What do you know, BERT ? Exploring the linguistic competencies of Transformer-based contextual word embeddings

Eleni Metheniti

Université Fédérale
Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE
Délivré par l'Université Toulouse 2 - Jean Jaurès

Présentée et soutenue par

Eleni METHENITI

Le 28 juin 2023

Qu'est-ce que tu sais, BERT? Explorer les compétences linguistiques des plongements lexicaux contextuels basés sur Transformers

Ecole doctorale : **CLESCO - Comportement, Langage, Education, Socialisation, Cognition**

Spécialité : **Sciences du langage**

Unité de recherche :
CLLE - Unité Cognition, Langues, Langage, Ergonomie

Thèse dirigée par
Nabil HATHOUT et Tim VAN DE CRUYS

Jury

Mme Marie CANDITO, Rapporteure
Mme Lonneke VAN DER PLAS, Rapporteure
M. Olivier FERRET, Examinateur
M. Nabil HATHOUT, Co-directeur de thèse
M. Tim VAN DE CRUYS, Co-directeur de thèse
Mme Cécile FABRE, Présidente

Acknowledgements

As one long prepared, and graced with courage,
as is right for you who proved worthy of this kind of city,
go firmly to the window
and listen with deep emotion, but not
with the whining, the pleas of a coward;
listen—your final delectation—to the voices,
to the exquisite music of that strange procession,
and say goodbye to her, to the Alexandria you are losing.

—**The God Abandons Antony** (1911), C.P. Cavafy (1863-1933)

*

* *

I would like to express my sincerest gratitude to my advisor, Nabil Hathout, for his steady guidance, continuous support, and hard work throughout my doctoral work. I would also like to deeply thank my advisor, Tim Van de Cruys, for his helpful guidance and essential ideas and insights for my research.

I am thankful to my colleagues at CLLE in the University of Toulouse 2, for their support and the friendly and uplifting environment, when the pandemic didn't keep us apart. Thank you to Filip, Julie, Marine, Daniele, Yizhe, and Sylvia, to mention only a few of those who lent a friendly ear to my worries.

I could not forget my colleagues at IRIT either, former and present coworkers, for their immense help and trust during not only during my thesis but also during the transitional period between my thesis and my postdoc. To Chloé Braud and Philippe Muller, I will always be grateful for your faith in me. Thank you to Laura, Kate, Nicolas, Fanny, and Fu-Hsuan, my musical friend.

Thank you to everyone in my alma mater, Saarland University, professors, coworkers and friends, for all the knowledge, support and opportunities to study and work on my passion.

I would not have made it this far without my parents, who have been enduring the sorrow of separation for too long, yet they still stand by my side.

Thank you to my childhood friend, Emma, for the decades of love and companionship. Thank you to Stefan for being a loyal friend and a provider of Brezeln.

Thank you, my dear Karel, for your love and for being by my side through thick and thin—especially during the hard times.

Thanks to cleopatrick and The Blue Stones for providing the soundtrack to long days and nights of work.

Finally, thank you to everyone who has shown me kindness. Even though I have never relied on it, it's a gift I rarely offer myself. And to everyone that I have forgotten to mention in these acknowledgments, inadvertently or intentionally, you were all part of the road that led to this work.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Natural Language Processing (NLP) has traditionally concentrated on defining and designing systems for the treatment, understanding, and production of language, with the motivation that success on these tasks would result in competent language systems for downstream applications. NLP applications include classification tasks on a sentence or document level (e.g. sentiment classification), sequence labeling tasks on a word or phrase level (e.g. syntactic parsing, named entity recognition), span relation classification, and generation tasks, which involve creating text output based on a given input (e.g. machine translation, dialogue generation, speech production). These task-specific algorithmic architectures could be combined with other models to execute complex tasks and could be themselves composed of different models, e.g. tokenizers and part-of-speech taggers. Originally built with hand-written rules by linguists, nowadays the use of advanced statistical methods of logistic regression and neural network models has become the norm in most applications of NLP.

In recent years, there have been monumental developments in the implementation of neural network architectures and giant leaps in their abilities to process, comprehend and produce language. These advancements and novel architectures have been following trends and developments in other fields of Computer Science and Machine Learning, most notably Computer Vision, i.e. the process of interpretation of visual information by a computer in order to acquire the same information that the human visual system can understand. The use of attention mechanisms in NLP became widespread after the works

of Bahdanau et al. (2014) and Luong et al. (2015) on neural machine translation, where the alignments between the source and target tokens were effectuated by an *attention* algorithm; soon after, attention mechanisms became commonly used in the architectures of Recurrent Neural Networks, for NLP applications beyond the scope of machine translation. Vaswani et al. (2017) introduced a new method of attention called *self-attention*, which is built in the architecture of a neural network called a *Transformer*. This attention mechanism is able to generate the attention weights of each token by observing its different hidden states in the sequence. It captures multiple representations with regard to the other tokens, with the use of multiple heads of self-attention.

These developments have allowed for the creation of Transformer architectures for NLP that use dynamic, contextualized word embeddings and are able to learn multiple tasks (even in parallel) through sequence learning. These architectures are trained with huge datasets on multiple computing units with massive processing power and produce language representations in the form of models of contextualized word embeddings. These models are capable of performing multiple NLP tasks with achieving state-of-the-art results, and can also be adapted to specific tasks (with the use of smaller datasets and common computers) in order to be even more competent on a particular function[1]. These features of *generality* and *adaptability* alongside their excellent performance on accuracy metrics have rendered them a staple in multiple NLP applications, especially ones that relied on the combination of multiple tasks (e.g. conversational agents). Models such as BERT (Devlin et al., 2019) have demonstrated an astonishing breadth of language skills and flexibility to a wide variety of linguistic circumstances, and recent NLP research has focused on their multiple applications and on analyzing their successes and weaknesses.

The ability of these systems to achieve human levels of performance on various NLP tasks is fascinating, but there are substantial differences between the way humans learn and develop language and how these models are trained to complete specific tasks. It

[1]Since their introduction, the NLP community has erroneously referred to Transformer architectures as *language models*; however, a more accurate description of BERT and BERT-like architectures would be a *machine learning framework* for NLP that can produce pretrained embeddings. Unlike traditional language models, these models do not contain actual probabilities, but the representations of words in different contexts, as learned by the Transformer architecture. Predicting the probability of a word is the objective of the pretraining process of most of those architectures, but the final output is discarded and the hidden state of the target word is kept.

is essential to comprehend the significance of this gap between machine and human language learning; humans are capable of learning semantic concepts and expressing them with the appropriate syntactic patterns, while research supports that Transformer models learn frequent artifacts from their vast corpora, some rudimentary idiosyncratic patterns of syntax, but no notions of semantics. Even when a model successfully generates a linguistic construction once, there is no guarantee that subsequent instances of that construction will be similar or consistent, especially following a shift in the subject matter's domain. Without reverting to systems that rely too heavily on strict linguistic norms, there needs to be a better understanding and an improved learning process of syntax and semantics.

Language acquisition is an inherent human ability that develops rapidly yet remains a lifelong process, as a human learns new terms and concepts, may acquire multiple languages, and evolves in a dynamic society. Language models are bound to be language representations "frozen in time" of the period when their corpora were created, and retraining them with new data is a costly and lengthy process. On the issue of corpora, these models require a vast amount of data in order to learn meaningful connections and representations; BERT was trained on 3.3 billion words, far more than a human will encounter and utter in a lifetime. Finally, one trade-off that has to be made in order to acquire these large datasets is the inability to fully take account of their content, thus resulting in fiction being interpreted as fact by the model (whereas for humans the distinction would be evident), in addition to the presence of inductive biases and the models learning and using them.

A few examples of pretrained models producing problematic outputs can be seen in Table 1.1; the task was to predict the five most likely words to fill the blank in short sentences. The models predicted words without distinguishing among parts of speech, sometimes even making syntactic mistakes (e.g. predicting *le* "the" at the end of the sentence). They relied on frequent co-occurrences ("drink"-"beer") without being mindful of semantic constraints in the context ("The boy", "breakfast"). They reproduced some learned artifacts regardless of their fit (e.g. "evalle" in ALBERT), and they also occasionally used offensive terms (the RoBERTa model). There are evident weaknesses in the capacity of these models to understand the content of the words and sequences

they have learned, and this directly affects how they produce language output. These mistakes should not be overlooked as a sacrifice to overall high accuracy. Reproducing harmful biases that target social and ethnic groups is an alarming problem for the NLP community, and proof that the computational power of Transformers does not come close to the sophistication of human cognition, regardless of reports of stellar accuracy and positive press.

Sentence	bert-base-uncased	roberta-base	albert-base-v2
My dog is [MASK].	dead, here, fine, gone, missing	dead, *, *, sick, *	evalle, joyah, lucivar, jaenelle, adorable
The boy drank [MASK].	deeply, slowly, again, heavily, it	beer, more, heavily, milk, water	evalle, joyah, whisky, vodka, whiskey
The boy drank [MASK] for breakfast.	water, milk, it, coffee, beer	milk, water, coffee, tea, juice	vodka, brandy, whiskey, beans, evalle
We ate [MASK] for lunch.	together, lunch, dinner, it, pizza	pizza, sushi, tacos, sandwiches, pasta	joyah, sandwiches, pizza, cookies, fries
I wore my [MASK] shoes and went running.	running, tennis, own, new, gym	running, tennis, normal, gym, hiking	jogging, tennis, hiking, tread, athletic

Sentence (French)	*Translation*	camembert-base	*Translation*	flaubert-base-uncased	*Translation*
Mon chien est [MASK].	"My dog is [MASK]."	malade, décédé, mort, heureux, diabétique	sick, deceased, dead, happy, diabetic	le, tout, aussi, un, à	the, all, also, one, at
Le garçon a bu du [MASK] au petit déjeuner.	"The boy drank [MASK] for breakfast."	lait, café, thé, vin, champagne	milk, coffee, tea, wine, champagne	poulet, lait, riz, vin, porc	chicken, milk, rice, wine, pork

Table 1.1: The top 5 predictions of three English language models (BERT, RoBERTa, ALBERT) and two French models (CamemBERT, FlauBERT) when asked to fill in the blank in a few sentences. With asterisk (*) are marked ableist slurs that the authors refuse to reproduce.

