previous and subsequent context window. For instance, the above rule would be violated when testing a model trained for MLM on the German pronoun *sie* "she" / "they" as can be observed in Example 6

(6) [Sie _____ das Buch.] → *liest* ✓
she.NOM read.PRES-3SG the.ACC book
'She reads the book.'

[Sie _____ das Buch.] → *lesen* ✓
they.NOM read.PRES-3PL the.ACC book

'They read the book.'

3.3 Subsequence Bias

There is another issue that we have found to be present within TSE research pertaining to token-length bias, that is, the bias introduced by more or less tokens being provided as input. This issue is persistent throughout different areas of model performance [Yang et al., 2024b, Levy et al., 2024, Phan et al., 2024]. Ueda et al. [2024] investigated how token-length bias effects minimal-pair evaluations across tests and the consequence of this for the final results. The issue we discuss here however pertains to a specific issue within individual syntactic tests, one which essentially renders them invalidated. Consider a grammatical/ungrammatical minimal pair (jog,jogs) in the masked sentence "the guy [MASK] every day after work", where after tokenisation they are ['jog'] is ['jog','##s'], respectively. The probability scores for each item in the minimal pair are calculated through the use of joint probability. The resulting probability in the first case would therefore be and in the second case would be:

$$\begin{aligned} Grammatical &= P(jog|the, guy, [MASK], after, work) \\ Ungrammatical &= P(run|the, guy, [MASK], after, work) \\ &\quad \cdot P(\#\#s|the, guy, jog, [MASK], after, work) \end{aligned}$$

The joint probability of the shorter word is therefore going to always be less than or equal to the larger word. This applies for any case where the set of tokens for one word is a subsequence of the other. We can define this principle as in Definition 3.

Definition 3. ***Subsequence Bias***: *Given a minimal pair* (w_1, w_2), *let their tokenisations be represented as sets of tokens* $T_1 = \{\tau_1, \tau_2, \ldots, \tau_m\}$ *and* $T_2 = \{\tau'_1, \tau'_2, \ldots, \tau'_n\}$. *Neither full set may be a subsequence of the other.*

3.4 What's a roof worth? A Critique of Sentence Probabilities

One common alternative to calculating *token-level* probabilities e.g. $P(eats|For, lunch, he)$ is to instead calculate *sentence-level* probabilities e.g. $P(For, lunch, he, eats, a, salad)$. This would then be compared to the ungrammatical alternative $P(For, lunch, he, eat, a, salad)$. This approach helps to ease the issues of next-token recovery and subsequence violations, often being implemented where the focus is on testing many syntactic phenomena or languages at once. For instance, it was used for the results presented in the cross-linguistic `MultiBLiMP` dataset [Jumelet et al., 2025] as well as research in the syntactic performance across the Chomsky hierarchy [Someya et al., 2024]. However, with these benefits we also adopt a significant issue that is not fully addressed in the literature: a sentence-level probability is *not* the calculation of the probability of a sentence as a whole, but rather the autoregressive product of probabilities for each token in the sentence. The issue is that the model now takes as *given* the fact that an ungrammatical token has been chosen for all probabilities subsequent to the target, causing a potential downstream bias against the ungrammatical item due to the effect this may have on the probability of words irrelevant for grammaticality. We can imagine, for instance, the likelihood of the sentence *"he takes✓/take✗ shelter under the roof"* for a simple word-level tokeniser:

$$
\begin{aligned}
P(he)\cdot \\
P(takes|he)\cdot \\
P(shelter|he, takes)\cdot \\
P(under|he, takes, shelter)\cdot \\
P(the|he, takes, shelter, under)\cdot \\
P(roof|he, takes, shelter, under, the)\cdot
\end{aligned}
$$

This is fine for the word 'takes' as it is grammatical in this context, however what effects might we see for the ungrammatical word 'take'? The sentence-level joint probability calculations would be carried out as follows:

$$
\begin{aligned}
P(he)\cdot \\
P(take|he)\cdot \\
\mathbf{P(shelter|he,take)}\cdot \\
\mathbf{P(under|he,take,shelter)}\cdot \\
\mathbf{P(the|he,take,shelter,under)}\cdot \\
\mathbf{P(roof|he,take,shelter,under,the)}\cdot
\end{aligned}
$$

With this approach we would therefore be taking into account the probability of tokens unaffected by the grammatical inflection point *given* that we have chosen the ungrammatical token. The ungrammatical item in the probability posterior for all tokens subsequent to the target introduces a potentially unexplored confounding variable and may conflate results in one direction or another, likely causing a stronger bias against the ungrammatical item in the results than the model has necessarily encoded. For each additional token after the target more variance unrelated to grammaticality is introduced and their contributions to the overall result become entangled with the model's ability to recover from an ungrammatical token. In the example given above, we can see for instance that *roof* will have an undue effect on the final sentence probability due to being the final word in the sentence. The difference between the probabilities shown in 3.3 and 3.4 therefore has a non-understood effect on our results despite *roof* being irrelevant for the determination of well-formedness.

$$P(roof|he, takes, shelter, under, the) \tag{3.3}$$

$$P(roof|he, take, shelter, under, the) \tag{3.4}$$

We therefore ask a key question in relation to this issue:

> *"What are the downstream effects of an ungrammatical word in the set of posterior tokens fed to a language model?"*

We could for instance imagine a naive model that sets all probabilities to 0 after encountering an ungrammatical token, or one that continues on as if the ungrammatical token were grammatical. We believe that the effect likely lies in a murky middle ground. This may be a fruitful avenue for future work as TSE research has often adopted this approach without addressing this question. We therefore do not rely on sentence-level probabilities for the experiments carried out in Chapter 5.

CHAPTER 4

METHODOLOGY

This section concerns the design and implementation of the central pipeline for this work, termed `GrewTSE`, or *Grew for Targeted Syntactic Evaluation.* Section 4.1 will give an overview of how it was built both from a design as well as function perspective. Section 4.2 will detail the individual stages of the pipeline and the final metrics used in model evaluation will be outlined and justified in Section 4.3. Finally, limitations of our methodology are discussed in Section 4.4. The pipeline is available as an open-source Python package located at `github.com/DanielGall500/Grew-TSE`.

4.1 Overview

To what degree has a given model M learned a syntactic construction C in language L? As seen in Section 2, there has been both significant progress and setbacks in answering this key question since the rapid improvement of transformer-based LMs. While the standard means of carrying out such evaluations has been established as minimal-pair syntactic tests, the availability of such tests cross-linguistically is limited due to the significant amount of man hours required in their creation. The pipeline presented here aims to alleviate this issue by taking a new approach; it allows the query-based generation of either masked or prompt-based minimal-pair datasets availing of the abundant time and effort that has already gone into the development of UD treebanks. The *masked* syntactic tests (e.g "the keys to the cabinet `[MASK]` on the table" → *are* ✓, *is* ✗) may be used for models trained on the task of masked language modelling (henceforth MLM), while the *prompt* tests (e.g "the keys to the cabinet " → *are* ✓, *is* ✗) may be used for those trained for next-token prediction (henceforth NTP). We design, build, and test a pipeline that carries out (1) the query-based isolation of syntactic constructions from treebanks, (2) the generation of either masked or prompt-based minimal-pair syntactic tests, and (3) the evaluation of transformer-based language models on these tests using the metrics of *accuracy*, *average surprisal difference* (ASD), and *certainty*, as described in Section 2. We have termed it `GrewTSE` to emphasise its reliance on the `Grew` query language for isolating syntactic phenomena. The use of treebanks in this pipeline is particularly beneficial for LRLs as there is already a significant number available. The pipeline is aimed at machine learning researchers interested in LM syntactic evaluation, linguists interested in carrying out experiments on the learnability of a syntactic construction, as well as anywhere in between on the inter-disciplinary spectrum of syntactic analysis. A full overview of the pipeline was provided in Section 1 and is shown again in Figure 4.1 for the Polish sentence *"Nie zjadłem jabłka"* lit. *"I didn't eat the apple"*.

Tools & Frameworks

We have developed this pipeline as an open-source package using `Python`, `GrewPy` [Guillaume, 2021a], and the Hugging Face `transformers` library [Wolf et al., 2020]. Continual framework testing was carried out using `pytest`, as well as `ruff` and `mypy` for linting, formatting, and type checking. The package is made available alongside this work through the `PyPi` package collection.

4.2 Pipeline Stages

We will first give an overview of both the preparatory as well as main steps involved. We must choose a particular language or set of languages, a syntactic construction or set of constructions, as well as a language model or set of models. There are a number of constraints on these. A UD treebank must be available for the languages that we chose, while the models must be trained for performing either MLM or next-token prediction and open-source on the Hugging Face platform. We then create a `Grew` query that allows us to isolate these constructions within a treebank and parse them. We may then run the `GrewTSE` pipeline for

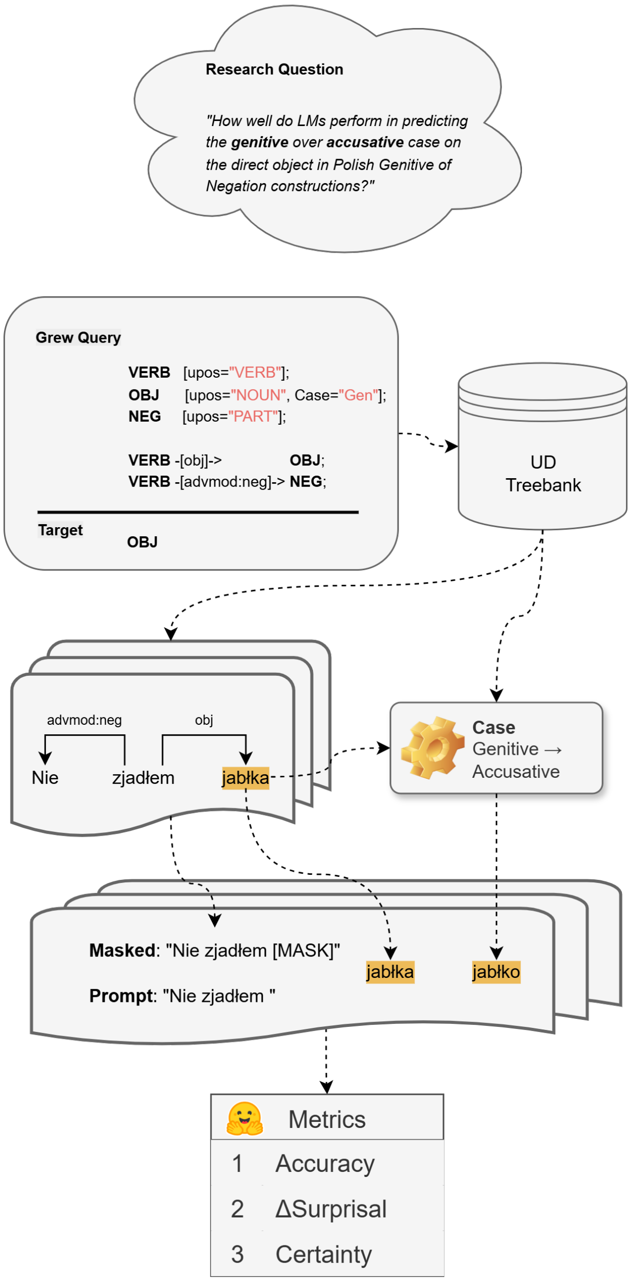


Figure 4.1: The full GrewTSE pipeline shown with an example of analysing Polish genitive of negation constructions.

each language and construction, which carries out the following:

1. **Build A Lexicon:** Build a lexicon of words and their features given a set of `.conllu` files representing a UD treebank.

2. **Isolate Syntactic Phenomenon:** Isolate sentences in the treebank with a given syntactic phenomenon defined by a given `Grew` query as well as the target word w_a.

3. **Mask or Prompt Generation:** For each sentence with the target syntactic phenomenon, perform 'token surgery' to insert a mask token or slice the string to create a prompt.

4. **Minimal-Pair Creation:** Search the lexicon for a word w_b with one differing syntactic feature from the original target word w_a to create a minimal pair. Additional semantically implausible minimal pairs can also be generated provided they were used in the same syntactic structure and have the same features as the original minimal pair.

5. **Hugging Face Evaluation:** Evaluate generated minimal-pair syntactic tests on language models available on the Hugging Face platform.

4.2.1 Building a Lexicon

We first build a *lexicon* of words and their features from the provided UD treebank. The pipeline allows one to provide multiple `.conllu` files for the creation of this lexicon. A `.conllu` file is a standardised text-based format for sharing treebanks [More et al., 2018]. Naturally, the larger the treebank is, then the more information such a lexicon contains. The size of the resulting minimal-pair dataset is primarily constrained by the number of lexical items available through parsing the treebank. For this reason, we allow the passing of multiple treebank files to build up a larger lexicon. This is useful due to how treebanks additionally tend to be typically broken up into smaller datasets. A sample of a lexicon for Polish can be observed in Table 4.1. In cases where a word has no value assigned for a given feature, that word will be assigned `null`. Using this collection of words and features, we can see how searching for a word with a given lemma and one syntactic feature adjusted would be made relatively easy. The datasets are indexed by (`sent-id`, `tok-id`) pairs.

sent-id	**tok-id**	**form**	**lemma**	**upos**	**case**	**gender**	**number**
s1774	4	sytuacji	sytuacja	NOUN	Gen	Fem	Sing
s4464	9	sytuacjach	sytuacja	NOUN	Loc	Fem	Plur
s11811	13	sytuacji	sytuacja	NOUN	Loc	Fem	Sing
s14234	52	sytuacji	sytuacja	NOUN	Gen	Fem	Sing
s14721	6	sytuację	sytuacja	NOUN	Acc	Fem	Sing

Table 4.1: Sample of a Polish lexicon. with each word in a treebank assigned featural information.

4.2.2 Isolating Syntactic Constructions

In Section 2, we introduced the `Grew` query language used to search for syntactic phenomena in UD treebanks. This pipeline is unique in that it makes use of `Grew`'s powerful pattern-matching capabilities and user-friendly syntax in order to allow the generation of syntactic tests. An example query that can be used for "that" complementiser clauses in English is shown in Example 4.2. The first two lines represent individual nodes with a universal position of `part` and `verb`, respectively. The third line finds any such nodes with a clause-marking particle relationship. We provide two parameters in this respect:

```
pattern {
    comp [upos="PART"];
    myTargetVerb [upos="VERB"];
    myTargetVerb -[mark:prt]-> comp;
}
```

Figure 4.2: An example Grew query for finding complementiser clauses in English.

- **Query:** the formal specification of the syntactic pattern to be matched within the dependency treebank, written within `pattern {}` braces. Each query defines a set of structural constraints (e.g. node labels, dependency relations) that identify instances of a particular syntactic construction within the database.

- **Target:** the specific variable in the query that is being targeted. For instance, in the previous query we may specify `myTargetVerb` as the target in order to target any verbs in the isolated sentences. We can view an example of this in Figure 4.3.

It is vital that whichever item we are targeting in a sentence is assigned a variable name that is then provided as the `target` argument to the pipeline.

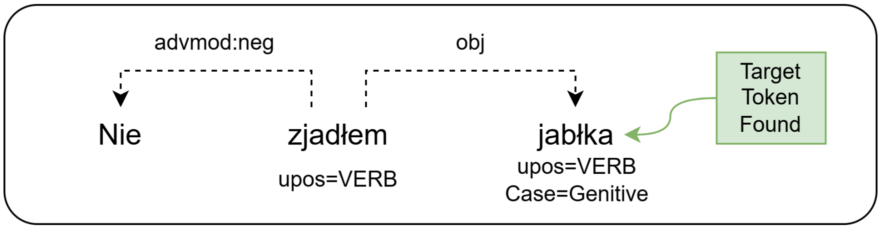


Figure 4.3: Match words based on their dependency structure and features.

4.2.3 Masked or Prompt Dataset Generation

This raw treebank data is subsequently converted into either masked sentences or prompts depending on the selected `target`. A default mask e.g. `[MASK]` is used for this, however some models use a different token. The pipeline automatically swaps out the token if necessary for the one defined by the model. In the case of masked sentences for evaluating models trained on the task of MLM, sentences then undergo a process we term *token surgery*. This is the process of taking a string, e.g "The wolf just hunted the rabbit", and replacing the target e.g *hunted* with the mask token e.g `[MASK]`. It is important here to make sure the correct word is replaced. An algorithm was created to recursively slice the sentence until the target token is found and swapping it for the mask. A visual of this process is provided in Figure 4.4.

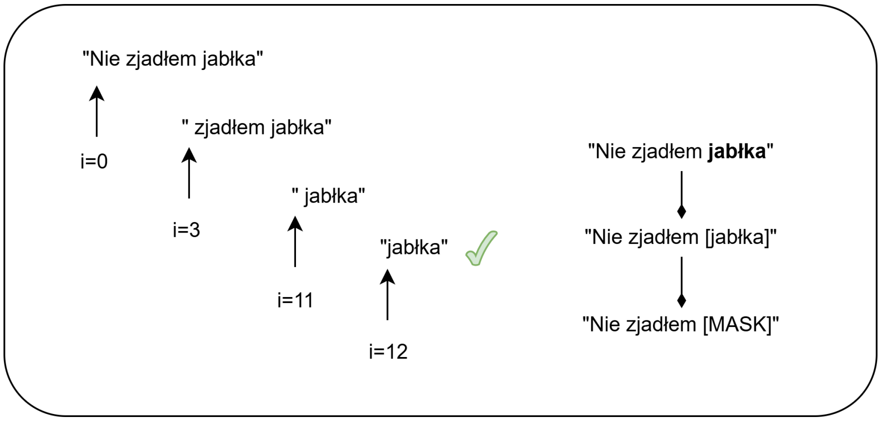


Figure 4.4: Perform recursive “token surgery” to create masked sentences.