1.2 Research objectives

A Transformer architecture's underlying structure is built on parallel operations, multiple layers, and multi-headed self-attention. Multi-headed attention in a multi-layer model means that every head, in every layer, computes its own weights and attends to the architecture's encoded input separately. This powerful mechanism has been credited with the outstanding performance of Transformer models but is also difficult to examine and comprehend with traditional NLP methods. There has been a great deal of study and discussion to determine whether these self-attention processes are interpretable, i.e. whether they yield results—correct or incorrect—that can be linked to the way they respond to the input.

The objective of this doctoral thesis is to study the linguistic abilities (if any) and the limitations of Transformer-based contextual word embeddings, with experiments on complex syntactic-semantic phenomena. The main hypothesis of this thesis is the following:

Can contextual word embeddings capture enough information, through pretraining and finetuning, to be competent in complex linguistic tasks? Are their successes due to a true understanding of token relations and hierarchies or a shallow repetition of patterns in the training set? Are their failures serious, and are they systematic weaknesses or random occurrences?

We selected linguistic features and phenomena that are easily perceived by a native speaker with mature syntactic-semantic competencies but have been traditionally hard to define with linguistic rules. Specifically, we are focusing on:

- **Selectional preferences**, i.e. the arguments and classes of arguments that best complement the meaning of the verb, resulting in grammatical and semantically acceptable (felicitous) sentences.
- **Lexical aspect**, i.e. is a set of features that determine a verb's temporal qualities regardless of grammatical features such as tense.
- **Word order** of epithet adjectives in French, a seemingly simple task but complex at times, due to adjective mobility based on linguistic, non-linguistic, and semantic factors.

Overall, we lay out the specific questions that will be addressed by the approaches proposed in this thesis:

- **Do contextual word embeddings capture context sufficiently and effectively?** This question is an observation on pretrained word embeddings. During the pretraining process, are the embeddings able to generalize and group contexts in classes—not as in linguistic classes, but as clusters of linguistic-semantic similarity that can be accessed by the model to make better predictions?

- **Do contextual word embeddings show sensitivity to semantic features and semantic felicity/infelicity?**
  Some phenomena such as lexical aspect are inherent properties and are not always expressed morphologically or with the help of the context. Have the models encoded enough information based on instances in pretraining, in order to successfully identify such phenomena? When faced with an infelicitous sentence, will the models reject it due to its low frequency or due to some internalization of semantics?

- **Is finetuning necessary, beneficial, and stable for challenging tasks?**
  Transfer learning is one of the most groundbreaking functionalities of transformer-based models, allowing the already powerful embeddings to become even more specialized on a task without the need for large datasets. However, there has been criticism of the stability of finetuning, moreover, we want to observe the amount of improvement that it offers.

- **What is the role of the attention mechanism in predictions**, with regard to our experimental questions? The high performance of Transformer-based models has been attributed to their multi-head self-attention architecture, a mechanism notoriously difficult to decipher. Are the choices of our models reflected in the inner workings of the layers and heads of the attention mechanism?

- **Is word order truly unimportant for transformer-based models?** The parallelization of the learning process in Transformer models means that these models do not view input sequentially. Research has shown insensitivity to word order, but are the models insensitive to it when word order is determined by the meaning of a token?

1.3 Contributions

We are hoping that our work brings practical and empirical contributions to the NLP community. Our work falls under the scope of the ongoing "BERTology" research, the unofficial moniker of the studies in comprehending and interpreting Transformer models. We aim to contribute to the ongoing discussions with our findings on the linguistic capabilities of Transformer-based models. Our motivation is based on language competencies examined through quantitative and qualitative measures, and moving beyond the scope of success on benchmarks and reported accuracies on NLP tasks. By unveiling the linguistic weaknesses of the models, we hope to dispel any claims that Transformer models truly understand and produce language in a similar way to humans, with the same syntactic and semantic capacities. However, we are also looking forward to observing the models' strengths, understanding the inner workings that make them indubitably very successful, and discovering patterns of behavior that could point to a degree of sophistication in language processing. Additionally, we would like to see if transfer learning, the method of specializing the models with explicit knowledge, can help them overcome some of their limitations.

1.4 Thesis outline

In Chapter 2, we present an overview of the technological advances that paved the way for the development of Transformer architectures. We present an overview of word embeddings, from traditional static word embeddings to Transformer-produced deep contextual word embeddings. We also discuss the development of the attention mechanism, and subsequently the self-attention mechanism and its parallel use with the Transformer architecture, to create contextual word embedding models. Finally, we present an overview of the architectures and models that will be used in our work.

In Chapter 3, we focus on the existing research on the abilities of Transformer architectures and contextual word embeddings. Our bibliography spans from 2019 to early 2022, since the rapid developments of Transformers in NLP were occurring in tandem with the work of this doctoral work and it is impossible—and not relevant—to exhaustively present all work. We propose a selected bibliography on the explainability of self-

attention, the linguistic analyses of contextual word embeddings, and the conclusions of several researchers on the linguistic abilities of the models and their embeddings.

The following chapters present our experiments with Transformer-based deep contextual word embeddings and the linguistic features that we selected to test the models on. The phenomena examined are based on syntactic and semantic competencies that develop naturally in native speakers, but could potentially be challenging for a model that has only acquired a superficial level of this information.

In Chapter 4, we are presenting the first series of experiments we conducted in English, on the abilities of contextual word embeddings to capture the selectional preferences of a verb for its arguments. We perform tests on the bert-base-uncased model (a well-studied model, at the time) in order to compare its predictions of verb dependents to the plausibility/felicity scores that speakers would assign to them.

In Chapter 5, we focus on two sets of experiments with contextual word embeddings from Transformer-based architectures (BERT, RoBERTa, XLNet, ALBERT, CamemBERT, FlauBERT), testing their abilities to classify verbal lexical aspect (telicity, duration) in a sentence. We conduct quantitative and qualitative analyses of the models' assessment of lexical aspect in English and French, and we observe the models' self-attention mechanism and surface-level linguistic preferences.

In Chapter 6, we present our experiments on the classification of adjective word order in French, with contextual word embeddings from Transformer-based architectures (CamemBERT, FlauBERT). We examine the strengths and weaknesses of the models in identifying adjective position, also in correlation to human preferences.

Finally, Chapter 7 contains our conclusion where we summarize our findings and provide our opinion on future endeavors.

1.5 Publications

The work in this dissertation principally relates to the following peer-reviewed articles (in order of publication):

- Metheniti, E., Van de Cruys, T., & Hathout, N. (2020). How Relevant Are Selectional Preferences for Transformer-based Language Models?. In *Proceedings of the 28th In-*

ternational Conference on Computational Linguistics (pp. 1266-1278). ACL: Association for Computational Linguistics.

- Metheniti, E., Van de Cruys, T., & Hathout, N. (2021). Prédire l'aspect linguistique en anglais au moyen de transformers (Classifying Linguistic Aspect in English with Transformers). In *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1: conférence principale* (pp. 209-218).

- Metheniti, E., Van De Cruys, T., & Hathout, N. (2022). About Time: Do Transformers Learn Temporal Verbal Aspect?. In *12th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2022)* (pp. 88-101).

- Metheniti, E., Van De Cruys, T., Kerkri, W., Thuilier, J. & Hathout, N. (to appear). "Chère maison" or "maison chère"? Transformer-based prediction of adjective placement in French. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Findings.*

While not directly related, the following articles have also been completed over the course of the doctoral work:

- Metheniti, E., & Neumann, G. (2020). Wikinflection corpus: A (better) multilingual, morpheme-annotated inflectional corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 3905-3912).

Chapter 2

Transformer-based contextual word embeddings

2.1 Introduction

This chapter presents the technical background of this doctoral thesis. An exhaustive overview of many decades of computer science, machine learning, and natural language processing developments that led to the release of large language models (LLMs) is impossible and not necessarily relevant to our work. LLMs have little to no linguistic motivation and share more commonalities with Computer Vision models than with statistical natural language processing methods and linguistic tools. Nevertheless, we present a brief history and features of static word embeddings, and we explain in detail the traditional attention mechanism, as a stepping stone to analyzing self-attention and Transformer architectures and models. These competencies are necessary, not only because they set the foundation for Transformer-based contextual word embeddings, but also for the experiments presented in the following chapters, as we make use of traditional methods (embedding visualizations, attention visualizations) to further study our experimental results.

Natural language processing methods commonly require text to be converted into vectors of numerical values. Encoding can be a succinct process of mapping values to a vocabulary either as indices or as optimized vectors for processing. A vector space model or **word embeddings** model is a semantic space, where lexical items (words or multi-word terms) are represented as vectors or embeddings. Vector similarities may correlate with semantic similarities, because words of the same class, function, or similar meaning

are encoded with similar vectors based on their similar occurrences in multiple contexts. This has led to the common assumption that vector space models contain may be able to capture important semantic information. The idea of language models representing semantics stems from structuralist linguistics and the philosophy of language (Harris, 1954; Firth, 1957). Initial attempts to measure semantic similarity via feature representations employed hand-crafted features (Osgood et al., 1957). Following the advancements in machine learning, statistical methods were introduced (e.g. Latent semantic analysis, Deerwester et al., 1990; Landauer et al., 1998), allowing for the extraction of distributions from large corpora (Salton et al., 1975) in an unsupervised way (Mikolov et al., 2013a; Pennington et al., 2014).

Even though one of the original motivations of word embeddings was to capture distributional information, static word embeddings focus on capturing the average meaning of a word rather than all its possible senses, its different uses, and its preferred context(s). Their structure of one embedding per word means that, unfortunately, words with ambiguous meanings or varied contexts will not be accurately represented. However, the advancements in neural network architectures, such as Transformer-based models, have allowed for more dynamic unsupervised learning of words and their context.

Following the initial approach of Bengio et al. (2000) to capture distributional information with a neural network, modern methods of creating word embeddings aim to create distributionally-informed word representations. The creation of word embeddings from neural architectures is possible due to the **embedding layer** of the model, which maps the input sequence into a series of vectors. These vectors are created by the model on a specific training task, therefore they contain all the learned information needed for said task. Deep contextualized language representations are **dynamic** because they are able to represent a word in multiple instances as a function of its context, capturing important and varied syntactic-semantic information. Additionally, the representations can be finetuned on a given task and dataset, thus becoming more specialized with precise knowledge.

While BERT was not the first architecture of deep contextualized word embeddings or based on a Transformer architecture, it radically changed the field of natural language processing, with its state-of-the-art results in multiple fields and tasks, and easy fine-

tuning process. The introduction of libraries in Python (e.g. Huggingface Transformers) that can automatically load the pretrained models and provide the code implementation for finetuning and for basic tasks (masked language modeling, next sentence prediction, binary/multiple sequence classification, token classification, question answering) has increased the popularity of BERT models even more and has solidified its presence as one of the standard practices in modern NLP. However, along with the praise has also been extensive work to analyze Transformer-based contextual word embeddings. In the following sections, a few pivotal architectures and their embeddings for English and French will be presented, focusing on the ones used in the course of this doctoral research.

2.2 Language encoding

Raw data for neural networks can be either binary (two possible values), categorical (three or more possible values), or numerical. Text may be broken into smaller units, either of linguistic significance like words, phrases, and sentences, or further broken down into characters or subwords (i.e. fragments of words) and is then encoded in vectors and passed as input (numerical data). Each unit of the original input corresponds to a real-valued vector: the most straightforward method would be with **index-based encoding**, where an index number represents every unique element and the input is a vector or a series of vectors. For example, a large text may be encoded per word; the set of words (i.e. the *vocabulary*) would be indexed, and each sentence can be a vector where each value corresponds to the index of a word. The downside of this method is that it creates vectors of large size, it cannot encode new elements which have not been seen in the original text (they are encoded *en masse* as an unknown token), and does not contain any meaningful information of the encoded element.

Another way to encode a large text per word would be with the **Bag-of-Words** algorithm; in this approach, the vocabulary does not have unique indices, but each element of the vocabulary receives a non-unique value (boolean in binary BoW, or with more values if the frequency of the element is taken into consideration). The sentence vector would be composed of these values, as seen in the example in Table 2.1. While this method allows for the encoding of large amounts of data while dealing with unknown elements, it does not encode the uniqueness of the elements, their word order, or systematically

capture linguistic information. Additionally, even with the ability to represent unknown elements, the produced vectors for a large dataset may be very large with lots of zero scores, called a *sparse vector* or sparse representation, that are more computationally expensive to process (Goldberg, 2016).

Document:	**Encoding:**	**Sentence:**
"She eats an apple."	{ she:1, eats:1, an:1,	"I like apple pie."
"We like your apple cake."	apple:2, we:1, like:2, your:2,	⇓
"They like pie."	cake:1, they:1, pie:1, .:3 }	[0, 2, 2, 1, 3]

Table 2.1: An example of Bag-of-Words encoding; how the frequency-based encodings are made from a document of sentences, and how to encode a new sentence.

In order to be able to capture semantic information about language, words (or units with semantic and linguistic importance) need to be treated as *categorical data*, i.e. labeled data. Words can be described, based on the presence or absence of a concept that is used as a label/feature, and this allows not only for a description of the meaning and function of the word, but can also create semantic links between words, e.g. synonymy, hyponymy, and hypernymy. The encoding of categorical data in a form that can be used by machine learning methods is done by encoding each element/category in its own *vector*, where every vector dimension corresponds to a feature. **One-hot encoding** maps these categories with binary values based on the presence/absence of each feature (see Figure 2.2). While this encoding scheme allows for feature labeling, the rigidity of the binary values results in sparse vectors, which, as previously discussed, are usually undesirable. Additionally, knowing merely the presence or absence of a feature is not sufficient to interpret the semantic relations, especially between synonyms and co-hyponyms. This difficulty is also reflected in the computational processing of these vectors. Other methods of encoding have been introduced to represent features in more compact ways by aggregating values for categories (e.g. target encoding), but they remain uninformed of the actual content of the category and may misinterpret its relation to other elements.

Word	Category				
	fruit	sweet	...	color	human
apple	1	1	...	0	0
orange	1	1	...	1	0
apple pie	0	1	...	0	0
...					

"I like apple pie"

⇓

```
[[0, 0, ..., 0, 1],
 [0, 0, ..., 0, 0],
 [0, 1, ..., 0, 0]]
```

Table 2.2: An example of one-hot encoding of categorical data, and how the resulting vectors can be used to encode a phrase.

2.3 Static word embeddings

Instead of relying solely on language encodings to fully capture the linguistic and semantic intricacies of language, an additional method of representing language may be used alongside them, the *word embeddings* in an *embedding layer*. A word embedding model is a set of vector representations, usually of 100-500 dimensions, where each dimension represents a learned feature of a word in a real number. These features, however, do not correspond to human-defined categories in unsupervised learning methods. The model is called static because each word is represented by one vector, its possible different meanings summed up. Static word embeddings are commonly created based on large corpora, thus they aim to represent a large vocabulary and are not specialized to a specific task. The broader their vocabulary, the better they are at representing semantic similarities and relations among words, as the model learns each word's "preferred company".

The idea of a semantic space with linguistic items (words or multi-word concepts) conveyed as vectors or embeddings may tackle the computational difficulties of computing categorical properties in the encoding stage. Indeed, Collobert et al. (2011) reported that word embeddings learned from significant amounts of unlabeled data are far more satisfactory than randomly initialized encodings, and Al-Rfou' et al. (2013) highlight their positive contribution even to tasks they were not originally created for (in this case, multilingual part-of-speech tagging). Determining the vectorial representations of words with semi- or self-supervised methods could either be performed as a function of a term's presence in a bag of documents, as a form of the *tf-idf* encoding used in Infor-

mation Retrieval (*document occurrence representation*), or as a function of similar words that appear in the same contexts (*term co-occurrence representation*).

History of distributional semantics

Distributional semantics, which focuses on understanding meaning in observed language, has employed *semantic vector space models* as a means of knowledge representation. The goal is to quantify and classify semantic similarities between linguistic items based on their distributional features in large samples of language data. This distributional approach is easy to scale on different corpora sizes, as it does not rely on specific resources, and additionally is able to model corpus-specific sense distributions. Harris (1954) supported that words with similar meanings tend to occur in the same contexts, and Firth (1957) developed the hypothesis that "a word is characterized by the company it maintains", an idea also explored by psycholinguists of the time (*semantic differential*, Osgood et al. (1957)).

The *vector space model* for information retrieval (Salton et al., 1975) is the first attempt to create semantic space models for use in computational linguistics; however, this method creates a very sparse, high dimensional vector space. To tackle the sparse vector problem, *latent semantic analysis* (Deerwester et al., 1990; Landauer et al., 1998) and the *random indexing* approach (Kanerva et al., 2000; Sahlgren, 2002) were introduced, in order to reduce the number of dimensions, using linear algebraic techniques like *singular value decomposition*. Since then, many clustering methods have been proposed in order to improve the quality and the performance of these word embeddings (Pantel and Lin, 2002). Pereira et al. (1993) created word clusters based on syntactic and co-occurrence relations, in order to reduce dimensions in a more linguistically-informed manner. The issue that persists with the distributional method, however, is the weakness of capturing salient meaning and distinguishing synonyms from antonyms, when compared with rule-based methods (Lin et al., 2003).

Vinokourov et al. (2002) introduced the concept of *multilinguality* in word representations since the same word in different languages should still occupy the same semantic space, and created word and document embeddings in a self-supervised manner (with the use of kernel Canonical Correlation Analysis). Morin and Bengio (2005) proposed a

hierarchical language model of a binary tree of words, built with neural networks alongside WordNet's human-crafted definitions and categories, and Mnih and Hinton (2008) proposed an automatic method of construction of these hierarchies.

The release of the **word2vec** algorithm has been pivotal in the development of word embeddings (Mikolov et al., 2013c,a). It consists of a two-layer neural network trained on a "fake task" to retrieve one random similar word for a word in the middle of a context window (e.g. a sentence) and uses the learned hidden weights. Its two main architectures are the *Continuous Bag-of-Words* model which predicts the target word based on the distributed representations of the context words, and the *Continuous Skip-Gram Model* which aims to learn and predict the context of the target word (McCormick, 2016). Word2Vec can provide accurate predictions about a word's meaning based on its usage and its associations in a (large) text, but more importantly, it is capable of assigning similar values to words of similar meanings and distributing words in meaningful groups. These meanings and semantic associations can be easily observed with algebraic operations, to visualize similarities and semantic clusters. In addition to the algorithm, pretrained word embeddings have been released, trained on large corpora (Google News corpus of 100 billion words, Wikipedia dumps) and with large vocabularies, ready to be used alongside NLP applications.