4.2.4 Generation of Minimal Pairs

We now have a dataset of masked or prompt-based sentences along with a corresponding target word for each (e.g. “the keys to the cabinet `[MASK]`” → *are*). For each *grammatical* token however, there must be an accompanying *ungrammatical* token. The goal of this step is defined as follows:

> **Goal:** Given a word w_a with lemma ℓ and features δ_1,δ_2...δ_n, in order to create a minimal pair we must find a word w_b also with lemma ℓ and the same syntactic features except for one pre-specified key feature.

A visualisation of this for the previous Polish example is shown in Figure 4.5. To use an English example, the word *hunted* has lemma *hunt* and UD features `verb.3sg.pst` for instance. In order to create a syntactic test for the sentence “The wolf just _ the rabbit”, then we may wish to search for the same word but with feature `tense` adjusted to PRES. If successful, this would yield *hunts*, as visualised in Table 4.2. The lexicon, which contains information about every word in the treebank as well as their features, allows us to search for the word w_b with one feature changed from w_a. This results in a minimal pair (w_a, w_b). Our syntactic tests are now fully formed.

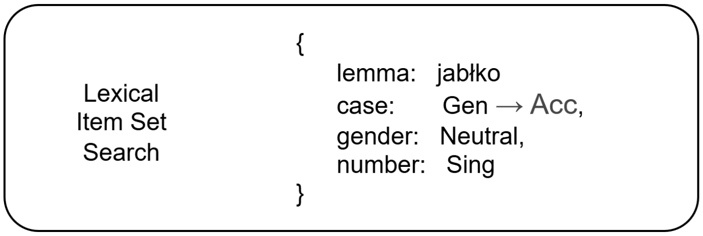


Figure 4.5: We search the lexicon for the same token but with a single differing syntactic feature. For instance, the conversion of GENITIVE → ACCUSATIVE.

form	**lemma**	**upos**	**tense**	**person**	**number**
hunts ✓w_B *found*	hunt	VERB	Pres	3	Sing

Table 4.2: Sample of lexicon entry if adjusting the `person` feature of the verb *hunt*.

4.2.5 Generation of Semantically Implausible Pairs

Due to the issues outlined in Section 3 with attempting to test a model's capacity for syntactic generalisation for any given construction in a world of uncontrolled training data, there is an additional optional stage which finds other minimal pairs (w_{a_1}, w_{b_1}), (w_{a_2}, w_{b_2}),...(w_{a_n}, w_{b_n}) that have the same features as the original minimal pair but with differing lemmas. These pairs increase the chances of testing data for which the model has not been trained on, particularly in cases where a sentence is fully grammatical yet semantically unlikely (e.g. *the book slept on the bookshelf*, *the baby oversees the restaurant*). These sentences are far less likely to have appeared in a model's training data yet are fully grammatical, and hence are strong candidates for testing syntactic generalisation. However, as discussed in Section 3, poorer model performance on these types of tests may *not* indicate a lack of generalisation. There are therefore limitations to this approach yet it offers *some* insight into a model's capabilities. Note that the process of generating semantically implausible minimal pairs significantly increases pipeline runtime. It is additionally important that pairs are chosen which may occur in the syntactic structure being tested. Therefore, the pipeline chooses these pairs based on those it finds in other sentences that have the same syntactic phenomenon.

4.3 Evaluation

An additional component provided in the pipeline is an *evaluation* module. This provides an interface for testing models available on the Hugging Face platform allowing for full evaluations to be carried out by providing a suitable model repository link e.g. *google-bert/bert-base-multilingual-cased* for Google's multilingual `BERT` model. We provide an overview of which models are able to be evaluated with this module in Section 4.3.1 and the resulting metrics in Section 4.3.2.

4.3.1 Integrated Model Tasks

We discussed previously important differences in the tasks that various transformer-based architectures are trained on. The two most relevant for this pipeline are MLM, associated with encoder-only models, and NTP, associated with decoder-only models. Both of these are available for evaluation as part of the `GrewTSE` Python package.

The most common masking approach for encoder-only models is *token-level*, where the model's task is to predict the most likely *token* which could be either a full word or a subword. In order to ensure that a full word is generated, one must insert additional masks after each prediction and re-run the model until a full word is formed. One alternative approach now sometimes implemented is *word-level* masking. Here the model is tasked with predicting an entire word based on the tokens it has available as opposed to one token at a time. The pipeline implements an evaluation procedure that takes both tasks into account and adjusts the evaluation by running the model multiple times with new masks in the former case or running the model once in the latter case.

4.3.2 Metrics

We implement the metrics *accuracy*, *average surprisal difference* (ASD), and the entropic *certainty* score. These were discussed in detail in Section 2 and we will therefore not repeat them here, however we will once again emphasise that it is typical in TSE research for accuracy scores to be determined slightly differently than in other domains. This is the proportion of tests for which a model 'chose' the grammatical item. In this case, this means that the probability assigned by the model to the grammatical item was *higher* than the probability assigned to the ungrammatical item. This distinguishes itself from the classical accuracy metric that simply takes the token assigned the highest probability and compared that to what is deemed grammatical or ungrammatical.

An important consideration taken into account for the `GrewTSE` pipeline pertains to calculations of model certainty. The level of uncertainty one has about the outcome of rolling a fair die is always higher than flipping a coin - a property determined by the number of possible outcomes. Possible *outcomes* in an LM can be interpreted as possible *tokens* in the vocabulary. One issue with our above metrics thus lies in comparing the certainty of multiple models with differing vocabulary sizes; differing values of n will lead models of a smaller vocabulary size to be naturally favoured regardless of model *certainty* or *uncertainty*. For instance, if two models assign the exact same probabilities to the same three tokens, but one model just happens to have a greater vocabulary size with all additional tokens assigned a probability of 0, then the certainty will be skewed in favour of that model which has the smaller vocabulary despite both models being just as certain about the given tokens. We have therefore taken a *top-k* approach which takes the top 100 highest probabilities for each prediction and uses these to compute the model's certainty. This means that for all models compared $n = 100$, which allows improved comparability across models of different vocabulary sizes while retaining the most important information about model certainty.

4.4 Limitations

There are many factors that determine the success of our evaluation pipeline. Firstly, the level of detail in a UD treebank will determine the *granularity* of any research question that may be explored. For instance, the inclusion of values for the `mood` and `aspect` of each verb allows the generation of tests for these features. We found during the trial-and-error process of experimentation that some treebanks tended to contain far more typological detail than others and this set an upper bound on the number of datasets that could be generated. Relatedly, *exploiting* that level of detail is an additional limitation. The most interesting phenomena for testing can be best identified by those who are most familiar with a language and we thus encourage those speakers of LRLs to determine those syntactic phenomena which may be of most evaluation interest.

We caution against drawing conclusions about the *nature* of individual languages themselves through such experiments. As we outlined in Section 1.1, it may indeed be the case that LMs can be used to study and contribute to theories of natural language syntax. However, individual experiments are limited in what they can determine for themselves with respect to this and future work may determine steps towards these types of conclusions.

In Section 3 we determined a number of important *biases* that may be present in TSE experiment results and future work may look towards finding and identifying more. These must be taken into account during experiment design. In order to create fully verified insights into LM syntactic performance it is necessary to consult with a native speaker or linguist familiar with the language.

CHAPTER 5

EXPERIMENTS

In this chapter we will introduce three experiments carried out in order to test the GrewTSE pipeline. The languages in question are Polish, German, and Georgian. The first experiment will examine encoder-only models for the genitive case in Polish, the second decoder-only models for conditional auxiliary constructions in German, and the third encoder-only models for the split-ergative case system in Georgian. Note that we will henceforth refer to the generated semantically implausible minimal-pair datasets as 'out-of-distribution' or 'OOD' to be concise, though we recognise that they may be more accurately deemed *potentially* out-of-distribution. A general discussion of the key findings is given in Section 5.4. We find that experiment design is significantly easier for encoder-only models trained on MLM tasks over decoder-only models on NTP tasks due to the former's inclusion of context both prior to the grammatical/ungrammatical target as well as after it. We used these learnings from the first and second experiment to inspire the third experiment for the low-resource language Georgian, where we focused exclusively on BERT models and controlled for any biases. Given that we do not speak Georgian, this final experiment was a test not just of the models or language but of the soundness of the methodology itself. A native speaker may determine the validity of this last dataset as an interesting avenue for future work.

5.1 The Genitive Case in Polish

This experiment will focus on the evaluation of various BERT-based models (cased) with regards to the Polish genitive case, as this case plays a particularly important role across multiple dimensions of Polish syntax. The BERT model architecture is pre-trained for the task of MLM, and therefore takes into account the full sentence bar a masked token when making a prediction. This is advantageous in evaluating a language model for a language with relatively free word order, as syntactically vital lexical items may appear after the token in question. Section 5.1.1 will introduce the three syntactic constructions that were tested for these experiments, while Sections 5.1.2 and 5.1.3 will introduce the generated minimal-pair datasets as well as the models under evaluation. Lastly, Section 5.1.4 will present the evaluation results and a discussion of the insights we can garner from them. Glossing will follow the *Leipzig Glossing Rules* [Max Planck Institute for Evolutionary Anthropology, 2015] unless otherwise specified.

Syntactic Evaluation of Polish

Polish is a West Slavic language that is syntactically characterised by its three grammatical genders, complex system of inflections corresponding to gender, person, animacy and case, as well as its pro-drop and reflexive pronoun system. As is typical for a highly-inflected language, its word order is relatively free. It does however default to a subject-verb-object order. Nouns which are masculine and animate (e.g *ojciec* ‘father’) carry with them a specific set of inflections that are distinct from a masculine inanimate noun (e.g *stół* ‘table’), while animacy causes no such change in feminine or neuter nouns.

Polish is classified as a low-resource language and lags behind other languages when it comes to both semantic and syntactic evaluation of language models. Evaluation of BERT embedding models for Polish has indeed been performed for semantic tasks [Dadas et al., 2020a, Mroczkowski et al., 2019] as well as probing of linguistic information retained in vector embeddings [Krasnowska-Kieraś and Wróblewska, 2019], however there are limited studies that pertain to targeted syntactic evaluation. The KLEJ benchmark [Rybak et al., 2020] was created to test the performance of Polish models on a variety of tasks such as question answering or named-entity recognition and will form the basis for our understanding of the ‘state-of-the-art’ in this area. However, there does not appear to exist a *Benchmark of Linguistic Minimal Pairs* that has been used to test the ability of these models to absorb

Polish language *structure*, as there is for Russian [Taktasheva et al., 2024b]. This experiment aims to help rectify this by performing an evaluation of a number of state of the art BERT models for three syntactic constructions pertaining to the *genitive* case. A dataset of 900 masked sentences each with a syntactic minimal pair will be created using our pipeline, representing a small step towards a full benchmark for Polish syntax.

5.1.1 Constructions

The genitive case is a common grammatical case among the world's languages and is typically associated with ownership i.e a possessor-possession relationship. There are many ways that this relationship can be marked. For instance, Kazakh has the suffix *-nıŋ* which attaches to the possessor and varies through consonant assimilation and vowel harmony (e.g *Bala-nıŋ oy-ı* 'the child's house'), while Spanish has the fixed adposition *de* that sits after the possessor and before the possession (e.g *El rey de la Habana* 'the king of Havana'). Polish marks the genitive case through a system of suffixes which can depend on number, person, gender, and animacy, attaching to both nouns and any adjectives modifying the nouns. For instance, the genitive suffix *-ów* in *kruków* 'of the ravens' depends on the the masculine gender, plural number, as well as animacy of the noun. This complex system of suffixes is what will be put under examination in this experiment. The three constructions that necessarily require the genitive case and how they will be tested are as follows:

1. **Genitive of Negation:** Case selection in negated transitive verb constructions.

2. **Verbal Genitive**: Case selection in transitive verbs which license the genitive case as opposed to the typical accusative for direct objects.

3. **Prepositional Genitive**: Case selection for nouns which depend on genitive-licensing prepositions.

Genitive of Negation

A common construction across Slavic languages is known as the *genitive of negation.* This construction causes verbs which normally take an accusative direct object (e.g *jeść* 'to eat', *robić* 'to do/make') to take a genitive direct object when negated. The pattern is as follows:

- If a transitive verb can take an accusative direct object, then it will do so if its verb phrase is not negated.

- If a transitive verb can take an accusative direct object but is negated, it will take a genitive direct object instead.

This can be observed for the neuter noun *jabłko* 'apple' in Example 7. In both cases it is a typical transitive verb phrase with a subject and direct object, however in the latter case there is negation. This causes the resulting δ_{case} feature to undergo the change ACC → GEN on the direct object.

(7) Zjad-ł-em jabłk-**o**
eat-PST-1SG.M apple-**N.SG.ACC**
'I ate the apple.'

Nie zjad-ł-em jabłk-**a**
NEG eat-PST-M.1SG apple-**N.SG.GEN**

'I didn't eat the apple.'

In English for instance, this would have the effect of producing *"I like eating fish"* and **"I don't like eating of the fish"*.

Verbal Genitive

The Polish accusative case when applied to direct objects requires either no explicit marking as with masculine inanimate or neuter nouns, or a suffix inflection as with feminine or masculine animate nouns. Note that the absence of an ANIM gloss implies that a noun is inanimate. For instance:

(8) Widz-ę kot-**a**
see-1SG.PRES cat-**M.ANIM.SG.ACC**
'I see a cat.'

Kupi-liśmy chleb-**ø**
buy-1PL.PST.PFV bread-**M.SG.ACC**

'We bought bread.'

Some non-negated transitive verbs in Polish license the genitive case on their direct object rather than the accusative. The number of verbs that have this feature is relatively small, however it applies to a number that tend to occur frequently (e.g *uczyć się* 'to learn', *szukać* 'to look for'). Take the following examples:

(9) Ucz-ę się język-**a** polski-ego
learn-1SG REFL language-**M.SG.GEN** Polish-GEN.SG
'I am learning Polish.'

Szuka-łem klucz-**a**
search-M.1SG.PST key-**M.SG.GEN**

'I was looking for the key.'

We can observe, for instance, 10,324 occurrences of accusative direct objects of non-negated transitive verbs in the Polish PDB-UD treebank [Wróblewska, 2018] while only 991 occurrences for the genitive direct objects in the same construction.

Prepositional Genitive

Prepositions in Polish will typically license the *locative*, *genitive*, *accusative*, or *instrumental* case on their dependant nouns. This primarily follows a fixed system where each preposition will license a specific case with a small minority depending on whether there is motion involved in the action. Of all prepositions in the Polish PDB-UD treebank, the proportional breakdown of casing for the dependant noun is as follows: locative: 63.08%, accusative: 15.31%, genitive: 11.13%, instrumental: 10.48%, nominative, dative, vocative: 0%. By a wide margin, the *locative* is thus the most common case to be licensed by a preposition in Polish. We can see examples of prepositions a preposition that licenses the *locative* and *genitive* case in Example 10.

(10) Nie mog-ę żyć bez jedzeni-**a**.
NEG can-1SG.PRES live without food-**N.SG.GEN**
'I cannot live without food.'

Myś-limy o jedzeni-**u**
think-1PL.PRES about food-**N.SG.LOC**

'We're thinking about food.'

The models will therefore be tested on how confident it is in choosing the *genitive* over the *locative* for sentences which use a *genitive*-licensing preposition.

5.1.2 Data

The grammatical form in the GenOfNeg construction is always the direct object with a `case=Genitive` feature, while the ungrammatical form is that same noun with feature `case=Accusative`. In order to isolate these constructions, we will use the Polish PDB UD treebank [Wróblewska, 2018] which contains a total of 22,152 sentences. For these experiments only the *train* subset of 17,722 sentences was used due to this set reaching a suitable amount of sentences for the purposes of this experiment. An overview of the construction datasets can be seen in Table 5.1. The full `Grew` queries provided as input to the pipeline are provided in Appendix A. Only nouns which vary by at least one character are considered, as otherwise the probability readings from the models would not be meaningful. A number of samples from the resulting dataset can be viewed in Example 11. Note that for these glosses we will not subdivide words into morphemes in order to represent more closely the raw syntactic test that is given to a model. In the end, a total of 900 masked, minimal-pair samples were collected for testing the Polish genitive case.

(11) Genitive of Negation

Kampania	nie	przyniosła	[skutku✓/skutek✗].
campaign	NEG	brought	[effect.GEN/.ACC]

'The campaign was not successful.'

Sentence ID train-s1739

Verbal Genitive

Szuka-jąc	[zysku✓/zysk✗],	musiały	ryzykować
seek	[result.SG.GEN/.SG.ACC]	must	risk

'In seeking profit, they had to take risks.'

Sentence ID train-s2075

Prepositional Genitive

Od	[strony✓/stronie✗]	wjazdu	zbliżał	się	ubłocony	fiat	Kosińskiego.
from	[side-GEN.SG/-ACC.SG]	entrance	approach	REFL	muddy	Fiat	Kosiński

'From the direction of the entrance approached Kosiński's mud-covered Fiat.'

Sentence ID train-s9228

5.1.3 Models

In setting out to find suitable models to evaluate, the three requirements aside from being a `BERT` model (cased) were (1) the model is pre-trained for the task of MLM, (2) the model is

GrewTSE Dataset	ID	Description	Size
Genitive of Negation	GenOfNeg	Tests the model in choosing between the *genitive* and *accusative* form for direct objects of a negated verb phrase.	378
Verbal Genitive	VerbGen	Tests the model in choosing between the *genitive* and *accusative* forms for a verb which requires a genitive direct object.	413
Prepositional Genitive	PrepGen	Tests the model in choosing between the *genitive* and *locative* form for dependant nouns of a genitive-licensing preposition.	109

Table 5.1: Overview of the three masked, minimal-pair datasets for Polish BERT models generated by the GrewTSE pipeline.