Soon after the release of word2vec, the **GloVe** (Global Vectors) learning algorithm was introduced (Pennington et al., 2014); it is also capable of creating word vector representations from large texts, in a distribution that reflects semantic relations. It is an unsupervised method like word2vec but uses local context information of words alongside an aggregated global word-to-word co-occurrence matrix (for example, *latent semantic analysis*) in order to come up with a principled loss function that uses both these. Thus GloVe vectors are able to examine global occurrences and co-occurrences of terms and determine the semantically-important context from regular context. The idea of using dimensionality reduction on the word co-occurrence matrix also appeared in other works of the time (Lebret and Collobert, 2014; Levy and Goldberg, 2014; Li et al., 2015), however, GloVe was also able to harness the benefits of the word2vec approach to capture synonymy. Pretrained word vector models made with the GloVe algorithm have also been made available and widely used, with large Internet-sourced corpora.

Bojanowski et al. (2017) created the **fastText** algorithm and pretrained embeddings, using the skip-gram method as word2vec and additionally by decomposing a word to *character n-gram information*, i.e. subword units. This approach aims to incorporate morphological information into vector embeddings, as the (automatically segmented) subwords may contain morpheme information with salient syntactic and semantic information –whether they coincide with morpheme boundaries or not. While this method strays away from word-level semantic information, and additionally the subword units are not linguistically informed, they have achieved similar or better performance in some NLP tasks, and demonstrated their usefulness with morphologically-rich languages, bringing up an important perspective in the usually anglocentric approach of NLP methods.

Even though the use of unsupervised methods has become the norm, especially after the widespread popularity of pretrained word embedding models, the need for explainability and salient semantic roles was not sated; Qureshi and Greene (2019) proposed a method of unsupervised word embeddings made with human-readable features, as a way to make similarity and semantic features more interpretable. However, the tides did not turn in the world of natural language processing; as machine learning methods have proved for decades, human cognition is most valuable as an inspiration and a stepping-off point, rather than a strict blueprint for algorithms.

Properties of static word embeddings

In the previous section, it was discussed how pretrained word embeddings from algorithms like word2vec are able to capture interesting semantic relations between words. This information can be exploited in order to assess a model's capacities and the contribution it has to language tasks. As previously explained, each word or subword has a corresponding vector of 100-500 dimensions.

Each of these vectors can be placed in a *continuous vector space*, thus creating spatial relations between words which can be used to compare and congregate them into *semantic spaces*. The word vectors can be easier visualized in 2- or 3-dimensional plots with dimension reduction methods such as *principal component analysis* (PCA) or *t-distributed Stochastic Neighbor Embedding* (t-SNE), and mathematical operations can be used to easier compare different vectors' magnitudes and directions.

In visualizations of vectors, similarity can be defined as the Euclidean distance between two vectors (i.e. the actual distance between points), or the cosine similarity of the vectors (the angle between two vectors in space). It is expected that similar word vectors will converge to similar locations in the word embedding space due to their similarity; in Figure 2.1, it is shown that words related to vehicles (synonyms or hyponyms) exist in a **cluster** far from unrelated words (e.g. *moon, tree*). Apart from semantic similarity, algebraic operations can also demonstrate different types of dependencies between words, on a more sophisticated level. For example, in Figure 2.2, Mikolov et al. (2013c) demonstrate that the word2vec embeddings for country names have negative values on the x-axis and the capital names have positive values. The countries have similar y-axis values to the corresponding capital, and the transformation between each country and its capital is similar for every pair, which suggests that these pairs have a similar semantic relation in the word2vec embeddings. Similar findings were reported for GloVe vectors, for example with semantic relations between non-verbal elements and words (postal codes and cities), with named entities (company names and CEOs). Relations of gender have also been reported to create similar patterns (see Figures 2.3 for word2vec and Figure 2.4 for GloVe). The existence of clusters and significant vector differences is not limited to semantic and world knowledge, but other linguistic properties have been identified in word embeddings. In Mikolov et al. (2013b), it has also been reported that linguistic patterns had been identified in word embeddings, such as plurals and verb tense (see Figure 2.3).

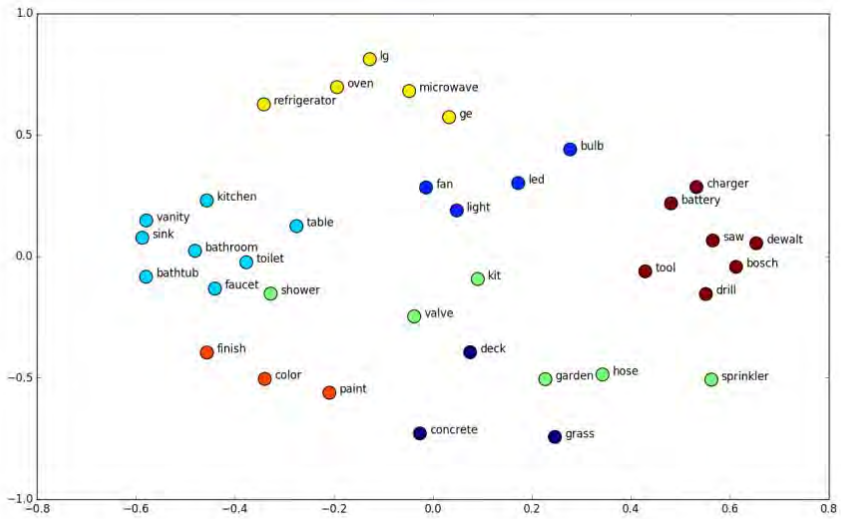


Figure 2.1: Three-dimensional PCA projection of 300-dimensional word2vec embeddings of various words. The colors mark manually annotated semantic clusters—though there are visualization methods for coloring generated clusters. Source: Lynn (2018)

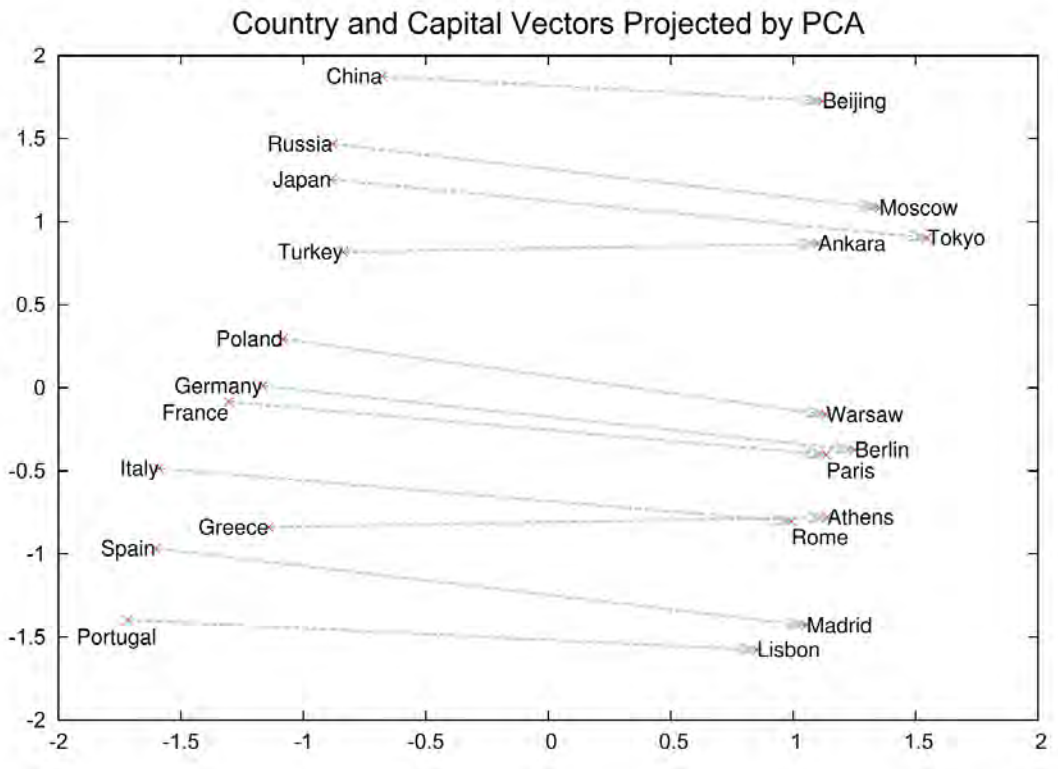


Figure 2.2: Two-dimensional PCA projection of 1000-dimensional word2vec word vectors of countries and their capital cities. Source: Mikolov et al. (2013c)

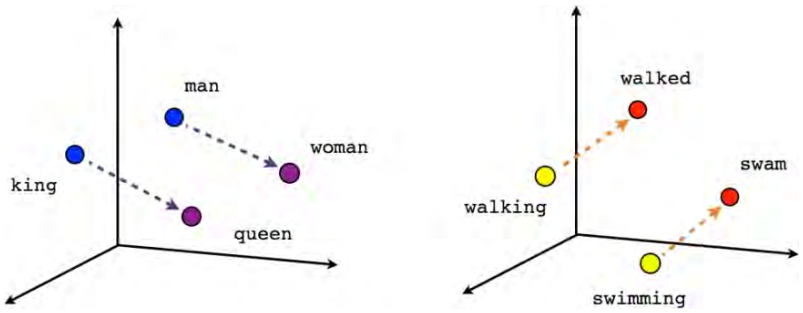
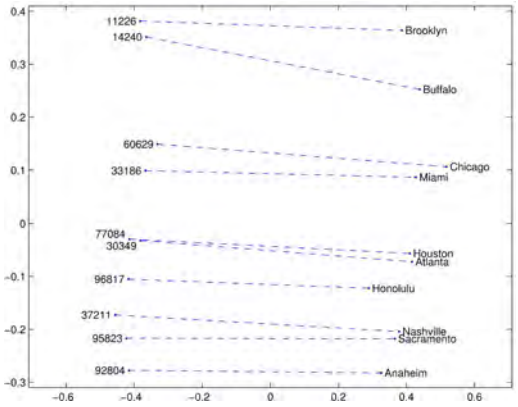
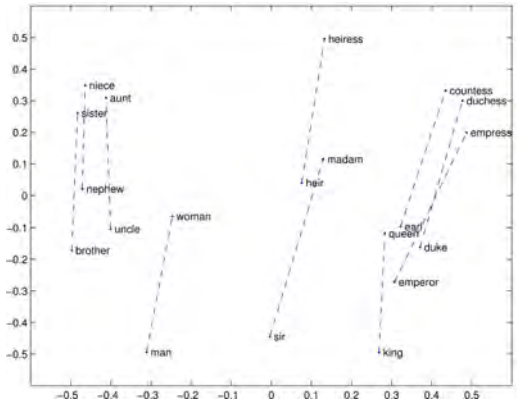


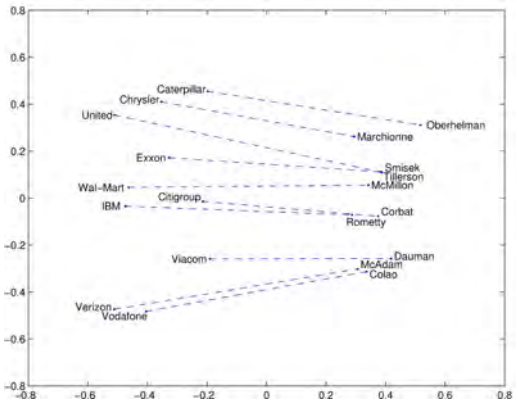
Figure 2.3: Two-dimensional PCA projection of 1000-dimensional word2vec word vectors of gender and verb tense word relations. Source: Lynn (2018)



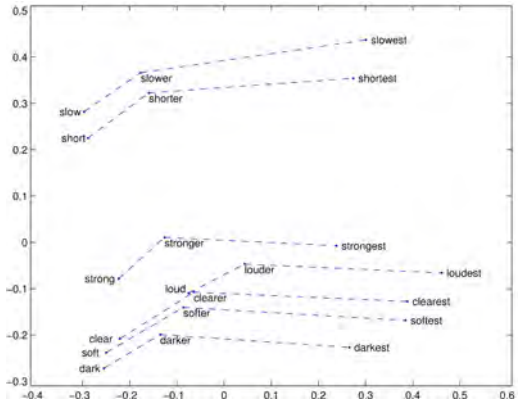
Zip code – City



Male – Female gender



Company – CEO



Comparative – Superlative degree

Figure 2.4: Two-dimensional PCA projection of 300-dimensional GloVe word vectors. Source: Pennington (2014)

2.4 Transformer architecture

2.4.1 Traditional attention mechanisms

The idea of creating an external memory of prior knowledge as a form of top-down attention has existed for a long time, with architectures such as the Neural Turing Machines (Graves et al., 2014) and End-to-End Memory Networks (Sukhbaatar et al., 2015). However, bottom-up attention, i.e. the ability to process input and deduce its most important parts without the need for prior specialized knowledge, has been further explored in machine learning. Early attention mechanisms, first used for convolutional neural networks in Computer Vision, were implemented as a saliency map of low and high-level visual features (Itti et al., 1998), and later as additional attention modules applied on feature maps (e.g. for computer vision in Rodriguez et al., 2018, for natural language processing in Shen and Huang, 2016).

However, the attention mechanisms that have been established and extensively used and developed for natural language processing are applied to sequential models. The first attention mechanism originated from Bahdanau et al. (2014) for neural machine translation encoder-decoder RNN models. This mechanism and its variations (e.g. different functions and optimization techniques, added elements) can be applied to seq2seq models regardless of their inner encoder-decoder architecture. The attention mechanism aims to encode the input sentence into a sequence of vectors, instead of treating the entire input simultaneously. Then, it dynamically chooses a subset of these encoded vectors in the decoding process. Its goal is to show how the input element of a sequence correlates to the other elements and "highlight" the important ones, in order to help the decoder make better-informed decisions. An illustration of additive/Bahdanau attention in a seq2seq model can be seen in Figure 2.5.

The encoder is a bidirectional RNN responsible for creating an annotation $\mathbf{h}_i$ for every word x_i in an input sequence of T elements (words). The annotation is the concatenation of the forward pass hidden state $\overrightarrow{\mathbf{h}_i}$ and the backward pass hidden state $\overleftarrow{\mathbf{h}_i}$ (see Equation 2.1).

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i^T}; \overleftarrow{\mathbf{h}_i^T}] \tag{2.1}$$

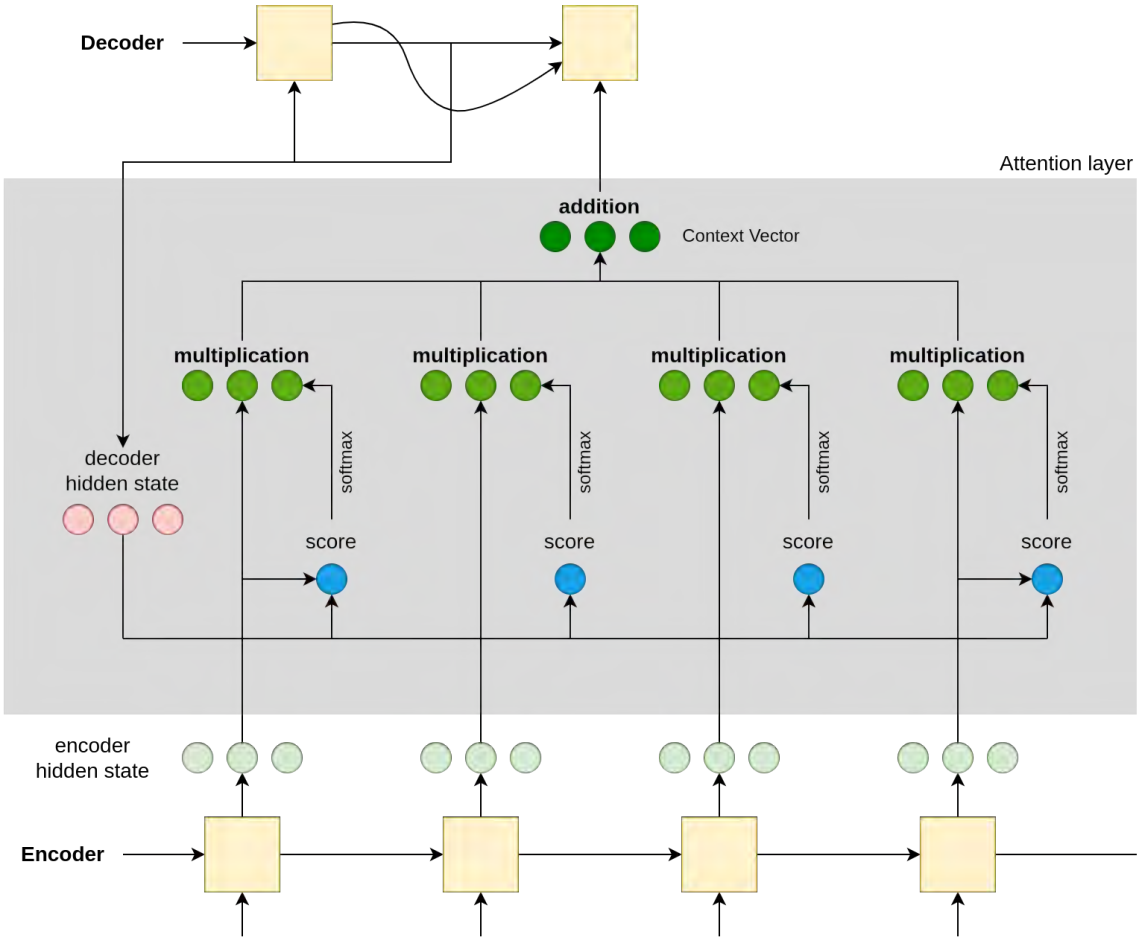


Figure 2.5: Visualization of the attention layer from the encoder input to the decoder output, for a neural architecture with Bahdanau attention. Source: Karim (2019)

The decoder computes its hidden states and outputs at each timestep t, as seen in Equation 2.2. In order to align the decoder output with the corresponding encoder input (which is important for machine translation), it uses the annotations of the encoder $\mathbf{h}_i$ and passes them to an alignment function $\alpha(.)$ alongside the decoder's output $\mathbf{s}_{t-1}$ at the previous timestep $t-1$, to create the attention score $e_{t,i}$ at timestep t (see Equation 2.3). The alignment function is an additive process, hence the alternative name *additive attention*.

$$\mathbf{s}_t = \text{RNN}_{\text{decoder}}(\mathbf{s}_{t-1}, y_{t-1}) \tag{2.2}$$

$$e_{t,i} = \alpha(\mathbf{s}_{t-1}, \mathbf{h}_i) \tag{2.3}$$

The implementation of the alignment function α can be performed either with a weight matrix $\mathbf{W}$ over the vectors $\mathbf{s}_{t-1}$ and $\mathbf{h}_i$ or by applying the attention matrices $\mathbf{W}_1, \mathbf{W}_2$ respectively on them (see Equation 2.4). The alignment outputs are parameterized as a feedforward neural network and jointly trained with the architecture, i.e. the model still observes the entire input, but the alignment function provides additional information on specific parts of the input.