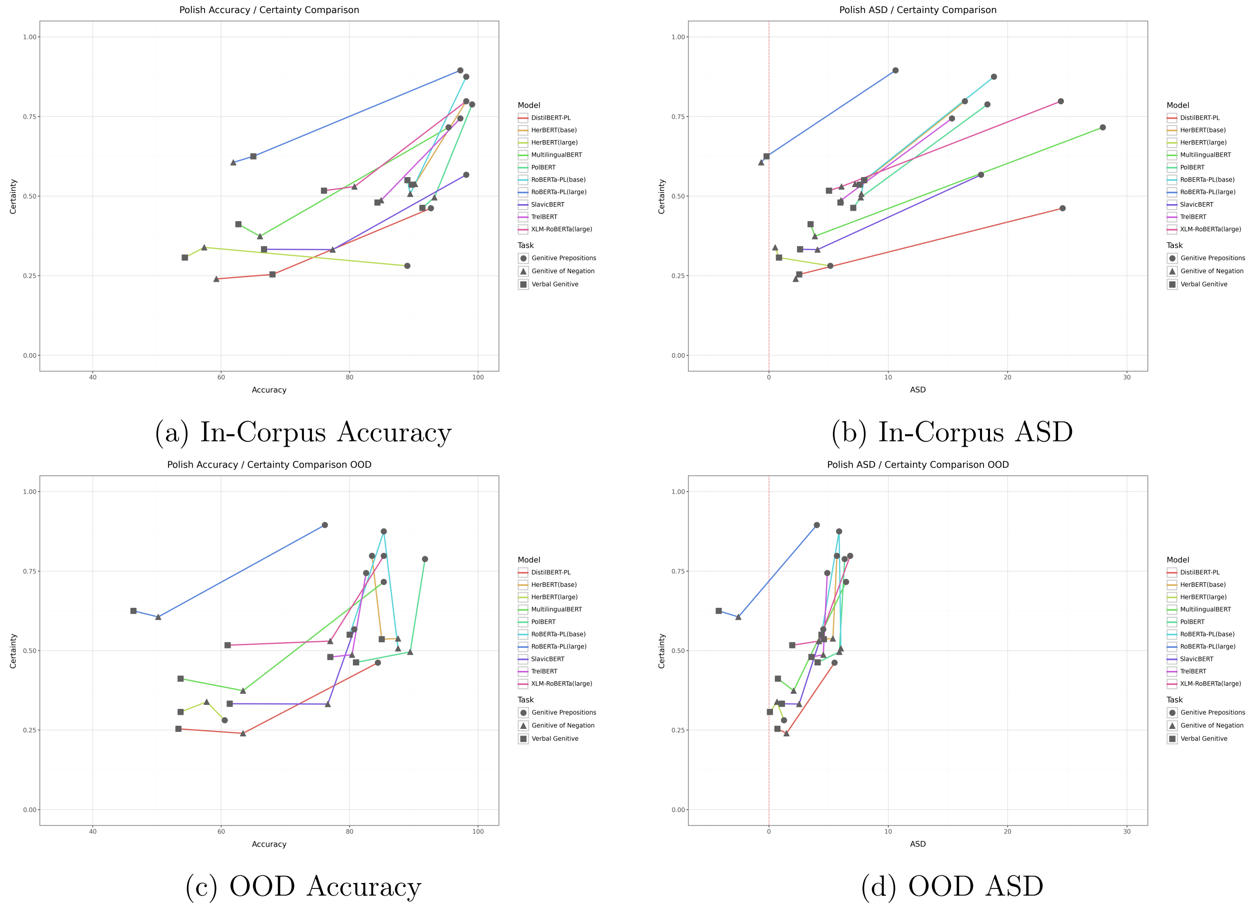
intended to be used fully or partially for the Polish language, and (3) the model is available to the public on the HuggingFace platform. The KLEJ benchmark [Rybak et al., 2020] measures model performance on a group of semantic tasks in Polish such as named-entity recognition (NER) on the NKJP corpus [Przepiórkowski et al., 2010], relatedness scoring between images and sentences, as well as question answering validation. It provides a ranked list of polish BERT models based on performance and therefore 9 of the 10 models were taken from here. A further model - Polish DistilBERT - was also included in order to examine how well a distilled model performed. For this experiment we test the multilingual models Multilingual BERT [Devlin et al., 2018] and XLM-RoBERTa [Conneau et al., 2019], as well as the monolingual Polish models PolBERT [Kłeczek, 2020], SlavicBERT [Arkhipov et al., 2019], HerBERT [Mroczkowski et al., 2021], DistilBERT-PL [Abdaoui et al., 2020], RoBERTa-PL [Dadas et al., 2020b], and TrelBERT [Szmyd et al., 2023]. Some of these models have `base` and `large` variants. These models and their individual properties are outlined in Table 5.2. The smallest model is the `Polish DistilBERT` model at approximately 60 million parameters and largest `XLM-RoBERTa (large)` at about half a billion. The majority of the models hover around the 100 - 200 million range. The last major distinction is in the number of languages used in model training and fine-tuning. All models used solely Polish except for the huge amount of languages used in training `XLM-RoBERTa (large)`, `Multilingual BERT` as well as the Slavic-focused `SlavicBERT`, which was trained on Bulgarian, Czech, Polish, and Russian.

Model	**#params**	**Tokeniser**	**#lgs**	**#steps**	**Vocab**
PolBERT	109M	WordPiece	1	1M	60K
SlavicBERT	180M	WordPiece	4	1M	120K
Multilingual BERT	179M	WordPiece	104	1M	120K
XLM-RoBERTa (large)	561M	SentencePiece UG	94	1.5M	250K
Polish DistilBERT	60.7M	WordPiece	1	N/A	22K
Polish RoBERTa (base) v2	124M	SentencePiece UG	1	400K	50K
Polish RoBERTa (large) v2	345M	SentencePiece UG	1	400k	128K
HerBERT (base)	109M	CharBPE	1	50K	50K
HerBERT (large)	355M	CharBPE	1	60K	50K
TrelBERT	109M	CharBPE	1	1 Ep	50K

Table 5.2: 10 Polish BERT-based models tested on genitive of negation performance in MLM tasks.

5.1.4 Results

The accuracy and certainty scores (as per Section 2.3.3) of each model for each task is visualised in Figure 5.1. A heat map of accuracy results across all models and tasks is shown in Figure 5.2, while scores are averaged across models in Figure 5.3 and across tasks in Figure 5.4. The full results are provided in Appendix B in Table B.1. All visualisations include the results on the OOD datasets.



(a) In-Corpus Accuracy

(b) In-Corpus ASD

(c) OOD Accuracy

(d) OOD ASD

Figure 5.1: Accuracy and Average Surprisal Difference (ASD) of all Polish encoder models for each task plotted against model Certainty. Accuracy determines the proportion of time the grammatical word is selected, while ASD indicates the magnitude of its confidence in the grammatical over the ungrammatical word. Certainty measures whether the model assigned, in general, similar probabilities to a lot of tokens (low score, uncertain) or high probabilities to few tokens (high score, certain).

5.1.5 Discussion

We do not wish to say that the insights offered in this section apply generally to all encoder model trained on Polish, nor do they apply cross-linguistically in any sense. Any generalisations made here are in relation to a limited set of models on a limited set of data, and are not to say that they would hold for all syntactic tests or Polish-language models. However, in combination with further work, they may offer some evidence towards particular conclusions about best practices for model setup and training as well as the difficulty or

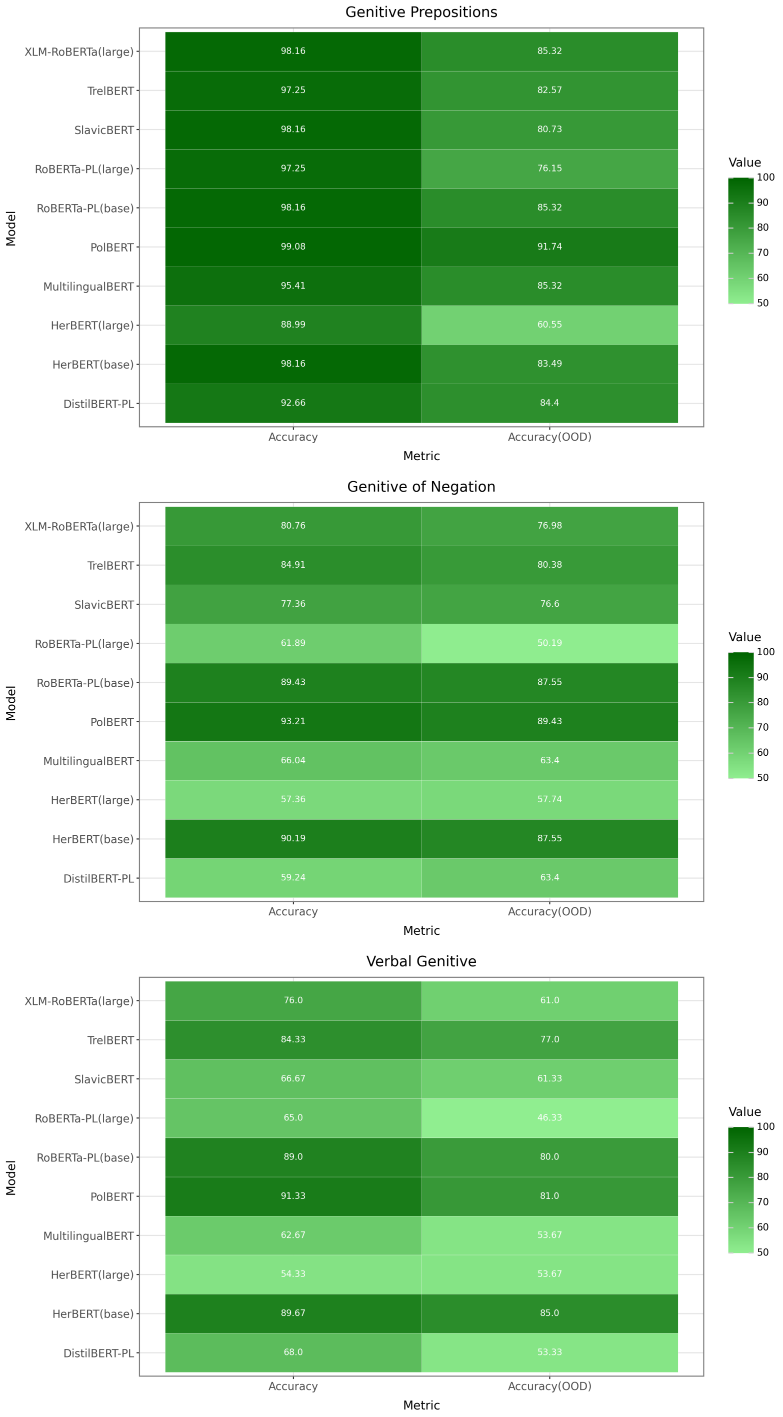


Figure 5.2: Accuracy results across models for the PrepGen, GenOfNeg, and VerbGen tests, respectively. The X axis contains the accuracy for the original and the OOD i.e. semantically implausible minimal pairs. The Y axis contains all models tested in this experiment. Darker indicates a higher accuracy.

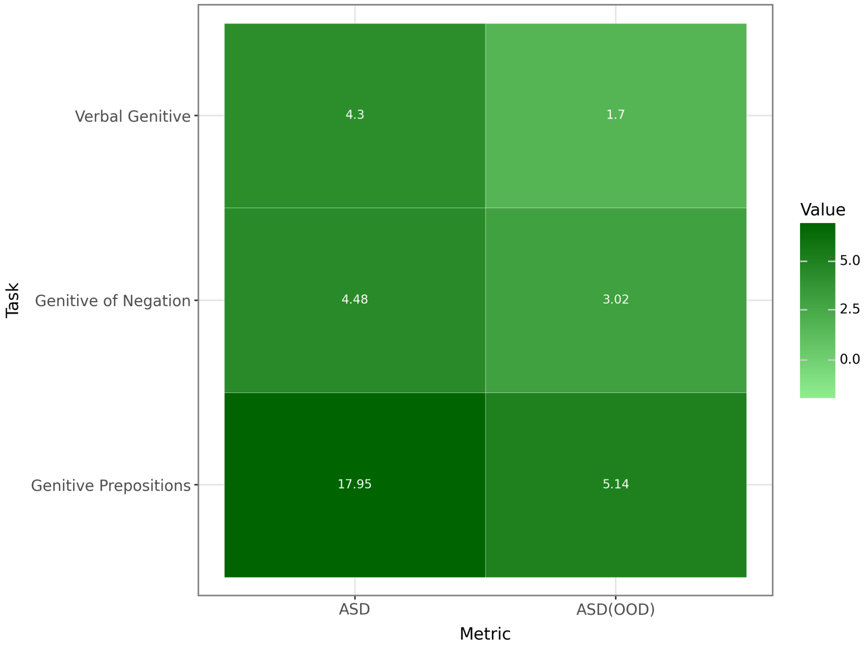


Figure 5.3: Mean ASD score across all models. An ASD score > 0 indicates a tendency towards the grammatical than ungrammatical item on a log base 2 scale.

frequency differences of various syntactic constructions. Three models have performed surprisingly well - `PolBERT`, `HerBERT(base)`, and `RoBERTa-PL(base) v2` - while the others lag behind. Interestingly, the worst-performing model is `HerBERT(large)`, the larger version of the best-performing model (more on this in Section 5.1.5). The model with the lowest general certainty was `DistilBERT`. The PrepGen task appeared to be the easiest and VerbalGen the most difficult. The visualisation of ASD would suggest that, for the most part, models tend towards the correct answer. We observe that model accuracy tends to be correlated with certainty. That is, the more adept a model becomes at predicting the correct word, the more certain about its answer it tends to be.

We do not find strong evidence that scaling alone is correlated with an improved capacity for absorbing language syntax, but rather that hyper-parameter choice as well as data quality all appear to play an important role. The success of the smaller models `PolBERT` and `HerBERT` may possibly be attributed to (1) high-quality training data, (2) their high numbers of Polish-focused tokens - with 60K and 120K, respectively, as well as (3) their high number of training

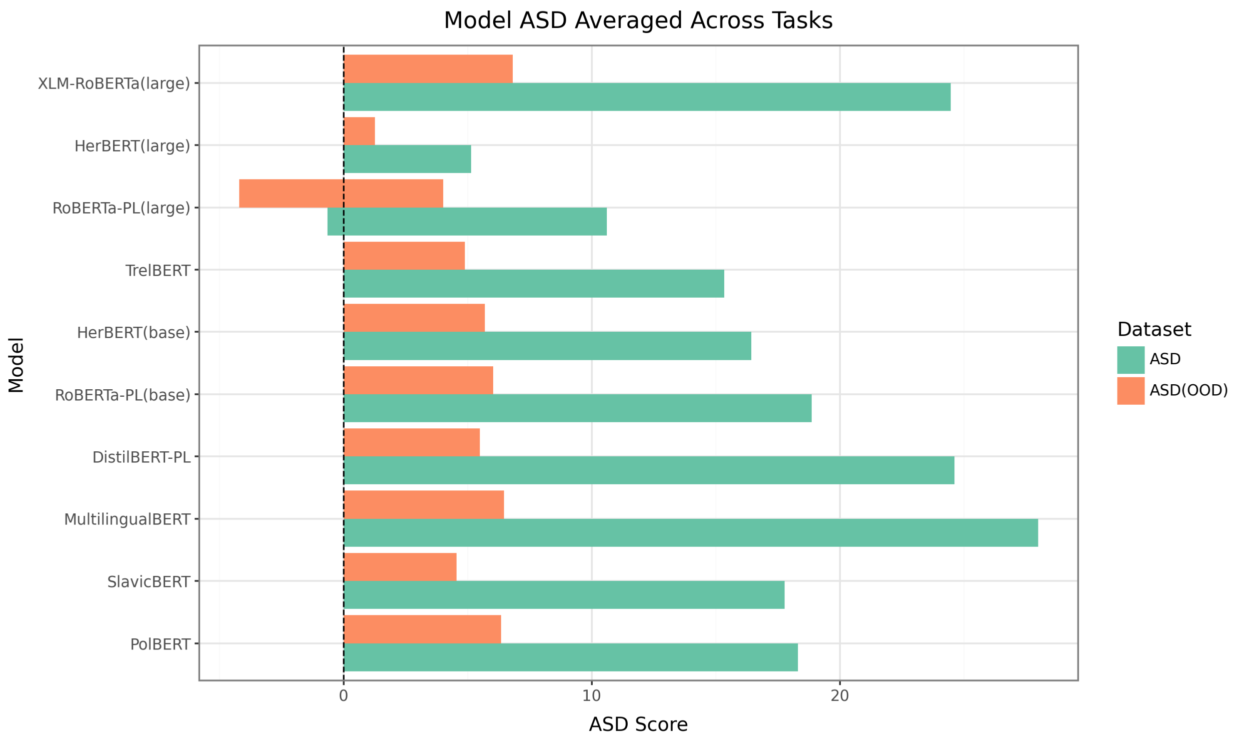


Figure 5.4: Polish-language BERT ASD Scores Across 3 Syntactic Constructions. Green indicates the original minimal pairs and orange the OOD i.e. semantically implausible minimal pairs. A higher value is better and indicates a higher average surprisal difference across all tests, though this can be skewed by outliers.

steps at 1 million each.

We can see in Figure 5.3 that models have a significantly higher tendency to strongly correctly predict the genitive case over the more common locative case for genitive-licensing prepositions, while they do not fare so well for the genitive of negation and verbal genitive constructions. It is clear that certain models struggle across these two latter tasks, such as `HerBERT(large)`, `RoBERTa-PL(large)`, `DistilBERT-PL`, and `MultilingualBERT`. This may be partially due to the short-distance dependencies present in the PrepGen task while the other two tasks require longer dependencies. The VerbGen task also may require the model to first rote-learn the verbs that take a genitive direct object and then generalise for variations of person, number, and so on. Concerning the task of parsing syntactic meaning, the choice of tokenisation algorithm seems like a strong candidate for having an effect on performance.

We see however similar performance across different tokenisers for the highest-performing models. Unlike analytic languages such as Chinese or Japanese that do not have significant morphological inflection, Polish relies heavily on morphemes at the sub-word level to express meaning. We may therefore expect that a tokeniser that can better identify these atomic units of meaning as individual tokens may be better able to aid the model in both syntactic and semantic makeup. However, we do not find evidence of one dominating tokeniser for these syntactic tests. We may summarise the insights that we can gather for these syntactic tests as follows:

1. Scaling of model and pre-training data alone may not improve syntactic performance despite improving semantic performance.
2. We observe a consistent decrease in model performance on OOD datasets than the original. Semantically unlikely sentences significantly affect a model's confidence in the grammatical word over the ungrammatical, but in most cases doesn't lead to significant accuracy decreases, indicating that some generalisation may have taken place.
3. The state-of-the-art models for the Polish KLEJ benchmark in the semantic domain do not perform as well in the syntactic domain for these tests.

Scaling Alone May Not Improve Syntactic Performance

The three top-performing models `PolBERT`, `HerBERT (base)`, and `RoBERTa (base) v2` all hover around approximately 110 million parameters. Of the range of models tested this is relatively low. While there are large models that perform well such as `XLM-RoBERTa (large)`, there are also those such as `HerBERT(large)` and `RoBERTa-PL(large) v2` that perform quite poorly. Particularly of interest is that the smaller `base` versions of these same models (12 layers, 12 attention heads, hidden dim. 768) perform consistently better than their `large` counterparts (24 layers, 16 attention heads, hidden dim. 1024) on these syntactic tests. This is despite the larger variants performing better on the KLEJ semantic benchmark. In Figure 5.3, `HerBERT(large)` drops 7 places down in the rankings when compared to where it stood for semantic tasks and, as seen in Figure 5.4, is the overall worst-performing model on the OOD datasets. One key factor here may be the quality of the data that was fed to the models and how that quality changed in the training of larger models. It is typical as models are scaled that the quality of their training data deteriorates [Goel et al., 2025]. Indeed, Mroczkowski et al. [2021] claim in their paper for the `HerBERT` models that the dataset for the large model is over five times bigger than for the smaller model and, most importantly, contains texts of a lower quality (CCNet and OpenSubtitles). Both models have similar vocabulary sizes and training steps, making them quite comparable. Similarly, Dadas et al. [2020b] also use a smaller but higher quality dataset for the `base` model that may contain more standard syntax and a much larger and coarser dataset for the `large` model.

It has been shown that larger models have a greater capacity for generalisation and hence compression of knowledge representations than smaller models [Lotfi et al., 2024]. However, recent research has also shown that larger models do not always appear to generalise better over their smaller counterparts due to over-fitting [Zhao et al., 2025] and that a controlled training data distribution may be a key factor in achieving a structure-aware model [Xu et al., 2025b]. The data degradation in these two `large` models against their `base` counterparts may therefore partially explain their overall poorer performance on syntactic tasks, despite their overall better performance on semantic tasks.

Consistent Performance Drop on OOD Tests

A consistent decrease in the confidence of the models for the OOD dataset versus the original dataset is observed across most models and tasks, however the ASD score in most cases remains above zero, indicating that some generalisation may have taken place. In Figure 5.1(d) we see a steep pull back in the average magnitude of the model's confidence in the grammatical token over the ungrammatical token. We have discussed a number of reasons why this may occur, such as non-additive semantic bias effects as well as a lack of generalisation (see Section 3.1), however the fact that the ASD remains above zero in most cases for both the within-distribution and out-of-distribution datasets indicates that *some* generalisation of the syntax. The accuracy is not as strongly affected by the OOD data as the ASD scores.

State-of-the-Art on Semantic KLEJ Fails on Syntax

Model	Mean Accuracy	KLEJ Rank Change
PolBERT	94.54%	↑+6
HerBERT (base)	92.67%	↑+2
RoBERTa-PL (base) v2	92.20%	0
TrelBERT	88.83%	↑+1
XLM-RoBERTa (large)	84.97%	↑+1
SlavicBERT	80.73%	↑+2
RoBERTa-PL (large) v2	74.71%	↓-7
Multilingual BERT	74.71%	↑+1
HerBERT (large)	66.89%	↓-7

Table 5.3: How performance rankings change in this syntactic test when compared with the semantic KLEJ benchmark. Rankings are compared only with each other and not with models that do not appear here. Note that DistilBERT was never submitted to KLEJ and hence does not appear.

The results show that the current top performers on the KLEJ semantic tasks do not hold the same rankings when it comes to these three syntactic tasks, as seen in Table 5.3. The KLEJ ranking change shows how these models change with respect to each other but does not take into account the models which were not tested in this experiment. In the original benchmark, `PolBERT` is ranked 18th and `HerBERT(base)` 8th, while `Polish RoBERTa(large) v2` and `HerBERT(large)` are ranked 1st and 2nd, respectively. This indicates that performance on semantic tasks may *not* be a strong indicator of performance on syntactic tasks. This does not hold for just one of the constructions but all three, as we see `PolBERT` makes a huge leap above all models in all three tasks. Note that two of the top-performing KLEJ models - `XLM-RoBERTa (large)` and `Polish RoBERTa(large) v2` fine-tuned on the NKJP corpus [Przepiórkowski et al., 2010] - ranked 3rd and 4th, respectively - were unfortunately not publicly available and therefore could not be tested. A more broad study of syntax in these models would be fruitful future work in order to determine why it might be that some models perform unexpectedly well on syntactic tasks.

5.1.6 Conclusion

In this experiment we performed a fine-grained syntactic performance evaluation of a number of `BERT` models on the genitive case in Polish. The three core findings were (1) model scaling alone appears to not play a key role in syntactic performance, (2) models did not perform as well on out-of-distribution as within-distribution minimal pairs, and (3) models ranked highly on the standard benchmark for Polish semantic tasks were outperformed by a number of lower-ranked models for these syntactic tasks. Future work may identify other typological datasets that could be tested for Polish and additionally include more of the models tested in the `KLEJ` benchmark.

5.2 Conditional Auxiliaries in German

We will next perform syntactic evaluation of five models on German subject-verb-object (SVO) and subject-object-verb (SOV) constructions for number agreement. These models are all transformer-based decoder models that are trained on the task of NTP. As discussed in Section 4 our pipeline will calculate the probability of a given token as $P(\text{token}_i \mid \text{token}_{<i})$ and if a word consists of a number of tokens n then the probability of that word is calculated as $\prod_{i=1}^{n} P(\text{token}_i \mid \text{token}_{<i})$. Distinct from the previous experiment, we therefore must focus on choosing constructions where the necessary grammatical feature is determined in the context prior to the target word itself. For this, we have chosen SVO and SOV conditional auxiliary constructions in German as the number agreement for the verb is determined by the subject which appears before the verb in all constructions tested. In Section 5.2.1 we will look in closer detail at the constructions we are testing. Section 5.2.2 and 5.2.3 introduce the data as well as models used in the experiment, while Section 5.2.4 shows the results which are further discussed in Section 5.2.5.

Syntactic Evaluation of German

German is a language in the West Germanic branch of the Indo-European language family closely related to Dutch and English. It is syntactically characterised by its inflection of nouns, pronouns, and adjectives for the four cases nominative, accusative, dative, and genitive, its free word order with certain restrictions such as verb-second requirements, and a mixture of strong and weak verbs with differing inflections. Despite being a majority language, there are limited experiments in German TSE that have been carried out on transformer-based encoder or decoder models trained partially or fully on German. Zaczynska et al. [2020b] performed experiments on a wide range of subject-verb agreement phenomena as well as reflexive anaphora for two transformer-based encoder models `GBERTlarge` and `DistilGBERT`, finding that the larger model performed better across nearly all tasks except for simple sentence subject-verb agreement for which `DistilGBERT` performed better. There are also limited resources for German syntactic tests in terms of linguistic minimal-pair benchmarks, however Jumelet et al. [2025] introduced MultiBLiMP which contains huge amounts of linguistic minimal pairs for 2 subject-verb agreement phenomena across 101 languages including German.

5.2.1 Constructions

In this experiment we will focus on both SVO and SOV conditional auxiliary constructions in German both for 3rd-person singular and plural. The verb *können* 'can / to be able to' will be the focus of the experiment and we will always test its conditional form "could / would be able to". The conditional inflection in the 3rd-person singular is the suffix *-te*, while for the 3rd-person plural is the suffix *-ten*. We can see an example for each permutation of SVO/SOV and singular/plural in 12.

(12) *SVO 3rd-Person Singular*

Er	könn-**te**	das	Buch	lesen
he.NOM	can-**3SG.SBJV**	the.ACC	book	read.INF

'He could read the book.'

SVO 3rd-Person Plural

Sie	könn-**ten**	die	Zeitung	kaufen
they.NOM	can-**3PL.SBJV**	the.ACC	newspaper	buy.INF

'They could buy the newspaper.'

SOV 3rd-Person Singular

...	dass	er	das	Buch	lesen	könn-**te**
	that	he.NOM	the.ACC	book	read.INF	can-**3SG.SBJV**

‘. . . that he could read the book.’

SOV 3rd-Person Plural

... dass sie die Zeitung kaufen könn-**ten**
that they.NOM the.ACC newspaper buy.INF can-**3PL.SBJV**

‘. . . that they could buy the newspaper.’

We must however be cautious when choosing constructions to test models trained for next-token prediction due to polysemous words swaying the results as well as next-token recovery biases, as outlined in Section 3.2. This condition describes the requirement that when testing grammatical/ungrammatical minimal pairs on a decoder model, we must be careful that the ungrammatical item cannot become grammatical merely through the addition of more tokens. One such danger with these constructions is the polysemous pronoun *sie* as it can mean either “she” or “they”. We therefore remove any constructions that include this pronoun, relying rather on feminine or plural noun forms. Our SOV examples, on the other hand, may cause a next-token recovery bias due to verb chaining at the end of a sentence.

An additional issue is that of *verb chaining*, that is, the chaining together of multiple verbs at the end of a German sentence. This can lead our ungrammatical item to later become grammatical. We can observe an instance of this in Example 13.

(13) ... dass er kommen können muss
that he.NOM come.INF can.INF must.3SG.PRES
‘. . . that he has had to be able to come.’

Furthermore, Example 13 shows that the infinitival form “to be able to” is the same as the 3rd-person plural form “they are able to”. In other words, we cannot say that *können* is an ungrammatical choice in this instance as the addition of more tokens may lead to *können muss* “he/she/it has to be able to” which would render it grammatical. In Example 14 however, we exchange the infinitival for the conditional verb form causing the sentence to become ungrammatical.

(14)	* ... dass	er	kommen	könnten✗	muss
	that	he.NOM	come.INF	can.3PL.SBJV	must.3SG.PRES

‘. . . that he must be able to come.’

The conditional form therefore blocks any further verb chaining in German and alleviates this issue.

5.2.2 Data

A full overview of the minimal-pair datasets generated for this experiment is given in Table 5.4, created through the GrewTSE pipeline applied to the German HDT UD treebank [Borges Völker et al., 2019]. Four datasets are created for each permutation of SVO/SOV and singular/plural constructions. We use relatively small datasets of approximately 50 sentences for each construction, however we used only a subset of the treebank collection due to resource limitations and future work may increase the size of the dataset. The full Grew queries provided as input to the pipeline are provided in Appendix A.

5.2.3 Models

The models were chosen based on their performance on the SuperGLEBer semantic benchmark for German [Pfister and Hotho, 2024] as well as their size. Only models that were approximately half a billion parameters or below were chosen for testing due to hardware restrictions. We examine Qwen 2.5 0.5B [Yang et al., 2024a, Team, 2024], LiquidAI's LFM2-350M [LiquidAI], LLäMmlein 120M [Pfister et al., 2025], Faust GPT-2 [Staatsbibliothek], and Bloom 560M [Workshop, 2023]. Details of the models and their properties can be observed in Table 5.5. All models use a Byte-Pair Encoding (BPE) tokeniser and are in the range of 100 - 560 million parameters in size. LiquidAI release dedicated Small Language Models (SLMs) that can run locally on relatively low- or average-cost hardware and their LFM2-350M model is one of their smallest at 350 million parameters. Bloom 560M, on the

GrewTSE Dataset	**ID**	**Description**	**Size**
SOV Singular	SOV-SG	Tests the model in predicting the correct conditional form of *können* in subordinate clauses with *subject-object-verb* order and singular subjects.	54
SOV Plural	SOV-PL	Tests the model in predicting the correct conditional form of *können* in subordinate clauses with *subject-object-verb* order and plural subjects.	48
SVO Singular	SVO-SG	Tests the model in predicting the correct conditional form of *können* in main clauses with *subject-verb-object* order and singular subjects.	53
SVO Plural	SVO-PL	Tests the model in predicting the correct conditional form of *können* in main clauses with *subject-verb-object* order and plural subjects.	46

Table 5.4: Overview of the four minimal-pair datasets for testing German conditional form of *können* in different word orders (SVO vs SOV) and subject numbers (singular vs plural), generated by the GrewTSE pipeline.

other hand, is somewhat larger but trained on a greater number of languages than all other models.

Model	**#params**	**Tokeniser**	**#lgs**
Qwen 2.5 0.5B (UnslothAI)	500M	BPE	13
LiquidAI LFM2-350M	354M	N/A	8
LLaMlein 120M	138M	BPE	1
Faust GPT2	124M	BPE	1
Bloom 560M	560M	BPE	46

Table 5.5: Decoder-only models used in the experiment and a number of their properties.

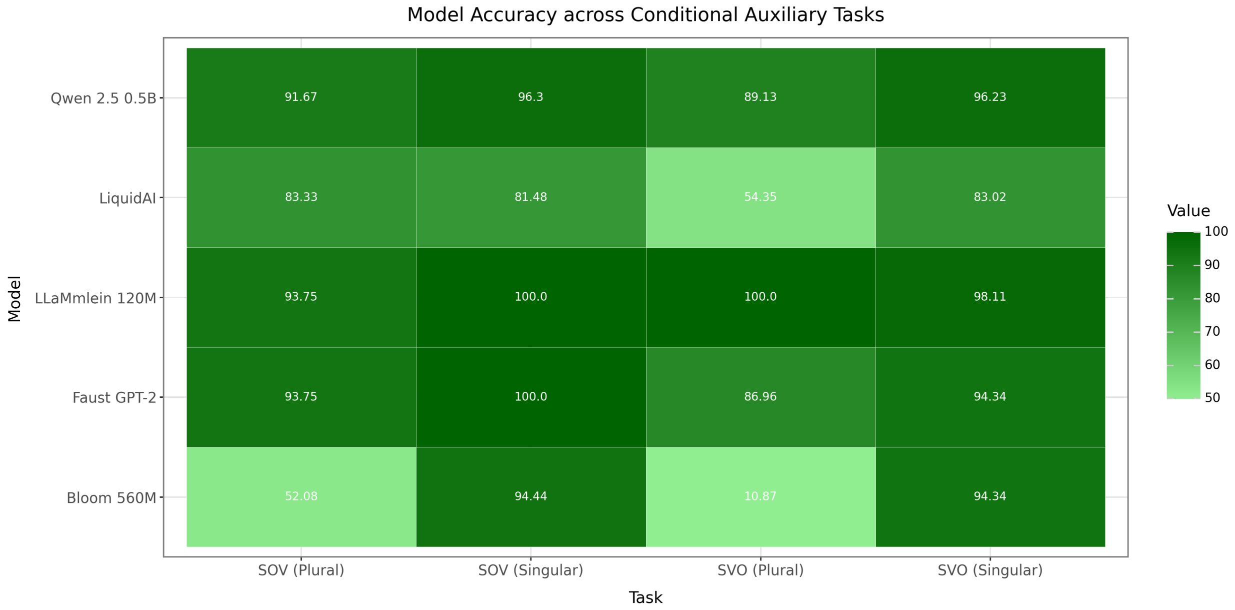


Figure 5.5: Accuracy across all models and tasks for subject-verb-object and subject-object-verb conditional auxiliaries.

5.2.4 Results

A heat map visualising performance across each permutation of singular / plural and SVO / SOV is shown in Figure 5.5. The full results are provided in Table B.2, Appendix B. In this case, we did not perform experiment for out-of-distribution minimal pairs due to the limited number of auxiliaries in German and the likelihood that they would be equally semantically plausible depending on context. ASD results were not visualised for this experiment as they did not contribute significantly to the final conclusions, however they can be seen in the appendix. We did not include as many visualisations as in the previous experiment as we did not feel they were necessary for our conclusions.

5.2.5 Discussion

The results indicate that the best-performing model on these four syntactic tests was `LLaMmlein 120M` and the poorest-performing `Bloom 560M`. Despite the larger number of parameters, the issue likely lies in the much larger number of languages seen in the training data by the poorer model than the German-focused training of the others. Similarly, we see a drop in performance for `LiquidAI LFM2-350M` which is trained on 8 languages. We find no strong evidence that models tend to perform worse on long-distance dependencies over short-distance dependencies, however we do see some unexpected effects of `singular` versus `plural` which we will discuss in Section 5.2.6.

5.2.6 *Semantic Rankings Remain Consistent*

The models show more alignment in syntactic-semantic performance than in Section 5.1 for the Polish models. Models along with their average accuracies are shown in Figure 5.6. There is a slight improvement in `Faust GPT-2` over `Qwen 2.5 0.5B`, however the difference in accuracy is too negligible and the dataset too small to be noteworthy. It would therefore be interesting to investigate the reasons why some models in Section 5.1 shot to the top of the syntactic rankings but remained low in the semantic rankings while the same effect was not observed in this experiment.

Model	Mean Accuracy	`SuperGLEBer` Rank Change
LLäMmlein 120M	97.97%	0
Faust GPT-2	93.76%	↑+1
Qwen 2.5 0.5B (UnslothAI)	93.33%	↓-1
Bloom 560M	62.93%	0

Table 5.6: How performance rankings change in this syntactic test when compared with the semantic `SuperGLEBer` benchmark. Rankings are compared only with each other and not with models that do not appear here. Note that `LiquidAI's LFM2-350M` was never submitted and hence does not appear.