$$\begin{aligned} \alpha(\mathbf{s}_{t-1}, \mathbf{h}_i) &= \mathbf{v}^T \tanh(\mathbf{W}[\mathbf{h}_i \; ; \; \mathbf{s}_{t-1}]) \\ \alpha(\mathbf{s}_{t-1}, \mathbf{h}_i) &= \mathbf{v}^T \tanh(\mathbf{W}_1\mathbf{h}_i + \mathbf{W}_2\mathbf{s}_{t-1}) \end{aligned} \tag{2.4}$$

A softmax function is applied to the decoder output weights, in order to normalize the output values in a range from 0 to 1. According to Bahdanau et al. (2014), this method allows the decoder to decide which parts of the input should be attended to. The annotations α are saved as a weighted sum in the context vector $\mathbf{c}_t$ (see Equation 2.5) which is updated after every decoding time step. The hidden state of the attention mechanism $\widetilde{\mathbf{s}}_t$ is computed based on a weighted concatenation of the context vector $\mathbf{c}_t$ and the current decoder hidden state $\mathbf{s}_t$, as seen in Equation 2.6. Finally, the decoder's final output y_t is computed, by applying a weight $\mathbf{W}_y$ and a softmax function to the attention's hidden state $\widetilde{\mathbf{s}}_t$ (see Equation 2.7).

$$\mathbf{c}_t = \sum_{i=1}^{T} \alpha_{t,i} \mathbf{h}_i \tag{2.5}$$

$$\widetilde{\mathbf{s}}_t = \tanh(\mathbf{W}_\mathbf{c}[\mathbf{c}_t \ ; \ \mathbf{s}_t]) \tag{2.6}$$

$$y_t = \text{softmax}(\mathbf{W}_y \widetilde{\mathbf{s}}_t) \tag{2.7}$$

Luong et al. (2015) proposed alternative ways to Bahdanau et al. (2014) of computing α alignment scores, either with the additive method of Bahdanau and a trainable weight matrix $\mathbf{W}_\alpha$ or with multiplicative attention and weights $\mathbf{W}_\alpha$, $\mathbf{v}_\alpha$ (the three methods proposed are seen in Equations 2.8). With multiplication, they claim to capture the similarity of the encoder's hidden state and the decoder's output more closely.

$$\begin{aligned} \alpha(\mathbf{s}_t, \mathbf{h}_i) &= \mathbf{v}_\alpha^T \tanh(\mathbf{W}_\alpha[\mathbf{s}_t \ ; \ \mathbf{h}_i)] \\ \alpha(\mathbf{s}_t, \mathbf{h}_i) &= \mathbf{s}_t^T \mathbf{h}_i \\ \alpha(\mathbf{s}_t, \mathbf{h}_i) &= \mathbf{s}_t^T \mathbf{W}_\alpha \mathbf{h}_i \end{aligned} \tag{2.8}$$

Other ways of diversifying attention have emerged for different tasks. Wu et al. (2016) are processing the context vector and decoder output by layering multiple LSTMs to further process the encoder and decoder outputs. *Global attention* is the mechanism that attends to all encoder hidden states (as described above) and *local attention* focuses on a window of hidden states in each timestep, either based on the current decoder position or with predictive alignments (Luong et al., 2015). Local attention creates the context vector by computing a weighted average over the set of annotations and hidden states $\mathbf{h}_i$ limited by a window around a position $\mathbf{p}_t$. The length of the window can be created either empirically or with trainable parameters based on the input.

Types of attention that emerged from computer vision terminology are *soft* and *hard* attention. Soft attention uses as input the encoder's weighted inputs (and is equivalent to global attention), while hard attention uses the attention scores to select one of the hidden states to focus on (Xu et al., 2015; Yang, 2020).

Self-attention or intra-attention was originally proposed by Cheng et al. (2016) for machine reading, with the motivation that each sequence would contain a representation of its own elements. They were also able to apply this mechanism to an encoder-decode architecture with promising results. However, the self-attention module that has revolutionized the world of natural language processing was introduced by Vaswani et al. (2017). In this work, multi-head self-attention is used alongside a novel neural network architecture called a *Transformer*.

2.4.2 The self-attention mechanism

Self-attention is a type of attention mechanism that allows the inputs of the model to interact with each other, unlike general attention in which the output interacts with each input. In Vaswani et al. (2017), self-attention is implemented with a *Transformer* encoder-decoder model: the encoder is made of 6 layers with 2 sub-layers each. Before explaining the structure of the Transformer in Section 2.4.3, it is necessary to examine how the self-attention mechanism is built and how it processes input.

In simple terms, the self-attention attention function is effectively a mapping from a *query* to a set of *key-value* pairs and then to an output. In order to explain the interaction of queries, keys, and values, a parallel could be drawn with information retrieval; a database of people contains tuples of keys-values, where a key would correspond to a last name and its value to the first name. A query to this dataset may be an exact or approximate match to one of the keys, in which case the value would be returned, or may not correspond to any keys, in which case there would be no valid answer (Zhang et al., 2021). During the training process, the self-attention mechanism will learn the similarity of a query and a key as an *attention weight*. In NLP, the keys and values correspond to the alignment of input and expected output (e.g. between the source and target tokens in machine translation) or the input and its extracted features (e.g. in classification).

The result of the self-attention mechanism is calculated as a weighted sum of the values, with each value's weight determined by the query's compatibility function with its associated key. A graphic demonstration of how the mechanism of self-attention produces output on the first input segment can be found in Figure 2.6. A mathematical presentation of the main components follows.

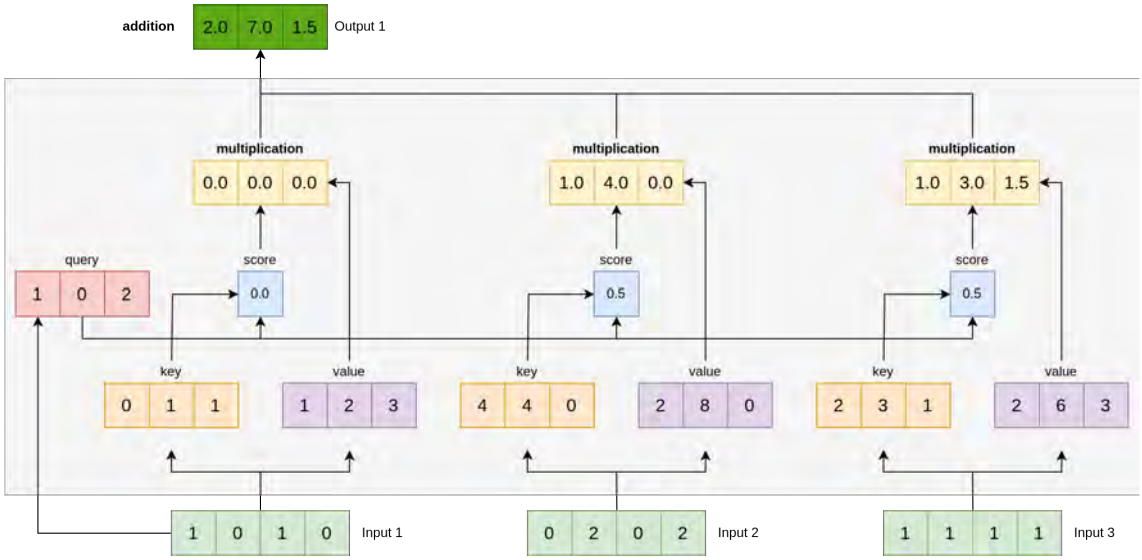


Figure 2.6: How the mechanism of self-attention produces output on the first input segment, treating each input segment separately but in parallel. The query comes from the decoder's hidden state, and the key and value come from the encoder's hidden states. Source: Karim (2022)

First of all, each component of the input vector is split into three representations: the query q, the key k, and the value v. These representations are contained in matrices Q, K, V respectively, but are not directly used by the model; instead, each attention module of the self-attention mechanism initializes its own projection matrices W^Q, W^K, W^V and aggregates the queries, keys, and values in the respective matrix (Adaloglou, 2021). In this stage, *bias* may also be added.

The attention scores for each input are the *dot product* of the input's query and of all the keys of each input, including the current input's key, and it is *scaled* down to create more stable gradients. Hence in Vaswani et al. (2017) it is referred to as *scaled dot-product attention*. Its calculation is shown in Equation 2.9. Vaswani et al. (2017) state that this method is equivalent to the multiplicative attention of Luong et al. (2015) with the added scaling factor. The scaling factor is a division by the square root of the dimension of query and key, $\frac{1}{\sqrt{d_k}}$ (Cristina, 2022a).

$$\text{attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\mathsf{T}}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.9}$$

In Equation 2.9, the alignment scores $\mathbf{QK}^\mathsf{T}$ are produced by multiplying the queries of matrix $\mathbf{Q}$ with the keys of matrix $\mathbf{K}$, where m is the length of queries and n is the length of keys:

$$\mathbf{QK}^T = \begin{bmatrix} e_{11} & e_{12} & \dots & e_{1n} \\ e_{21} & e_{22} & \dots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \dots & e_{mn} \end{bmatrix}$$

Each of these alignment scores e is scaled and passed through a softmax function, in order to obtain a matrix of weights. The weight is assigned based on the similarity of the query with the corresponding key. The scores are normalized with a softmax function, and then each softmaxed attention score for each input is multiplied by the input's corresponding value to produce a weighted value.

$$\text{softmax}\left(\frac{\mathbf{QK}^T}{\sqrt{d_k}}\right) \cdot \mathbf{V} = \begin{bmatrix} \text{softmax}(\frac{e_{11}}{\sqrt{d_k}} & \frac{e_{12}}{\sqrt{d_k}} & \dots & \frac{e_{1n}}{\sqrt{d_k}}) \\ \text{softmax}(\frac{e_{21}}{\sqrt{d_k}} & \frac{e_{22}}{\sqrt{d_k}} & \dots & \frac{e_{2n}}{\sqrt{d_k}}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{softmax}(\frac{e_{m1}}{\sqrt{d_k}} & \frac{e_{m2}}{\sqrt{d_k}} & \dots & \frac{e_{mn}}{\sqrt{d_k}}) \end{bmatrix} \cdot \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1d_v} \\ v_{21} & v_{22} & \dots & v_{2d_v} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \dots & v_{nd_v} \end{bmatrix} \tag{2.10}$$

All weighted values for each input are then added, in order to produce the query representation, a sum of attention scores that also includes the input's self-representation.