Frequency Bias

We can see in Figure 5.6 that the frequency of the singular form *könnte* versus the plural form *könnten* is consistently higher when we look at Google books N-Gram statistics. Models tend to perform worse on constructions that are less frequent in the training data than those that

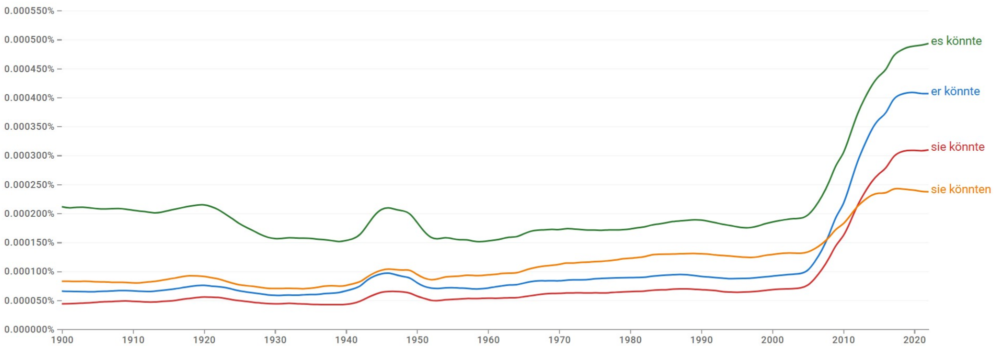


Figure 5.6: Frequency of auxiliary 'könnte' (singular) versus 'könnten' (plural or formal) in German texts over time.

are more frequent [Gulordava et al., 2018]. The lower overall frequency of the 3rd-person plural form versus the 3rd-person singular form, coupled with the fact that the 3rd-person singular form is also the same as the 1st-person singular form, may therefore play a role in why the models tend to perform worse on plural✓/singular✗ over singular✓/plural✗ minimal pairs.

A particularly stark result is how poorly `Bloom 560M` performs on any syntactic test where the plural form is grammatical. This is despite the fact that some ambiguities were removed such as the pronoun *sie* 'she / they'. For the case of SVO constructions which take a plural verb form, the model in fact only predicts the grammatical form in 10.87% of cases. In contrast, for SVO constructions that take a singular verb, the model makes the correct prediction in 94.34% of cases. While the results are not as stark as this in SOV constructions, they are still striking. We can therefore observe that the model appears to be heavily biased towards the singular auxiliary form *könnte*. Future work may look at expanding this surprising finding to other auxiliaries and gaining an insight into whether this phenomenon has occurred in this model on an individual basis for this singular verb or whether it has simply failed to learn more broadly.

Potential Subsequence Biases

One possible reason that may explain the poor results for is that of a *subsequence* bias, as discussed in Section 3.3. We can observe that the singular form ‘könnte’ is a substring of the plural form ‘könnten’ resulting in a possible bias towards the singular form due to tokenisation overlap. In some cases, the models *do* perform better in predicting the plural over the singular such as `LiquidAI LFM2-350M` for SOV or `LLaMmlein 120M` for SVO constructions. However, in the vast majority of cases they perform better on predicting the singular form. While exploring these issues further was outside the scope of this thesis, we did decide to formalise this issue in Section 3.3 and adjust the `GrewTSE` pipeline implementation to automatically avoid such biases for Section 5.3.

5.2.7 Conclusion

We have examined both monolingual and multilingual decoder-only models on both short- and long-distance German conditional auxiliary constructions. We learned from this experiment that it is significantly more difficult to design syntactic evaluation experiments for decoder-only models than encoder-only models (as we had in Experiment 5.1) due to the much higher probability of both next-token recovery and subsequence biases. For instance, the conditional form was chosen as this would nullify any possibility of verb chaining in the context after the target. The results indicate that in most cases models appear to perform better at predicting the singular correctly than the plural form, possibly due to the lexical item frequency of the singular form over the plural. Building on Experiment 5.1, see additional evidence that monolingual models, despite smaller size, perform better on fine-grained syntactic tasks than multilingual models. We will explore this further in Experiment 5.3 on Georgian.

5.3 Split Ergativity in Georgian

We now aim to bring together what we have learned from Section 5.1 and 5.2 to evaluate models on Georgian, a low-resource language that has been underrepresented in TSE research. Georgian is a Kartvelian language that is syntactically characterised by its derivational morphology, leading to a great diversity of word forms, as well as its split-ergative case alignment system. This is not a language that we speak and therefore the careful design of this evaluation is of key importance. Firstly, we will evaluate only `BERT`-based encoder models trained for the task of MLM, as this includes the full context as opposed to only the context prior to the target token and avoids the aforementioned biases that partially hindered our decoder experiment. This along with the fact that each minimal pair (w_a, w_b) must have different string forms in order to be considered minimises any possibility of a partially or fully invalid syntactic evaluation.

The Split-Ergative Phenomenon

This experiment will concern itself with Georgian's unique case alignment system of *split ergativity.* Ergativity is a morphosyntactic alignment system that uses the same form for the subject of an intransitive sentence (e.g. **he** runs) and the object of a transitive sentence (e.g. she likes **him**). This is known as subject-object *alignment.* We mark this as the *absolutive* case. The subject of a transitive sentence is marked as the *ergative* case. The distinction of this alignment to nominative-accusative can be observed in Figure 5.7. Other such case alignment systems are *transitive*, where the alignment is between the agent and patient, *direct*, where they all receive the same case, and *tripartite*, where each receives a different case. Overall, the distinguishing feature of ergative languages is that the transitive subject takes a special marking [Nash, 2017].

Ergative languages are typically only partially ergative; under certain conditions they display nominative properties. Georgian is one such language and is therefore labelled as

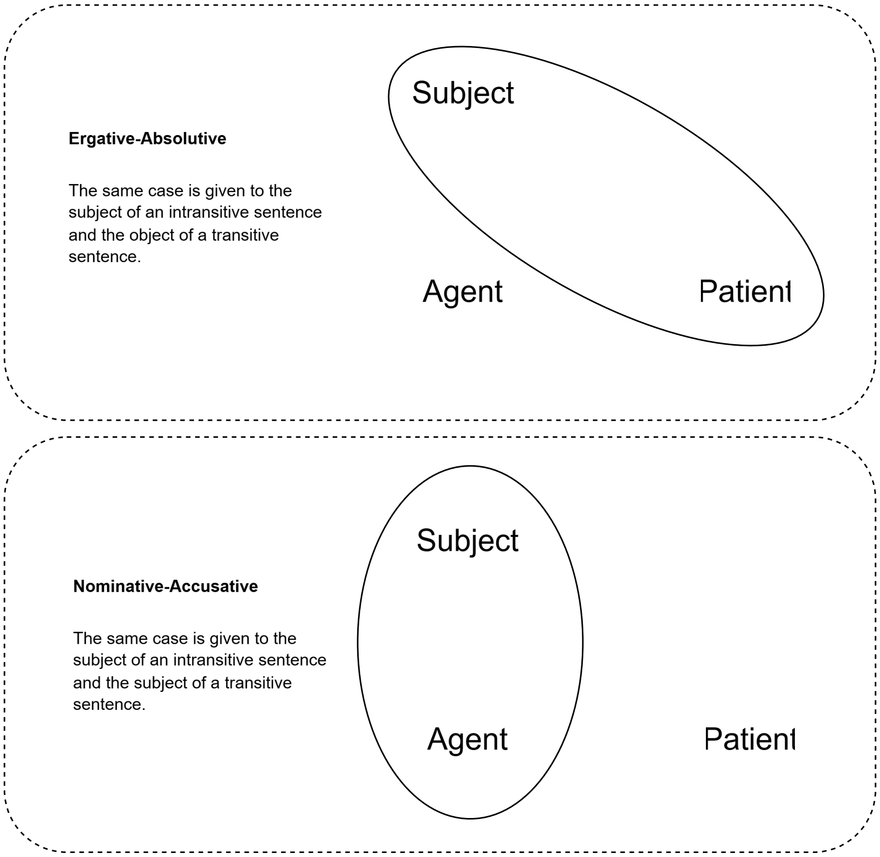

Ergative-Absolutive
The same case is given to the subject of an intransitive sentence and the object of a transitive sentence.
Subject
Agent
Patient
Nominative-Accusative
The same case is given to the subject of an intransitive sentence and the subject of a transitive sentence.
Subject
Agent
Patient

Figure 5.7: The *ergative-absolutive* and *nominative-accusative* case alignment. We can substitute either subject/object or agent/patient.

split-ergative. It exhibits ergative case alignment in the perfective past (or *aorist*) and the conditional, hence assigning the ergative case to the subject of a transitive sentence and absolutive case to the object of a transitive sentence and the subject of an intransitive sentence. Nominative-accusative case alignment on the other hand, as is typical among Indo-European languages, is exhibited in the imperfective past, present, and future. Examples of the Georgian case-alignment system are provided in Example 15 and 16 taken from Nash [2017].

(15) NOMINATIVE TENSES

vano-**ø**	xaT-av-s	/ xaT-av-d-a	mankana-**s**
Vano-**NOM**	draw-3S.TS	/ draw-3SG.TS.PST	car-**ACC**

‘Vano is drawing/was drawing a car.’

vano**-ø** Ċam-ø-s / Ċam-ø-d-a kada**-s**
Vano**-NOM** eat-TS-3S / eat-3SG.TS.PST cake-**ACC**

‘Vano is eating/was eating a cake.’

vano**-ø** alag-eb-s / alag-eb-d-a otax**-s**
Vano**-NOM** tidy-3S.TS / tidy-3S.TS.PST room-**ACC**

‘Vano is/was tidying the room.’

(16) ERGATIVE TENSES

vano**-m** xaT-a / xaT-o-s mankana**-ø**
Vano-**ERG** draw-3SG.AOR / draw-3S.SBJN car-NOM

‘Vano drew/draw a car.’

vano**-m** Ċam-a / Ċam-o-s kada**-ø**
Vano-**ERG** eat-3S.AOR / eat-3SG.SBJN cake-NOM

‘Vano ate/eat a cake.’

vano**-m** alag-a / alag-o-s otax**-i**
Vano-**ERG** tidy-3S.AOR / tidy-3S.SBJN room-NOM

‘Vano tidied/tidy the room.’

Georgian’s split-ergative system is regarded as *active* within particular contexts and *inactive* within others. Ergativity is prominent among the three Caucasian families (Northwest Caucasian, Northeastern Caucasian/Nakh-Dagestanian, South Caucasian/Kartvelian) of northeastern Europe. The majority of languages recorded by `WALS` however have a neutral or nominative-accusative case system. These factors along with Georgian being a low-resource language contribute to this syntactic phenomenon being of particular interest for evaluation in language models.

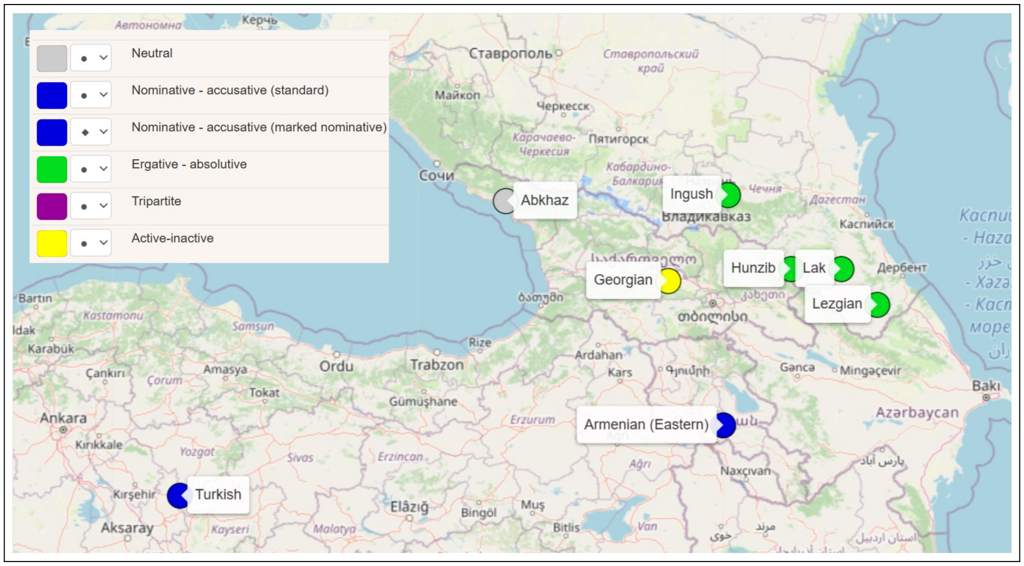


Figure 5.8: Case alignment of languages in and around Georgia as recorded by the WALS dataset Feature 98A: *Alignment of Case Marking of Full Noun Phrases*. Georgian is marked as *active-inactive*, otherwise known as *split ergativity*.

Syntactic Evaluation of Georgian

Georgian is classified as a low-resource language and therefore ought to be the focus of greater research in TSE and has been tested in very few TSE experiments to date. The language's complex system of split ergativity has therefore not been fully evaluated in TSE research and there are still very few mono-lingual language models trained or fine-tuned on Georgian data. Jumelet et al. [2025] include Georgian as one of the languages in their large dataset for TSE covering subject-verb and subject-participle agreement for both `number` and `person`. However, the resources that focus specifically on evaluating this language are next to none, a point that is especially unfortunate given its multitude of unique properties. With this experiment we therefore contribute the first minimal-pair dataset for evaluating syntactic performance on the split-ergative system in Georgian and in doing so provide a small step towards a full benchmark for its syntactic evaluation.

5.3.1 Constructions

We will focus here on three different transitive verb constructions. We emphasise that the above task names correspond only to the following template and make no reference to word order:

SUBJECT-OBJECT

Following this template, the three constructions are labelled and described as follows:

- **Nom-Dat**: Transitive verb constructions with a *nominative* subject and *dative* object.
- **Erg-Abs**: Transitive verb constructions with an *ergative* subject and an *absolutive* object.
- **Dat-Nom**: Transitive verb constructions with a *dative* subject and *nominative* object.

Notice that we use the *dative* case here as a replacement for the *accusative* that was expected. This is due to how Lobzhanidze et al. [2024] decided to label the cases in their Georgian UD treebank and we will continue with their own labelling. The dative is a productive case for the subject in Georgian. This applies to many verbs of perception, possession, affection, belief, thinking, and others. Sentences with an imperfective verb are classified as *active*. The arguments passed to imperfective verbs are typically assigned either the nominative or dative case. The nominative can mark the *agent*, the main argument of state verbs, the *patient* in the main position in passive forms, and the *effector* of the verbs with an inactive subject. The dative case, on the other hand, can mark the *experiencer*, *possessor*, *recipient*, *benefactive* and *patient*. Georgian is unusual within the ergative landscape in that the distribution of ergative and nominative for the agent and subject is not reliant on the transitive/intransitive distinction.

5.3.2 Data

We use the recently published Georgian UD treebank containing 3,013 annotated sentences from Georgian Wikipedia [Lobzhanidze et al., 2024]. We will create separate datasets for masking the subject and the object, with datasets further divided up depending on how the case feature of the target word is adjusted (e.g. Dat → Nom). For each sample, two more minimal pairs are generated for the OOD dataset. We set a maximum of 300 samples for each dataset to avoid individual datasets becoming significantly larger than others, however this issue does still persist as ergative constructions appear to be naturally less frequent in the corpus. Our evaluation will keep the datasets as they are, however some caution should be taken when comparing across constructions especially for the smallest datasets. The full `Grew` queries provided as input to the pipeline are provided in Appendix A. Datasets will be named according to the template:

Subject Case - Object Case: Related Tested → New Case

We can interpret the first part as the original construction specification (e.g. ergative case - nominative case) and the subsequent part as the grammatical relation being targeted as well as to which case it is converted to establish the minimal pair. 5 examples syntactic tests are provided in the below from Example 21.

(17) Nominative - Dative: Subj → Erg

[vano✓/vanom✗] xaTavs mankanas
Vano.NOM/Vano.ERG draw.3SG.TS car.DAT

‘Vano is drawing a car.’

(18) Ergative - Nominative: Subj → Dat

[vanom✓/vanos✗] xaTa mankana
Vano-ERG/Vano-DAT draw-3SG.AOR car-NOM

‘Vano drew a car.’

(19) Dative - Nominative: Subj → Erg

[vanos✓/vanom✗]	mankana-ø	xaT-e-bia
Vano.DAT/Vano.ERG	car-NOM	draw-3SG.PRF

'Vano has drawn the car.'

(20) Nominative - Dative: Obj → Nom

vano	xaTavs	[mankanas✓/mankana✗]
Vano.NOM	draw.3SG.TS	car.DAT/car.NOM

'Vano is drawing a car.'

(21) Ergative - Nominative: Obj → Dat

vanom	xaTa	[mankana✓/mankanas✗]
Vano-ERG	draw-3S.AOR	car-NOM

'Vano drew a car."

This results in the creation of 12 datasets as outlined in Table 5.7 and a resulting total dataset of 1,213 minimal-pair syntactic tests for evaluating encoder models on transitive Georgian case alignment, equating to 766 tests for the Nom-Dat constructions, 187 for the Erg-Nom constructions, and 220 for the Dat-Nom constructions.

Token-Length Bias of Nominative Case

One issue that we encounter in this experiment however is that of bias towards the shorter word in a minimal pair, as discussed in Section 3.3. We can refresh the principle as follows:

> **Subsequence Bias**: Given a minimal pair (w_1, w_2), let their tokenisations be represented as sets of tokens $T_1 = \{\tau_1, \tau_2, \ldots, \tau_m\}$ and $T_2 = \{\tau'_1, \tau'_2, \ldots, \tau'_n\}$. Neither full set may be a subsequence of the other.