*

* *

In Vaswani et al. (2017), the concept of *multi-headed attention* was also introduced (see Figure 2.7). Their self-attention mechanism is composed of 8 attention heads, where each head has different weight matrices for queries, keys, and values. The operations of self-attention are the same as described above, but they are performed by each attention head separately. This means that the attention heads have identical architecture and operate on the same feature space, but each attention head has its own key, value, and query matrices, and computes its own output matrix. Hence they are "free" to attribute different weights by learning different functions. Thus, the attention function

is able to extract information from different representation sub-spaces. While there is no assurance that these weights will be learned differently, the gradient descent process encourages following heads to become increasingly more sophisticated (Bloem, 2019). The attention function for each attention head can be summed up in Equation 2.11 as the weighted output of Equation 2.10, i.e. the head's learned projection matrices. The attention weights of the multi-head attention mechanism are saved in matrix $\mathbf{W}^O$, and the attention head functions are concatenated in Equation 2.12. The final output is the weighted dot product of the concatenation. This brings a groundbreaking capability to neural network models to perform parallel computations and create sub-spaces with different specializations.

$$\text{head}_i = \text{attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \tag{2.11}$$

$$\text{multihead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}^O \tag{2.12}$$

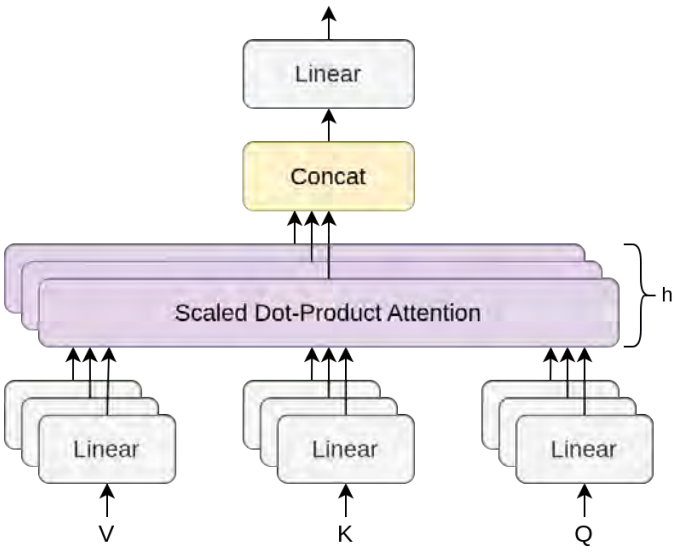


Figure 2.7: The mechanism of multi-headed self-attention. The layers in the figure represent the different attention heads. Source: Vaswani et al. (2017)

2.4.3 The Vaswani Transformer architecture

Multi-headed attention was introduced alongside a novel neural network architecture, the **Transformer** architecture (Vaswani et al., 2017). It is an encoder-decoder model that utilizes multi-headed attention in three different ways; the encoder-decoder mechanism interacts via attention in the same way that general attention allows. However, both the encoder and the decoder contain their own self-attention layer: self-attention in the encoder is responsible for providing attention scores to the current encoder state from the encoder's previous states. Meanwhile, the decoder's self-attention mechanism informs the decoder's current state on past and current positions (up to the current one included) but also attends to the (unknown) future states by *masking* them (Alammar, 2018). At each step the model is autoregressive, consuming the previously generated symbols as additional input when generating the next. A visualization of the Transformer architecture can be seen in Figure 2.8, and a detailed explanation of each component follows.

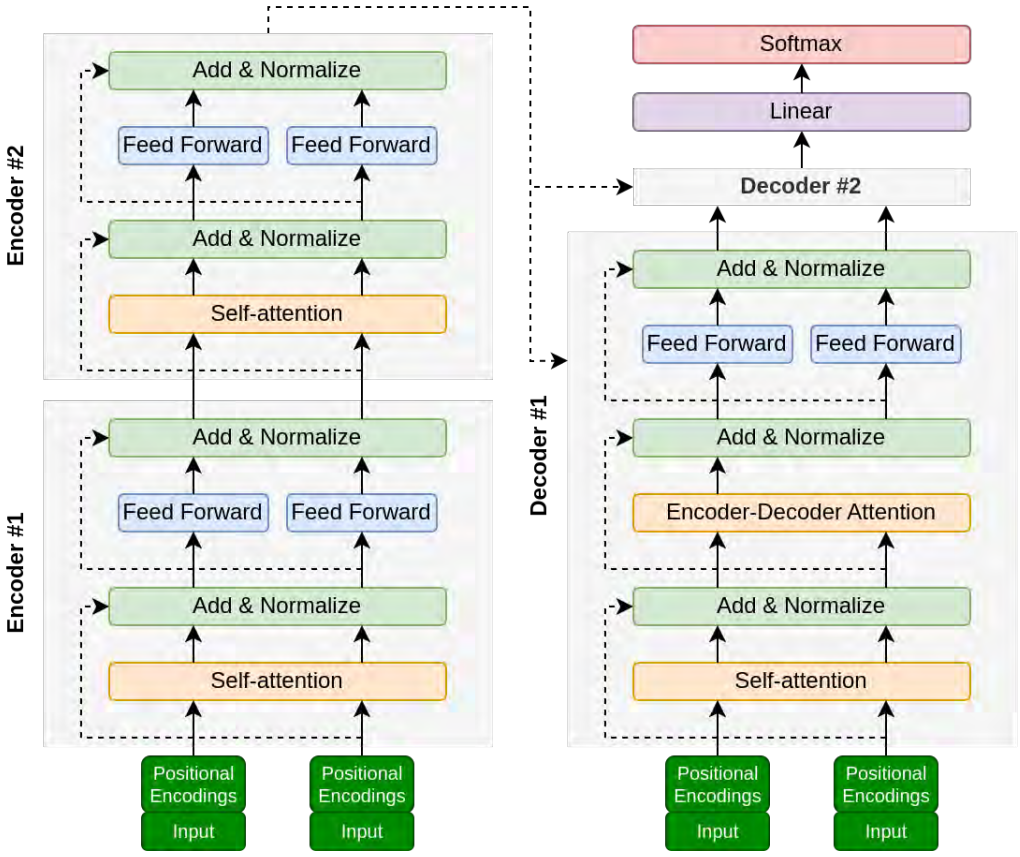


Figure 2.8: Visualization of a double encoder-decoder neural network with the Transformer architecture. Source: Alammar (2018)

Input

In Transformer models, input is composed of two vectors: the *embedding vector* and the *positional encodings* (see Figure 2.9). The embedding process involves the conversion of each text token into a vector with continuous values, based on a pre-existing vocabulary and mappings or arbitrary values. Splitting text into tokens may be performed on a character level, sub-word or word level, or longer sequences.

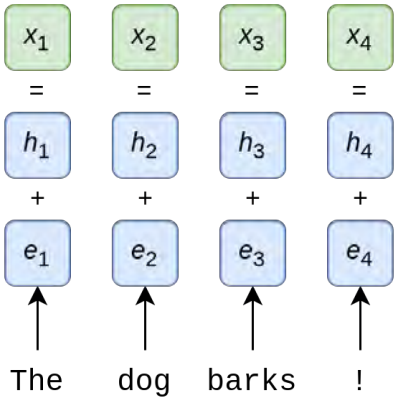


Figure 2.9: A visualization of converting textual input to the input sequence for a Transformer model.

The length of each embedding vector is predefined by the model as d_{model}. Each embedding vector is also augmented by summing it (element-wise) to a positional encoding vector of the same d_{model} length. The addition of positional encodings in Transformer neural architectures is necessary since they do not process input linearly as earlier neural architectures do, hence they do not retain word position information.

The positional encoding provides the location or position of an item in a sequence, in order to provide each token a distinct representation. At a first glance, indexing the input with integers would be a sufficient way to represent the token position; however, absolute positional information has proven to be problematic for the self-attention mechanism. The parallel operations of the attention mechanism would be inefficiently slow with a large input sequence. Normalizing the indices to a scale of 0-1 (e.g. with a sigmoid function) could cause additional problems since the position would not be properly represented with very small numbers.