This issue can in particular effect languages which have marking for some features and not for others. The nominative case in Georgian is not marked while the dative and ergative cases

Dataset	Subj	Obj	Relation Tested	Minimal Pair	Size
Nom-Dat:Subj⇒Erg	Nom	Dat	Subject	(Nom, Erg)	115
Nom-Dat:Subj⇒Dat	Nom	Dat	Subject	(Nom, Dat)	295
Nom-Dat:Obj⇒Erg	Nom	Dat	Object	(Dat, Erg)	56
Nom-Dat:Obj⇒Nom	Nom	Dat	Object	(Dat, Nom)	300
Erg-Nom:Subj⇒Dat	Erg	Nom	Subject	(Erg, Dat)	66
Erg-Nom:Subj⇒Nom	Erg	Nom	Subject	(Erg, Nom)	70
Erg-Nom:Obj⇒Dat	Erg	Nom	Object	(Nom, Dat)	69
Erg-Nom:Obj⇒Erg	Erg	Nom	Object	(Nom, Erg)	22
Dat-Nom:Subj⇒Erg	Dat	Nom	Subject	(Dat, Erg)	29
Dat-Nom:Subj⇒Nom	Dat	Nom	Subject	(Dat, Nom)	114
Dat-Nom:Obj⇒Erg	Dat	Nom	Object	(Nom, Erg)	10
Dat-Nom:Obj⇒Dat	Dat	Nom	Object	(Nom, Dat)	67

Table 5.7: Overview of the 12 generated datasets for testing transitive sentences in the Georgian case alignment system.

are, resulting in the nominative item in a given minimal pair potentially being a substring of the other and consequently a potential token subsequence. In order to quantify this bias, we therefore check the proportion of tests for each task for which the number of overlapping tokens between items is equal to the length of one of the items, expecting that this will cause a positive push towards predicting the *nominative* case. Indeed, we find for this dataset that 34% of tests suffer from this issue when tokenised with the `HPLT-BERT-ka` tokeniser and 41% with the `Multilingual BERT` tokeniser. We thus present two sets of results: one set covers the full dataset without any alterations, and the other set covers only those samples from the dataset that are not affected by a subsequence bias for a given tokeniser. This will allow us to determine whether any conclusions we draw are being unduly affected by an inherent bias towards the unmarked nominative case.

5.3.3 Models

There is a stark lack of models available for the Georgian language. We therefore focus primarily on the evaluation of multilingual models with the exception of 1 monolingual model. We will be evaluating a number of cased BERT models that have been at least partially trained using Georgian data. While the original BERT model does not meet this requirement, MultilingualBERT does [Devlin et al., 2019, 2018]. XLM-RoBERTa (base) and XLM-RoBERTa (large) were also both trained on a greater number of Georgian texts [Conneau et al., 2019]. Our final model is HPLT-BERT-ka, one of the very few available monolingual models for Georgian [Burchell et al., 2025]. Applicable models were trained and datasets were generated in the original Georgian script, known as *Mkhedruli*, and therefore no transliterations are required for the evaluation. An overview of the models and their properties can be seen in Figure 5.8.

Model	**#params**	**tok**	**#lgs**	**#steps**	**vocab**
Multilingual BERT	179M	WordPiece	104	1M	120K
XLM-RoBERTa (base)	278M	SentencePiece UG	94	1.0M	250K
XLM-RoBERTa (large)	561M	SentencePiece UG	94	1.5M	250K
HPLT-BERT-ka	110M	N/A	1	N/A	32K

Table 5.8: 4 BERT-based models used in the experiment. All are either partially or fully trained on Georgian.

5.3.4 Results

A heat map visualising the results across all minimal-pair types and models is shown in Figure 5.9. A further heat map which removes bias from *subsequence* biases, as described in Section 3.3 and found as a potential issue with experiment design in Section 5.2, is shown in Figure 5.10. A full table of all results are provided in Table B.3, Appendix B. ASD results were not visualised for this experiment as they did not contribute significantly to the final conclusions.

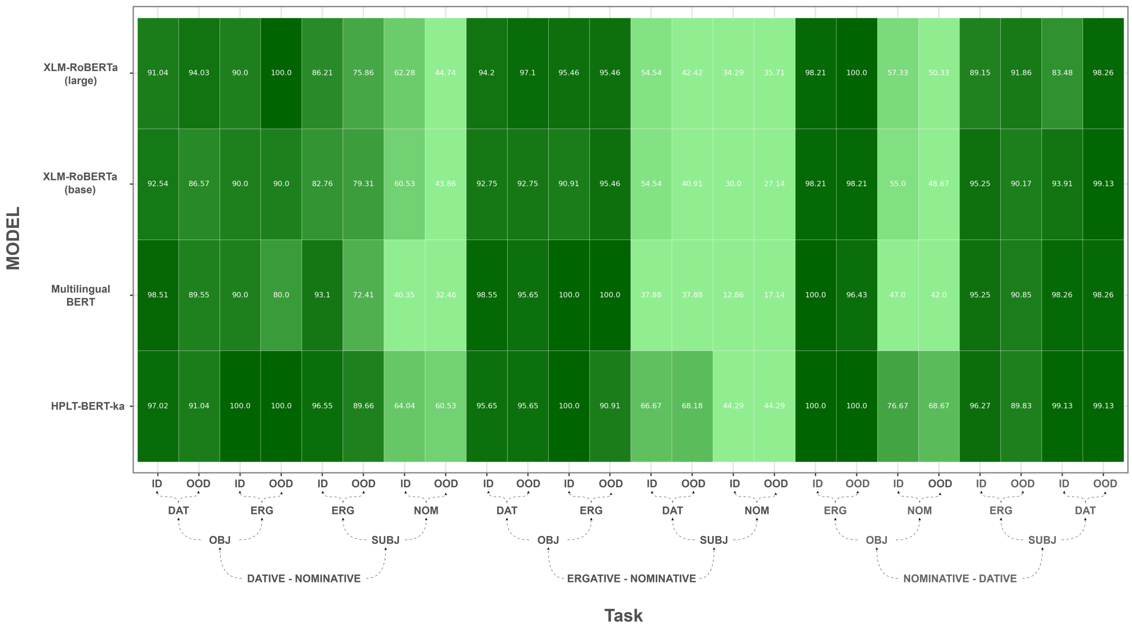


Figure 5.9: Accuracy results for the **full dataset** across all Georgian models and tasks.

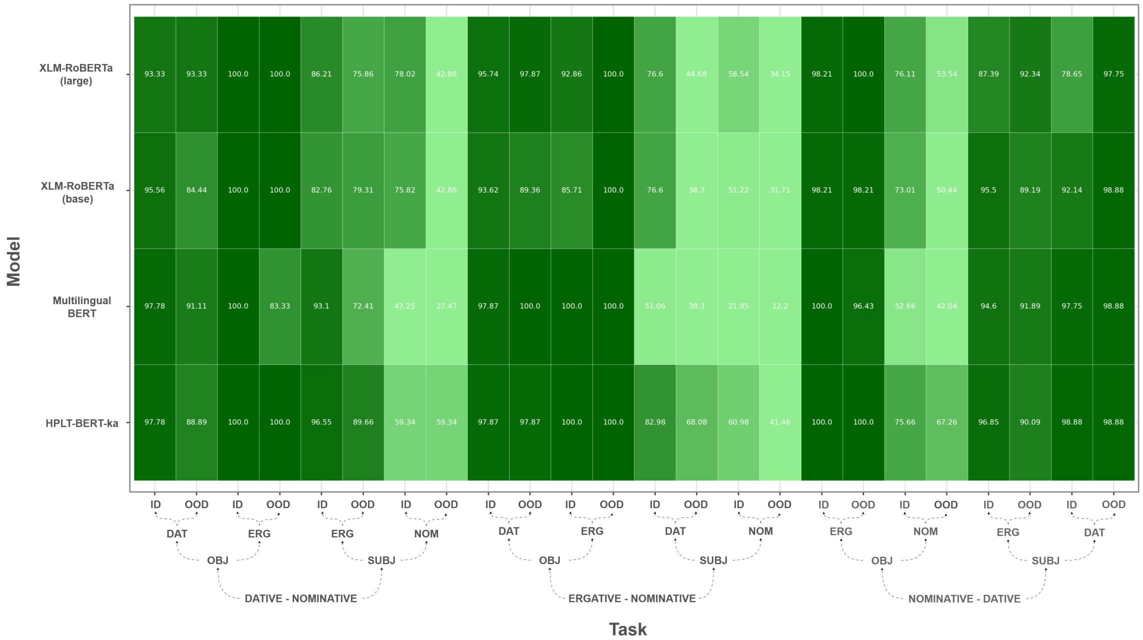


Figure 5.10: Accuracy results for the **adjusted dataset to remove subsequence biases** across all Georgian models and tasks.

5.3.5 Discussion

The results indicate that models appear to approximately align with regard to which syntactic tests are more or less difficult across tasks. They tended to struggle with all tests on the subject in ergative-nominative constructions, which would in this case be tests for the grammatical ergative subject against the ungrammatical dative or nominative subjects, while performing very well when they were tested on a grammatical dative against an ungrammatical ergative as well as a grammatical nominative against an ungrammatical ergative. We observe that the removal of subsequence biases does not largely interfere with the overall patterns that we see within the data, indicating that subsequence biases have not been of significant detriment within this experiment. We do however see the effect of the bias towards the unmarked Georgian nominative case.

Performance Drops on Ergative

We observe that model performance drops significantly when predicting the ergative case in aorist or subjunctive constructions. This is one of the key tasks of our dataset, as it indicates to what extent the model has familiarised itself with the environment in which the ergative case alignment is active. The lack of performance for this feature which occurs only in specific verbal paradigms indicates a lack of generalisation. We would expect given sufficient exposure that the model may learn to associate the aorist verb suffix -*a* with an ergative subject. However, it is possible that the model has not seen enough training samples with the ergative causing a bias towards the other more common cases of nominative and ergative. This is reflected when we analyse the Georgian treebank for nouns with each of these three distinct cases. We find that there are 11,428 nouns with the nominative case, 10,036 with the dative, and only 475 for the ergative. There is an additional clear bias by the model towards the nominative case, even when compared with the dative which appears to have a similar frequency. This can be observed in tasks DATIVE-NOMINATIVE:SUBJ→NOM and Nominative-Dative:Obj→Nom. Interestingly, the effect persists even after removing samples that suffer from subsequence biases as discussed in Section 5.3.2 and shown in Figure 5.9 and 5.10.

Broader Implications for Low-Frequency Syntax

The results outlined here may have broader implications for syntactic performance on LRLs in specific, relatively low-frequency environments such as unique verbal paradigms. This corroborates other findings pertaining to lexical frequency. Kryvosheieva and Levy [2024] found that performance on Basque auxiliary agreement was significantly worse for indirect objects despite high performance in other areas, likely due to their relatively low frequency in the training corpora. Both the inherent distribution of these types of constructions as well as a more limited set of data for LRLs may contribute to this performance drop. We therefore encourage a greater focus on larger and more syntactically balanced datasets being used in training.

5.3.6 Conclusion

An avenue for future work would be to incorporate a wider range of ergative constructions as well as a greater number of ergative languages. Here we have focused on transitive constructions for instance, which lacks insight into performance on intransitive constructions. While some work has been carried out for the ergative languages Georgian and Basque, it is still a research area that has been largely under-explored. As a greater number of treebanks are annotated and more models are trained for ergative languages, it is our hope that a greater amount of effort will be put into this research direction.

5.4 General Discussion

We have carried out experiments across a range of languages, models, and model architectures. We can summarise our findings across all three experiments with the following insights:

1. Models tend to perform better on semantically plausible over implausible minimal pairs, despite the grammatical/ungrammatical relation remaining the same. This was observed across most results.

2. Rarer feature values (e.g. ergative case, auxiliary plural) are consistently underrepresented in model performance. This indicates strong sensitivity to frequency distributions and in some cases a lack of pre-training data.

3. Monolingual models trained on the target language tend to perform better than multilingual models for which the target language made up only a portion of the dataset, despite size advantages of the multilingual models.

Far more experiments would be required in order to make claims as to the nature of different language models or languages, however these patterns apply consistently across experiments and leave interesting avenues for future work. For the encoder models tested on various genitive constructions in Polish, for instance, we saw that the scaling of model and pre-training data alone may not be sufficient to improve syntactic performance despite improvements in semantic performance. Furthermore, the models considered state-of-the-art for semantic tasks fell in their rankings for our syntactic tests. For the decoder models tested on conditional auxiliaries in German, performance appeared to depend more on lexical frequency than on dependency distance, with a heavy bias towards the more-frequent singular forms than the less-frequent plural forms, further corroborated by the final experiment on Georgian split ergativity. We saw that models performed poorly on any task where they must predict the grammatical ergative, and well where they must predict a grammatical nominative or dative over an ungrammatical ergative. Interestingly, there was a natural bias towards the nominative over the dative, despite the dative having a similar distribution within Georgian on the sentences provided by the Georgian UD treebank. Once again the models tended to perform better on the original than the OOD minimal pairs.

One issue that is also highlighted by the results is also that of imbalanced model multilinguality. Often models are presented with a set of languages that they have been trained on, for instance LiquidAI's `LFM2-350M` is listed as a multilingual model that can speak English, Arabic, Chinese, French, German, Japanese, Korean, and Spanish. Sections 5.1, 5.2, and 5.3 have shown us however that these models can perform quite poorly in comparison with their monolingual counterparts and performance seems to deteriorate rapidly for rarer constructions. This indicates that the binary listing of model languages is somewhat misleadingly presented and calls for greater transparency in the proportion of data used for each language in model pre-training. Future work may look towards carrying out a more comprehensive set of experiments cross-linguistically on the same set of syntactic phenomena to determine how performance differs for multilingual models, particularly in the case of low-resource languages.

We thus reintroduce the original research questions outlined in Section 1 and explore what insights we may have achieved through the experiments.

RQ1 *To what extent can transformer-based language models accurately predict key syntactic words or morphemes within syntactic structures across typologically diverse languages, and how does performance vary by syntactic phenomenon?*

RQ2 *How do differences in architecture, size, and tokenisation affect performance on the same syntactic phenomenon?*

RQ3 *What differences do we observe in model performance when we evaluate on syntactic minimal pairs that are semantically implausible, and what does this tell us about a model's generalisation of syntactic structure?*

RQ1 tackled how transformer models perform across a diverse array of syntactic phenomena for the languages analysed in this thesis. We found that both the encoder-only and decoder-only models tested in this thesis tend to perform well across a wide range of syntactic tasks, with especially high performance for those constructions which are both *common* and *consistent* such as prepositional genitive constructions in Polish or conditional auxiliaries in German. However, rarer phenomena in underrepresented languages tended to result in poorer evaluations across models, such as Polish verbal genitive or Georgian ergative constructions.

RQ2 pertained to the differences we seen in model performance depending on architecture, size, and tokeniser. Due to the large number of variables present, it was difficult to make generalisations across any one of these domains. However, we did not find that scaling model size *alone* was sufficient to improve syntactic performance for the phenomena examined in our experiments. In terms of architecture, encoder-only models trained on MLM tasks tended to be far easier to perform evaluations on due to the entire context of the sentence being taken into account. This avoids what we have defined as the *next-token recovery* bias, where models trained on the NTP task suffer from a bias due to a lack of insight into the intended continuation of the sentence after the grammatical/ungrammatical target. We examined a range of tokenisers in the three experiments but further evaluations would be required in order to make any conclusive results about the 'ideal' tokenisation algorithm for syntactic performance.

RQ3 tackled the differences we observed in performance for semantically plausible versus implausible syntactic tests. We found that models tended to perform better on semantically plausible over semantically implausible minimal pairs, despite the grammatical/ungrammatical relation remaining the same. This effect was found across most models and phenomena, with exceptions being relatively close. This may indicate a lack of generalisation, training data bias, or non-independent syntactic-semantic interactions. However, the consistency with which we saw stronger performance on the original minimal pairs indicates that we have found a consistent and systematic syntactic performance degradation for minimal pairs that are semantically implausible and hence possibly unobserved during model training.

CHAPTER 6
CONCLUSION

In this thesis we introduced a novel means of creating minimal-pair datasets for TSE research through the use of Grew-based treebank querying. The pipeline is released as an open-source Python package that handles TSE dataset generation, model evaluation, and result visualisation. It opens up many possibilities for future work, as it allows the rapid creation of minimal-pair datasets for any syntactic construction that can be described with a Grew query from any of the over 150 languages for which there is a UD treebank available. We discussed a number of issues in this domain such as *next-token recovery* and *subsequence* biases, and additionally argued for the use of *token-level* as opposed to *sentence-level* probabilities for evaluations. In order to examine model generalisation, we implemented a partial solution through the use of semantically implausible minimal pairs. Experiments were carried out for a variety of syntactic phenomena in Polish and Georgian encoder-only models trained on the MLM task as well as German decoder-only models trained on the NTP task. Our core findings were that (1) models tended to perform better on grammatical / ungrammatical minimal pairs that are semantically plausible, indicating a possible lack of generalisation, training data bias, or non-additive semantic violations effects, (2) rarer syntactic constructions are consistently underrepresented in model performance with a possible strong sensitivity to lexical frequency, and (3) models trained on a single target language tended to perform better on syntactic tasks than those trained on many languages. This work was partially limited by resource constraints, with most models tested under approximately 500 million parameters. Another important limitation is that we did not control for sentence length in the datasets and future work may look towards developing the an implementation of these improved controls in the GrewTSE Python package. We especially encourage the use of the GrewTSE pipeline for the evaluation of those rarer constructions in low-resource languages that have been underrepresented in syntactic evaluation research.

REFERENCES

Amine Abdaoui, Camille Pradel, and Grégoire Sigel. Load what you need: Smaller versions of mutlilingual bert. In *SustaiNLP / EMNLP*, 2020.

Mostafa Abdou, Vinit Ravishankar, Artur Kulmizev, and Anders Søgaard. Word order does matter and shuffled language models know it. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6907–6919, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi:10.18653/v1/2022.acl-long.476. URL `Mhttps://aclanthology.org/2022.acl-long.476/`.

Anthropic. On the biology of a large language model. `Mhttps://transformer-circuits.pub/2025/attribution-graphs/biology.html`, March 2025. Accessed: 18-08-2025.

Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, and Alexey Sorokin. Tuning multilingual transformers for language-specific named entity recognition. In Tomaž Erjavec, Michał Marcińczuk, Preslav Nakov, Jakub Piskorski, Lidia Pivovarova, Jan Šnajder, Josef Steinberger, and Roman Yangarber, editors, *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 89–93, Florence, Italy, August 2019. Association for Computational Linguistics. doi:10.18653/v1/W19-3712. URL `Mhttps://aclanthology.org/W19-3712/`.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016. URL `Mhttps://arxiv.org/abs/1409.0473`.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. URL `Mhttps://arxiv.org/abs/2204.05862`.

Ezgi Başar, F.P Padovani, Jaap Jumelet, and Arianna Bisazza. Turblimp: A turkish benchmark of linguistic minimal pairs, 06 2025.

Núria Bel, Marta Punsola, and Valle Ruiz-Fernández. EsCoLA: Spanish corpus of linguistic acceptability. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6268–6277, Torino, Italia, May 2024. ELRA and ICCL. URL `Mhttps://aclanthology.org/2024.lrec-main.554/`.

Emanuel Borges Völker, Maximilian Wendt, Felix Hennig, and Arne Köhn. HDT-UD: A very large Universal Dependencies treebank for German. In Alexandre Rademaker and Francis Tyers, editors, *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 46–57, Paris, France, August 2019. Association for Computational Linguistics. doi:10.18653/v1/W19-8006. URL `Mhttps://aclanthology.org/W19-8006/`.

Laurie Burchell, Ona de Gibert, et al. An expanded massive multilingual dataset for high-performance language technologies (HPLT). In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17452–17485, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi:10.18653/v1/2025.acl-long.854. URL `Mhttps://aclanthology.org/2025.acl-long.854/`.

Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L. Leavitt, and Naomi Saphra. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in mlms, 2025. URL `Mhttps://arxiv.org/abs/2309.07311`.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014. URL `Mhttps://arxiv.org/abs/1406.1078`.

N. Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, 1956. doi:10.1109/TIT.1956.1056813.

Noam Chomsky. *Syntactic Structures*. Mouton, The Hague, 1957.

Noam Chomsky. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA, USA, 1965.

Noam Chomsky. A minimalist program for linguistic theory, 1993. URL `Mhttps://philpapers.org/rec/CHOAMP`.

Noam Chomsky and Ramin Mirfakhraie. Chatgpt and human intelligence: Noam chomsky responds to critics. Interview published on Chomsky.info, April 24 2023. URL `Mhttps://chomsky.info/20230424-2/`. Available online at `Mhttps://chomsky.info/20230424-2/`.

Charles Jr. Clifton, Adrian Staub, and Keith Rayner. Eye movements in reading words and sentences. In Roger P. G. van Gompel, Martin H. Fischer, Wayne S. Murray, and Robin L. Hill, editors, *Eye movements: A window on mind and brain*, pages 341–371. Elsevier, 2007. doi:10.1016/B978-008044980-7/50017-3.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116, 2019. URL `Mhttp://arxiv.org/abs/1911.02116`.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514, 2021. ISSN 2329-9304. doi:10.1109/taslp.2021.3124365. URL Mhttp://dx.doi.org/10.1109/TASLP.2021.3124365.

Slawomir Dadas, Michał Perełkiewicz, and Rafał Poświata. Evaluation of sentence representations in Polish. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1674–1680, Marseille, France, May 2020a. European Language Resources Association. ISBN 979-10-95546-34-4. URL Mhttps://aclanthology.org/2020.lrec-1.207/.

Sławomir Dadas, Michał Perełkiewicz, and Rafał Poświata. Pre-training polish transformer-based language models at scale, 2020b. URL Mhttps://arxiv.org/abs/2006.04229.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In Nicoletta Calzolari, Khalid Choukri, Aldo Gangemi, Bente Maegaard, Joseph Mariani, Jan Odijk, and Daniel Tapias, editors, *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA). URL Mhttps://aclanthology.org/L06-1260/.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. Universal Stanford dependencies: A cross-linguistic typology. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4585–4592, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL Mhttps://aclanthology.org/L14-1045/.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. Universal Dependencies, June 2021. URL Mhttps://aclanthology.org/2021.cl-2.11/.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL Mhttp://arxiv.org/abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL Mhttps://arxiv.org/abs/1810.04805.

Marina Ermolaeva. Deconstructing syntactic generalizations with minimalist grammars. In Arianna Bisazza and Omri Abend, editors, *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 435–444, Online, November 2021. Association for Computational Linguistics. doi:10.18653/v1/2021.conll-1.34. URL Mhttps://aclanthology.org/2021.conll-1.34/.

Richard Futrell and Kyle Mahowald. How linguistics learned to stop worrying and love the language models. *Behavioral and Brain Sciences*, page 1–98, July 2025. ISSN 1469-1825. doi:10.1017/s0140525x2510112x. URL Mhttp://dx.doi.org/10.1017/S0140525X2510112X.

Richard Futrell, Kyle Mahowald, and Edward Gibson. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341, 2015. doi:10.1073/pnas.1502134112. URL Mhttps://www.pnas.org/doi/abs/10.1073/pnas.1502134112.

Philip Gage. A new algorithm for data compression. *C Users J.*, 12(2):23–38, February 1994. ISSN 0898-9788.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning, 2017. URL Mhttps://arxiv.org/abs/1705.03122.

Yash Goel, Ayan Sengupta, and Tanmoy Chakraborty. Position: Enough of scaling llms! lets focus on downscaling, 2025. URL Mhttps://arxiv.org/abs/2505.00985.

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation, 2015. URL Mhttps://arxiv.org/abs/1502.04623.

Gaël Guibon, Marine Courtin, Kim Gerdes, and Bruno Guillaume. When collaborative treebank curation meets graph grammars. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5291–5300, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL Mhttps://aclanthology.org/2020.lrec-1.651/.

Bruno Guillaume. Graph Matching for Corpora Exploration. In *JLC 2019 - 10èmes Journées Internationales de la Linguistique de corpus*, Grenoble, France, November 2019. URL Mhttps://inria.hal.science/hal-02267475.

Bruno Guillaume. Graph Matching and Graph Rewriting: GREW tools for corpus exploration, maintenance and conversion. In *EACL 2021 - 16th conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, Kiev/Online, Ukraine, April 2021a. URL Mhttps://inria.hal.science/hal-03177701.

Bruno Guillaume. Graph Matching and Graph Rewriting: GREW tools for corpus exploration, maintenance and conversion. In *EACL 2021 - 16th conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, Kiev/Online, Ukraine, April 2021b. URL Mhttps://inria.hal.science/hal-03177701.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. Colorless green recurrent networks dream hierarchically. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi:10.18653/v1/N18-1108. URL `Mhttps://aclanthology.org/N18-1108/`.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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)*, pages 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi:10.18653/v1/N19-1419. URL `Mhttps://aclanthology.org/N19-1419/`.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory, 11 1997. ISSN 0899-7667. URL `Mhttps://doi.org/10.1162/neco.1997.9.8.1735`.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. A systematic assessment of syntactic generalization in neural language models. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online, July 2020. Association for Computational Linguistics. doi:10.18653/v1/2020.acl-main.158. URL `Mhttps://aclanthology.org/2020.acl-main.158/`.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks, 2016. URL `Mhttps://arxiv.org/abs/1506.02025`.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy, July 2019. Association for Computational Linguistics. doi:10.18653/v1/P19-1356. URL `Mhttps://aclanthology.org/P19-1356/`.

Jaap Jumelet, Leonie Weissweiler, Joakim Nivre, and Arianna Bisazza. Multiblimp 1.0: A massively multilingual benchmark of linguistic minimal pairs, 2025. URL `Mhttps://arxiv.org/abs/2504.02768`.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in c++, 2018. URL `Mhttps://arxiv.org/abs/1804.00344`.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time, 2017. URL `Mhttps://arxiv.org/abs/1610.10099`.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. Mission: Impossible language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14691–14714, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.acl-long.787. URL `Mhttps://aclanthology.org/2024.acl-long.787/`.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL `Mhttps://arxiv.org/abs/2001.08361`.

Ronald Kaplan and Joan Bresnan. Lexical-functional grammar: A formal system for grammatical representation, 01 1982.

Darek Kłeczek. dkleczek/bert-base-polish-uncased-v1. `Mhttps://huggingface.co/dkleczek/bert-base-polish-uncased-v1`, 2020. Hugging Face model card.

Greg Kobele. Lexical decomposition. `Mhttps://home.uni-leipzig.de/gkobele/courses/2018.SS/MGs/decomposition.pdf`, 2018. Accessed: 2025-08-29.

Gregory M. Kobele. Minimalist grammars and decomposition. submitted, 2021, 2021.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In Sophia Ananiadou, editor, *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `Mhttps://aclanthology.org/P07-2045/`.

Katarzyna Krasnowska-Kieraś and Alina Wróblewska. Empirical linguistic study of sentence embeddings. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5729–5739, Florence, Italy, July 2019. Association for Computational Linguistics. doi:10.18653/v1/P19-1573. URL `Mhttps://aclanthology.org/P19-1573/`.

Daria Kryvosheieva and Roger Levy. Controlled evaluation of syntactic knowledge in multilingual language models, 2024. URL `Mhttps://arxiv.org/abs/2411.07474`.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi:10.18653/v1/D18-2012. URL `Mhttps://aclanthology.org/D18-2012/`.

Artur Kulmizev. *The Search for Syntax: Investigating the Syntactic Knowledge of Neural Language Models Through the Lens of Dependency Parsing*. Phd dissertation, Acta Universitatis Upsaliensis, 2023. URL Mhttps://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-508379.

Artur Kulmizev and Joakim Nivre. Schrödinger's tree – on syntax and neural language models, 2021. URL Mhttps://arxiv.org/abs/2110.08887.

Artur Kulmizev and Joakim Nivre. Investigating UD treebanks via dataset difficulty measures. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1076–1089, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi:10.18653/v1/2023.eacl-main.76. URL Mhttps://aclanthology.org/2023.eacl-main.76/.

Shalom Lappin and Stuart Shieber. Machine learning theory and practice as a source of insight into universal grammar. *Journal of Linguistics*, 43:393 – 427, 07 2007. doi:10.1017/S0022226707004628.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models, 2024. URL Mhttps://arxiv.org/abs/2402.14848.

Roger Levy and Galen Andrew. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In Nicoletta Calzolari, Khalid Choukri, Aldo Gangemi, Bente Maegaard, Joseph Mariani, Jan Odijk, and Daniel Tapias, editors, *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA). URL Mhttps://aclanthology.org/L06-1311/.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019. URL Mhttps://arxiv.org/abs/1910.13461.

Wolfgang Lezius, Hannes Biesinger, and Ciprian-Virgil Gerstenberger. Tigersearch manual, 02 2002.

Johnny Lin. Neuronpedia: Interactive reference and tooling for analyzing neural networks, 2023. URL Mhttps://www.neuronpedia.org. Software available from neuronpedia.org.

Yongjie Lin, Yi Chern Tan, and Robert Frank. Open sesame: Getting inside BERT's linguistic knowledge. In Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes, editors, *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy, August 2019. Association for Computational Linguistics. doi:10.18653/v1/W19-4825. URL Mhttps://aclanthology.org/W19-4825/.

LiquidAI. Liquidai/lfm2-350m. Mhttps://huggingface.co/LiquidAI/LFM2-350M. Accessed: 06-10-2025.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL Mhttps://arxiv.org/abs/1907.11692.

Irina Lobzhanidze, Erekle Magradze, Svetlana Berikashvili, Anzor Gozalishvili, and Tamar Jalaghonia. Building a Universal Dependencies treebank for Georgian. In Daniel Dakota, Sarah Jablotschkin, Sandra Kübler, and Heike Zinsmeister, editors, *Proceedings of the 22nd Workshop on Treebanks and Linguistic Theories (TLT 2024)*, pages 40–45, Hamburg,Germany, December 2024. Association for Computational Linguistics. URL Mhttps://aclanthology.org/2024.tlt-1.5/.

Sanae Lotfi, Marc Finzi, Yilun Kuang, Tim G. J. Rudner, Micah Goldblum, and Andrew Gordon Wilson. Non-vacuous generalization bounds for large language models, 2024. URL Mhttps://arxiv.org/abs/2312.17173.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL Mhttps://aclanthology.org/J93-2004/.

Rebecca Marvin and Tal Linzen. Targeted syntactic evaluation of language models. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi:10.18653/v1/D18-1151. URL Mhttps://aclanthology.org/D18-1151/.

Max Planck Institute for Evolutionary Anthropology. The leipzig glossing rules: Conventions for interlinear morpheme-by-morpheme glosses. Max Planck Institute for Evolutionary Anthropology Website, 2015.

Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943. doi:10.1007/BF02478259. URL Mhttps://doi.org/10.1007/BF02478259.

Amir More, Özlem Çetinoğlu, Çağrı Çöltekin, Nizar Habash, Benoît Sagot, Djamé Seddah, Dima Taji, and Reut Tsarfaty. CoNLL-UL: Universal morphological lattices for Universal Dependency parsing. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL Mhttps://aclanthology.org/L18-1608/.

Robert Mroczkowski, Janusz Tracz, and Piotr Rybak Ireneusz Gawlik. Evaluation of bert-based models for the polish language understanding. *ML in PL Conference*, 2019.

Robert Mroczkowski, Piotr Rybak, Alina Wróblewska, and Ireneusz Gawlik. HerBERT: Efficiently pretrained transformer-based language model for Polish. In Bogdan Babych, Olga Kanishcheva, Preslav Nakov, Jakub Piskorski, Lidia Pivovarova, Vasyl Starko, Josef Steinberger, Roman Yangarber, Michał Marcińczuk, Senja Pollak, Pavel Přibáň, and Marko Robnik-Šikonja, editors, *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 1–10, Kiyv, Ukraine, April 2021. Association for Computational Linguistics. URL `Mhttps://aclanthology.org/2021.bsnlp-1.1/`.

Aaron Mueller, Yu Xia, and Tal Linzen. Causal analysis of syntactic agreement neurons in multilingual language models. In Antske Fokkens and Vivek Srikumar, editors, *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 95–109, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. doi:10.18653/v1/2022.conll-1.8. URL `Mhttps://aclanthology.org/2022.conll-1.8/`.

Léa Nash. *The Structural Source of Split Ergativity and Ergative Case in Georgian*, pages 175–200. Oxford, 08 2017. ISBN 0198739370. doi:10.1093/oxfordhb/9780198739371.013.8.

Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. Refining targeted syntactic evaluation of language models. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3710–3723, Online, June 2021. Association for Computational Linguistics. doi:10.18653/v1/2021.naacl-main.290. URL `Mhttps://aclanthology.org/2021.naacl-main.290/`.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. Universal Dependencies v2: An evergrowing multilingual treebank collection. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL `Mhttps://aclanthology.org/2020.lrec-1.497/`.

Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL `Mhttps://aclanthology.org/L12-1115/`.

Jan Pfister and Andreas Hotho. SuperGLEBer: German language understanding evaluation benchmark. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceed-*

ings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 7904–7923, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.naacl-long.438. URL `Mhttps://aclanthology.org/2024.naacl-long.438/`.

Jan Pfister, Julia Wunderle, and Andreas Hotho. Llämmlein: Transparent, compact and competitive german-only language models from scratch, 2025. URL `Mhttps://arxiv.org/abs/2411.11171`.

Buu Phan, Marton Havasi, Matthew Muckley, and Karen Ullrich. Understanding and mitigating tokenization bias in language models, 2024. URL `Mhttps://arxiv.org/abs/2406.16829`.

Carl Pollard, Ivan Sag, Edited Goldsmith, James D, Jerrold Sadock, John Nerbonne, Klaus Netter, and Jean-Pierre Koenig. Head-driven phrase structure grammar, 07 2002.

Adam Przepiórkowski, Rafał L. Górski, Marek Łaziński, and Piotr Pęzik. Recent developments in the National Corpus of Polish. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA). URL `Mhttps://aclanthology.org/L10-1097/`.

Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training, 2018. URL `Mhttps://api.semanticscholar.org/CorpusID:49313245`.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL `Mhttps://arxiv.org/abs/1910.10683`.

Keith Rayner and Charles Jr. Clifton. Language processing in reading and speech perception is fast and incremental: Implications for event-related potential research. *Biological Psychology*, 80(1):4–9, 2009. doi:10.1016/j.biopsycho.2008.05.002.

Georg Rehm and Andy Way. *European language equality: A strategic agenda for digital language equality*. Springer Nature, 2023.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. doi:10.1037/h0042519. URL `Mhttps://doi.org/10.1037/h0042519`.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. doi:10.1038/323533a0. URL `Mhttps://doi.org/10.1038/323533a0`.

Piotr Rybak, Robert Mroczkowski, Janusz Tracz, and Ireneusz Gawlik. KLEJ: comprehensive benchmark for polish language understanding. *CoRR*, abs/2005.00630, 2020. URL `Mhttps://arxiv.org/abs/2005.00630`.

Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152, 2012. doi:10.1109/ICASSP.2012.6289079.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi:10.18653/v1/P16-1162. URL `Mhttps://aclanthology.org/P16-1162/`.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension, 2018. URL `Mhttps://arxiv.org/abs/1611.01603`.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. doi:10.1002/j.1538-7305.1948.tb01338.x. URL `Mhttps://doi.org/10.1002/j.1538-7305.1948.tb01338.x`.

Taiga Someya and Yohei Oseki. JBLiMP: Japanese benchmark of linguistic minimal pairs. In Andreas Vlachos and Isabelle Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1581–1594, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi:10.18653/v1/2023.findings-eacl.117. URL `Mhttps://aclanthology.org/2023.findings-eacl.117/`.

Taiga Someya, Ryo Yoshida, and Yohei Oseki. Targeted syntactic evaluation on the Chomsky hierarchy. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15595–15605, Torino, Italia, May 2024. ELRA and ICCL. URL `Mhttps://aclanthology.org/2024.lrec-main.1356/`.

Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, and Mohit Iyyer. Sling: Sino linguistic evaluation of large language models, 2022. URL `Mhttps://arxiv.org/abs/2210.11689`.

Bayerische Staatsbibliothek. dbmdz/german-gpt2-faust. `Mhttps://huggingface.co/dbmdz/german-gpt2-faust`. Accessed: 06-10-2025.

Michelle Suijkerbuijk, Zoë Prins, Marianne de Heer Kloots, Willem Zuidema, and Stefan L. Frank. Blimp-nl: A corpus of dutch minimal pairs and acceptability judgments for language model evaluation, 05 2025. ISSN 0891-2017. URL `Mhttps://doi.org/10.1162/coli_a_00559`.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014. URL `Mhttps://arxiv.org/abs/1409.3215`.

Wojciech Szmyd, Alicja Kotyla, Michał Zobniów, Piotr Falkiewicz, Jakub Bartczuk, and Artur Zygadło. TrelBERT: A pre-trained encoder for Polish Twitter. In *Proceedings of the 9th Workshop on Slavic Natural Language Processing 2023 (SlavicNLP 2023)*, pages 17–24, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. URL `Mhttps://aclanthology.org/2023.bsnlp-1.3`.

Ekaterina Taktasheva, Maxim Bazhukov, Kirill Koncha, Alena Fenogenova, Ekaterina Artemova, and Vladislav Mikhailov. Rublimp: Russian benchmark of linguistic minimal pairs, 2024a. URL `Mhttps://arxiv.org/abs/2406.19232`.

Ekaterina Taktasheva, Maxim Bazhukov, Kirill Koncha, Alena Fenogenova, Ekaterina Artemova, and Vladislav Mikhailov. RuBLiMP: Russian benchmark of linguistic minimal pairs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9268–9299. Association for Computational Linguistics, 2024b. doi:10.18653/v1/2024.emnlp-main.522. URL `Mhttps://aclanthology.org/2024.emnlp-main.522`.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL `Mhttps://qwenlm.github.io/blog/qwen2.5/`.

Gabriele N. Tornetta. Entropy methods for the confidence assessment of probabilistic classification models, 2021. URL `Mhttps://arxiv.org/abs/2103.15157`.

Naoya Ueda, Masato Mita, Teruaki Oka, and Mamoru Komachi. Token-length bias in minimal-pair paradigm datasets. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16224–16236, Torino, Italia, May 2024. ELRA and ICCL. URL `Mhttps://aclanthology.org/2024.lrec-main.1410/`.

Antony Unwin and Kim Kleinman. The iris data set: In search of the source of virginica. *Significance*, 18, 2021. URL `Mhttps://api.semanticscholar.org/CorpusID:244763032`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL `Mhttps://arxiv.org/abs/1706.03762`.

Elena Volodina, Yousuf Ali Mohammed, and Julia Klezl. DaLAJ – a dataset for linguistic acceptability judgments for Swedish. In David Alfter, Elena Volodina, Ildikó Pilan, Johannes Graën, and Lars Borin, editors, *Proceedings of the 10th Workshop on NLP for Computer Assisted Language Learning*, pages 28–37, Online, May 2021. LiU Electronic Press. URL `Mhttps://aclanthology.org/2021.nlp4call-1.3/`.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019. URL Mhttps://arxiv.org/abs/1804.07461.

Yiwen Wang, Jennifer Hu, Roger Levy, and Peng Qian. Controlled evaluation of grammatical knowledge in Mandarin Chinese language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5604–5620, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi:10.18653/v1/2021.emnlp-main.454. URL Mhttps://aclanthology.org/2021.emnlp-main.454/.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The benchmark of linguistic minimal pairs for English, 2020a. URL Mhttps://aclanthology.org/2020.tacl-1.25/.

Alex Warstadt, Yian Zhang, Haau-Sing Li, Haokun Liu, and Samuel R. Bowman. Learning which features matter: Roberta acquires a preference for linguistic generalizations (eventually), 2020b. URL Mhttps://arxiv.org/abs/2010.05358.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. Blimp: The benchmark of linguistic minimal pairs for english, 2023. URL Mhttps://arxiv.org/abs/1912.00582.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks, 2015. URL Mhttps://arxiv.org/abs/1410.3916.

Martin Weyssow, Houari Sahraoui, and Eugene Syriani. Recommending metamodel concepts during modeling activities with pre-trained language models. *Software and Systems Modeling*, 21, 06 2022. doi:10.1007/s10270-022-00975-5.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020. URL Mhttps://arxiv.org/abs/1910.03771.

BigScience Workshop. Bloom: A 176b-parameter open-access multilingual language model, 2023. URL Mhttps://arxiv.org/abs/2211.05100.

Alina Wróblewska. Extended and enhanced Polish dependency bank in Universal Dependencies format. In Marie-Catherine de Marneffe, Teresa Lynn, and Sebastian Schuster, editors, *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 173–182, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi:10.18653/v1/W18-6020. URL Mhttps://aclanthology.org/W18-6020/.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016. URL Mhttps://arxiv.org/abs/1609.08144.

Tianyang Xu, Tatsuki Kuribayashi, Yohei Oseki, Ryan Cotterell, and Alex Warstadt. Can language models learn typologically implausible languages?, 2025a. URL Mhttps://arxiv.org/abs/2502.12317.

Xingcheng Xu, Zibo Zhao, Haipeng Zhang, and Yanqing Yang. Principled understanding of generalization for generative transformer models in arithmetic reasoning tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 4721–4747. Association for Computational Linguistics, 2025b. doi:10.18653/v1/2025.acl-long.235. URL Mhttp://dx.doi.org/10.18653/v1/2025.acl-long.235.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024a.

Jin Yang, Zhiqiang Wang, Yanbin Lin, and Zunduo Zhao. Problematic tokens: Tokenizer bias in large language models. In *2024 IEEE International Conference on Big Data (BigData)*, pages 6387–6393, 2024b. doi:10.1109/BigData62323.2024.10825615.

Xiulin Yang, Tatsuya Aoyama, Yuekun Yao, and Ethan Wilcox. Anything goes? a crosslinguistic study of (im)possible language learning in lms, 2025a. URL Mhttps://arxiv.org/abs/2502.18795.

Xiulin Yang, Zhuoxuan Ju, Lanni Bu, Zoey Liu, and Nathan Schneider. UD-English-CHILDES: A collected resource of gold and silver Universal Dependencies trees for child language interactions. In Gosse Bouma and Çağrı Çöltekin, editors, *Proceedings of the Eighth Workshop on Universal Dependencies (UDW, SyntaxFest 2025)*, pages 52–58, Ljubljana, Slovenia, August 2025b. Association for Computational Linguistics. ISBN 979-8-89176-292-3. URL Mhttps://aclanthology.org/2025.udw-1.6/.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics. doi:10.18653/v1/N16-1174. URL `Mhttps://aclanthology.org/N16-1174/`.

Karolina Zaczynska, Nils Feldhus, Robert Schwarzenberg, Aleksandra Gabryszak, and Sebastian Möller. Evaluating german transformer language models with syntactic agreement tests, 2020a. URL `Mhttps://arxiv.org/abs/2007.03765`.

Karolina Zaczynska, Nils Feldhus, Robert Schwarzenberg, Aleksandra Gabryszak, and Sebastian Möller. Evaluating german transformer language models with syntactic agreement tests, 2020b. URL `Mhttps://arxiv.org/abs/2007.03765`.

Daniel Zeman. Reusable tagset conversion using tagset drivers. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, pages 213–218, 2008.

Ziyin Zhang, Yikang Liu, Weifang Huang, Junyu Mao, Rui Wang, and Hai Hu. MELA: Multilingual evaluation of linguistic acceptability. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2658–2674, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.acl-long.146. URL `Mhttps://aclanthology.org/2024.acl-long.146/`.

Guangxiang Zhao, Saier Hu, Xiaoqi Jian, Jinzhu Wu, Yuhan Wu, Change Jia, Lin Sun, and Xiangzheng Zhang. Large language models badly generalize across option length, problem types, and irrelevant noun replacements, 2025. URL `Mhttps://arxiv.org/abs/2502.12459`.

Imry Ziv, Nur Lan, Emmanuel Chemla, and Roni Katzir. Large language models as proxies for theories of human linguistic cognition, 2025. URL `Mhttps://arxiv.org/abs/2502.07687`.

APPENDIX A

GREW QUERIES FOR EXPERIMENTS

This appendix provides all `Grew` queries passed to the `GrewTSE` pipeline for Experiments 1-3. A useful resource for quickly testing them is *https://universal.grew.fr/*, a tool which has been enormously helpful in carrying out this research.

```
V [upos="VERB"];
N [upos="NOUN", Case="Gen"];
NEG [upos="PART"];
V -[obj]-> N;
V -[advmod:neg]-> NEG;
```

Figure A.1: Query to identify a Genitive of Negation construction in the Polish PUD treebank.

```
pattern {
  V [upos="VERB"];
  SUBJ [Case="Nom"];
  OBJ [Case="Gen"];
  NEG [upos="PART", Polarity="Neg"];

  V -[nsubj]-> subj;
  V -[obj]-> obj;
}

without {
  V -[advmod:neg]-> NEG
}
```

Figure A.2: Query to identify a Verbal Genitive construction in the Polish PUD treebank.

```
pattern {
  P [upos="ADP"];
  N [Case="Gen"];
  P -[fixed]-> N;
}
```

Figure A.3: Query to identify a Prepositional Genitive construction in the Polish PUD treebank.

```
pattern {
    AUX [upos="AUX", Number="Sing", Person="3", form="könnte"];
    VERB [upos="VERB"];
    VERB -[aux]-> AUX;
    VERB -[nsubj]-> SUBJ;
    SUBJ << AUX;
    AUX << VERB;
}

without {
 SUBJ [upos=PRON,Gender=Fem];
}
```

Figure A.4: Query to identify SVO constructions with a 3rd-person singular conditional auxiliary 'könnte' in the German HDT UD treebank. Personal pronouns with feminine agreement were excluded because their forms are homonymous with the 3rd-person plural, introducing noise into the analysis.

```
pattern {
    AUX [upos="AUX", Number="Plur", Person="3", form="könnten"];
    VERB [upos=VERB];
    VERB -[aux]-> AUX;
    VERB -[nsubj]-> SUBJ;
    SUBJ << VERB;
    AUX >> VERB;
}

without {
 SUBJ [upos=PRON,Person=3,Number=Plur];
}
```

Figure A.5: Query to identify SOV constructions with a 3rd-person singular conditional auxiliary 'könnte' in the German HDT UD treebank. Personal pronouns with plural agreement were excluded because their forms are homonymous with the female personal pronoun, introducing noise into the analysis.

```
pattern {
  V [upos="VERB"];
  SUBJ [Case="Nom"];
  OBJ [Case="Dat"];
  V -[nsubj]-> SUBJ;
  V -[obj]-> OBJ;
}
```

Figure A.6: Query to identify transitive constructions with a nominative subject and dative object in the Georgian GLC UD treebank.

```
pattern {
  V [upos="VERB"];
  SUBJ [Case="Erg"];
  OBJ [Case="Nom"];
  V -[nsubj]-> SUBJ;
  V -[obj]-> OBJ;
}
```

Figure A.7: Query to identify transitive constructions with an ergative subject and nominative object in the Georgian GLC UD treebank.

```
pattern {
  V [upos="VERB"];
  SUBJ [Case="Dat"];
  OBJ [Case="Nom"];
  V -[nsubj]-> SUBJ;
  V -[obj]-> OBJ;
}
```

Figure A.8: Query to identify transitive constructions with a dative subject and nominative object in the Georgian GLC UD treebank.

APPENDIX B

FULL RESULTS FOR EXPERIMENTS

This appendix lists the full set of results from Experiments 1-3. This gives a more complete picture of the results of the three experiments and the best result in each column is highlighted in boldface. In all experiments we calculated *accuracy*, as this is the standard metric used in TSE research. We also include some results for the average surprisal difference in cases where they were referenced in the experiment.

GenOfNeg	In Distribution		Out-of-Distribution	
Model	Accuracy	ASD	Accuracy	ASD
PolBERT	**93.2%**	7.70	**89.4%**	5.90
SlavicBERT	77.4%	4.07	76.6%	2.53
Multilingual BERT	66.0%	3.85	63.4%	2.07
XLM-RoBERTa (large)	80.8%	6.08	77.0%	4.20
DistilBERT-PL	59.2%	2.24	63.4%	1.47
Polish RoBERTa (base)	89.4%	**7.75**	87.5%	**6.03**
Polish RoBERTa (large)	61.9%	-0.65	50.2%	-2.56
HerBERT (base)	90.2%	7.22	87.5%	5.36
HerBERT (large)	57.4%	0.52	57.7%	0.68
TrelBERT	84.9%	6.04	80.4%	4.56

PrepGen	In Distribution		Out-of-Distribution	
Model	Accuracy	ASD	Accuracy	ASD
PolBERT	**99.1%**	18.31	**91.7%**	6.35
SlavicBERT	98.2%	17.78	80.7%	4.56
Multilingual BERT	95.4%	**27.98**	85.3%	6.47
XLM-RoBERTa (large)	98.2%	24.47	85.3%	**6.81**
DistilBERT-PL	92.7%	24.62	84.4%	5.50
Polish RoBERTa (base)	98.2%	18.86	85.3%	5.88
Polish RoBERTa (large)	97.2%	10.61	76.1%	4.02
HerBERT (base)	98.2%	16.42	83.5%	5.70
HerBERT (large)	89.0%	5.15	60.6%	1.26
TrelBERT	97.2%	15.34	82.6%	4.89

VerbGen	In Distribution		Out-of-Distribution	
Model	Accuracy	ASD	Accuracy	ASD
PolBERT	**91.3%**	7.07	81.0%	4.09
SlavicBERT	66.7%	2.61	61.3%	1.08
Multilingual BERT	62.7%	3.50	53.7%	0.75
XLM-RoBERTa (large)	76.0%	5.03	61.0%	1.96
DistilBERT-PL	68.0%	2.54	53.3%	0.71
Polish RoBERTa (base)	89.0%	**7.98**	80.0%	4.40
Polish RoBERTa (large)	65.0%	-0.20	46.3%	-4.21
HerBERT (base)	89.7%	7.61	**85.0%**	**4.60**
HerBERT (large)	54.3%	0.85	53.7%	0.09
TrelBERT	84.3%	6.00	77.0%	3.56

Table B.1: Full accuracy (%) and ASD results for Experiment 1 on the Polish genitive case.

SVO Conditional Auxiliary	Singular		Plural	
Model	Accuracy	ASD	Accuracy	ASD
LiquidAI LFM2-350M	83%	1.764	54.3%	0.723
LLaMmlein 120M	**98.1%**	3.726	**100%**	3.482
Faust GPT-2	94.3%	4.235	87%	**4.203**
Bloom 560M	94.3%	3.542	10.9%	-2.339
Qwen 2.5 0.5B	96.2%	**5.73**	89.1%	1.775

SOV Conditional Auxiliary	Singular		Plural	
Model	Accuracy	ASD	Accuracy	ASD
LiquidAI LFM2-350M	81.5%	2.233	83.3%	2.711
LLaMmlein 120M	**100%**	7.848	**93.8%**	**5.613**
Faust GPT-2	**100%**	**12.764**	**93.8%**	5.385
Bloom 560M	94.4%	2.695	52.1%	0.012
Qwen 2.5 0.5B	96.3%	7.011	91.7%	3.564

Table B.2: Full accuracy (%) and ASD results for Experiment 2 on the German 3rd-person conditional auxiliary ‘könnte’.

DAT-NOM	ID				OOD			
	Subject		Object		Subject		Object	
Model	→ERG	→NOM	→DAT	→ERG	→ERG	→NOM	→DAT	→ERG
XLM-RoB-ls	86.2%	**78%**	93.3%	**100%**	75.9%	42.9%	**93.3%**	**100%**
XLM-RoB	82.8%	75.8%	95.6%	**100%**	79.2%	42.9%	84.4%	**100%**
MBERT	93.1%	47.3%	**97.8%**	**100%**	72.4%	27.5%	91.1%	83.3%
HPLT-ka	**96.6%**	59.3%	**97.8%**	**100%**	**89.7%**	**59.3%**	88.9%	**100%**

ERG-NOM	ID				OOD			
	Subject		Object		Subject		Object	
Model	→DAT	→NOM	→DAT	→ERG	→DAT	→NOM	→DAT	→ERG
XLM-RoB-ls	76.6%	58.5%	95.7%	92.9%	44.7%	34.2%	97.9%	**100%**
XLM-RoB	76.6%	51.2%	93.6%	85.7%	38.3%	31.7%	89.4%	**100%**
MBERT	51.1%	22%	**97.9%**	**100%**	38.3%	12.2%	**100%**	**100%**
HPLT-ka	**83%**	**61%**	**97.9%**	**100%**	**68.1%**	**41.5%**	97.9%	**100%**

NOM-DAT	ID				OOD			
	Subject		Object		Subject		Object	
Model	→ERG	→DAT	→ERG	→NOM	→ERG	→DAT	→ERG	→NOM
XLM-RoB-lg	87.4%	78.7%	98.2%	**76.1%**	**92.3%**	97.8%	**100%**	53.5%
XLM-RoB	95.5%	92.1%	98.2%	73%	89.2%	**98.9%**	98.2%	50.4%
MBERT	94.6%	97.8%	**100%**	52.7%	91.9%	**98.9%**	96.4%	42%
HPLT-ka	**96.9%**	**98.9%**	**100%**	75.7%	90.1%	**98.9%**	**100%**	**67.3%**

Table B.3: Accuracy (%) results for Experiment 3 on the Georgian case system for subject-object transitive verbs. Each →X represents how the grammatical syntactic feature is adjusted to become ungrammatical. Note that the dative-nominative, ergative-nominative, and nominative-dative does not imply any specific word order but rather the subject and object.