Transformer models implement positional encodings in which each location or index

is converted into a vector. The output of the positional encoding layer is a matrix, where each row of the matrix represents a token that has been encoded and given positional information. The vectors can either be learned or fixed a priori—in Vaswani et al. (2017) they are fixed, based on a schema of sine and cosine functions in different frequencies (Zhang et al., 2021). For an input sequence of length $\mathbf{L}$, the vector of dimension d with the position p, for a token with position k and index i, is given by Equation 2.13[1]. In simple terms, every even position P in the final vector is calculated by the sine function and every odd with the cosine function (Saeed, 2022).

$$\begin{aligned} P_{k,2i} &= \sin\left(\frac{k}{10000^{2i/d}}\right), \\ p_{k,2i+1} &= \cos\left(\frac{k}{10000^{2i/d}}\right). \\ 0 \leq & k < \mathbf{L}/2 \\ 0 \leq & i < d/2 \end{aligned} \tag{2.13}$$

For example, the positional encoding of size $d = 3$ for the second token of a sequence with $k = 1$ would be:

$$\begin{aligned} [P_{10}, P_{11}, P_{12}] &= \left[\sin\left(\frac{1}{10000^{2\frac{0}{3}}}\right), \cos\left(\frac{1}{10000^{2\frac{1}{3}}}\right), \sin\left(\frac{1}{10000^{2\frac{3}{3}}}\right)\right] \\ &= [0.84, 0.99, 0.00] \end{aligned}$$

Encoder

The task of the encoder in a neural architecture is to create a mapping of the input sequence into an abstract continuous representation. This representation is encoded and contains learned information on the properties and patterns of the input, in order to pass to the decoder.

Focusing on the encoder of the Vaswani et al. (2017) Transformer model, it consists of a stack of 6 identical layers, built by 2 sub-layers. The first sub-layer contains the multi-head self-attention mechanism, as described in Section 2.4.2. The second sub-

[1]The value $10,000$ is a user-defined scalar, set to $10,000$ in Vaswani et al. (2017).

layer is a fully connected feed-forward network composed of two linear transformations with Rectified Linear Unit (ReLU) activation. The mathematical interpretation of the feedforward neural network can be seen in Equation 2.14 (Cristina, 2022b). Each sublayer applies the same transformations to the input sequence but uses different weights $\mathbf{W}_1$, $\mathbf{W}_2$ and biases b_1, b_2.

$$\text{FFN}(x) = \text{ReLU}(\mathbf{W}_1 x + b_1)\mathbf{W}_2 + b_2 \tag{2.14}$$

Each sublayer is succeeded by a normalization layer, that normalizes the sum computed between the sublayer input and the output generated by the sublayer itself (see Equation 2.15). The self-attention sublayer is followed by a layer-normalization step, a technique developed by Ba et al. (2016) which aims to stabilize the hidden state dynamics across training, computing the mean and variance used for normalization from all of the summed inputs in a layer.

$$\text{layernorm}(x + \text{sublayer}(x)) \tag{2.15}$$

The encoder blocks are also connected with each other with *skip* or *residual connections*, pictured in Figure 2.8 as dashed lines. Residual connections were originally introduced for image processing with convolutional networks (He et al., 2016; Ronneberger et al., 2015; Huang et al., 2017), and have proven to be beneficial against the *vanishing gradient problem*. Their objective is to allow gradients to pass through a network directly, without passing through activation functions. The training signal gets multiplied by the derivative of the activation function, and in the case of ReLU, the gradient often gets multiplied by zero. The addition of residual connections ensures the transfer of the training signal without being lost or affected. Additionally, the existence of this original, unchanged information allows the Transformer to "remember" the original state alongside the transformed representations, thus better representing the input (Libovický, 2022).

The current encoder outputs are passed to a feedforward layer, in order to project the

output of self-attention in a higher dimensional space. This output is also normalized and another skip connection is added.

Decoder

The decoder, on the right half of the architecture, receives as input its own predicted output word at time step $t-1$ and additionally the positional encoding, in the same way as the encoder. The decoder layer in Vaswani et al. (2017) is composed of 6 layers, each containing 3 sub-layers. While the building blocks of the decoder appear to be similar to those of the encoder architecture in Figure 2.8, they perform different functions.

The encoder is made to attend to every word in the input sequence regardless of its position. However, the decoder is modified to focus exclusively on the words that come before the current input. Receiving the previous output of the decoder stack, the first decoder sublayer adds positional information to it and applies multi-head self-attention to it. Therefore, the prediction for a word at a given position i in the sequence can only be based on the known outputs for the antecedent words. This is accomplished in the multi-head self-attention mechanism by applying a mask on the results of the scaled multiplication of matrices $\mathbf{Q}$ and $\mathbf{K}$. Masking is the suppression of the matrix values that correspond to the connections that should be ignored (i.e. the subsequent words), as seen in Equation 2.16.

$$\text{mask}(\mathbf{Q}\mathbf{K}^T) = \text{mask}\left(\begin{bmatrix} e_{11} & e_{12} & \dots & e_{1n} \\ e_{21} & e_{22} & \dots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \dots & e_{mn} \end{bmatrix}\right) = \begin{bmatrix} e_{11} & -\infty & \dots & -\infty \\ e_{21} & e_{22} & \dots & -\infty \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \dots & e_{mn} \end{bmatrix} \tag{2.16}$$

The second sub-layer of the decoder implements a multi-head self-attention mechanism. It receives the queries from the previous decoder sublayer and the keys and values from the output of the encoder. This allows the decoder to attend to all the words in the input sequence. The third sub-layer is composed of a fully connected feed-forward network, followed by a softmax layer, to generate a prediction for the next word of the output sequence. Finally, the three decoder sub-layers are connected with residual connections and are followed by a normalization layer (Cristina, 2022b).

2.4.4 Transfer learning

The learned representations of a neural network can be extracted and used as embeddings for a new model. It is also possible to train them again and further specialize them for a specific task, thus infusing them with additional knowledge for a target task. This method is called **transfer learning**; apart from the benefit of additional knowledge and specialization, it also is computationally cheaper, since it requires a smaller amount of data and processing, while still providing state-of-the-art results.

Transfer learning is a machine learning research topic that allows the storage of knowledge obtained while training for one task, and applying it to another, unrelated task. This method has been shown to improve the performance of the second task, in line with previous findings that pre-existing knowledge is better than random initializations of input encodings. In transfer learning, a base network is trained on a source dataset and task, its features are extracted, and these learned features are transferred to a second target network that is trained on a target dataset and task. If the traits are general—that is, applicable to both the base task and the target task—rather than task-specific, this procedure is more likely to succeed (Yosinski et al., 2014).

Ruder (2019) defines two categories and four types of transfer learning. In Figure 2.10, the taxonomy that they have proposed is presented.

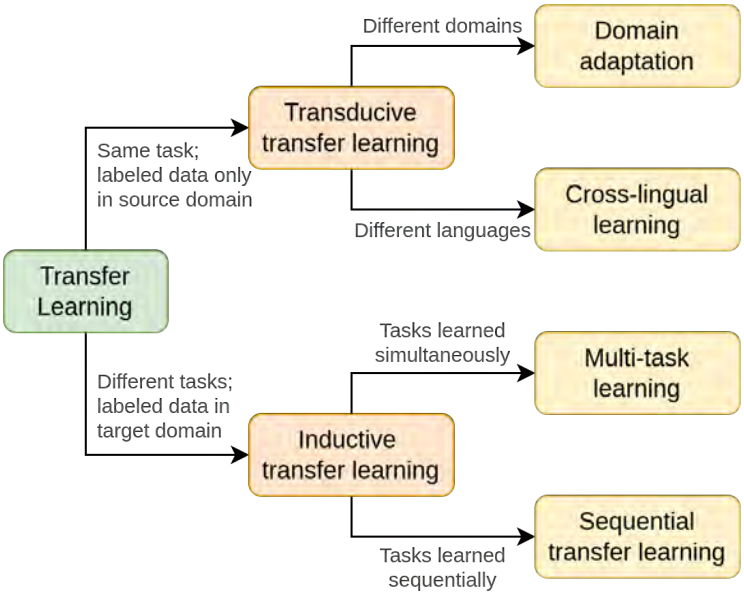


Figure 2.10: Taxonomy of transfer learning methods. Source: Ruder (2019)

Transductive transfer learning involves the same source and target tasks in both stages, and *domain adoption* uses data from different domains while *cross-lingual learning* uses data from other languages. In *inductive transfer learning*, the source and target tasks are different; in *multi-task learning* the models are trained on multiple related tasks simultaneously, while in *sequential transfer learning* the model is trained on source data, and as a second phase, the trained model is trained again for the target task, in order to become adapted and specialized to the target task. These two phases involve the *pretraining* process, followed by *feature extraction* or *finetuning*.

Pretraining is the phase where the model is being trained with the source data and task, which are ideally as close to the target task as possible. If the pretrained model's weights are kept, they can be immediately "adopted" as embeddings (feature extraction), or they can be further adjusted to the target task (finetuning). In feature extraction, single parts (sentences or characters) are extracted to a fixed-length matrix with dimensions $\mathbb{R}^n \times k$, where n is the size of the vocabulary and k is the fixed length. The weights of the model do not change, but the model learns a linear combination of the top layer (Peters et al., 2019). However, fine-tuning involves a second training process in which the pretrained model's weights are updated on the target task. This allows for better performance and specialization on a specific task. However, this updating technique may bring some shortcomings. A finetuned model may lose some general knowledge and relationship between words learned during the pretraining phase, or only update words that exist in the target data, with other words being left "unseen".

The finetuning process can be performed in multiple ways, by training the entire architecture, training some layers, or **freezing** the entire architecture. When training the entire architecture, the pretrained model is trained with the target dataset, and the output is passed to a softmax layer. The error is backpropagated through the entire architecture and the pretrained weights of the model are updated based on the target task. Freezing a layer or the entire architecture means disabling gradient computation and backpropagation for the weights of these layers, i.e. disallowing the update of these weights. For example, the weights of the initial layers of the model can be kept frozen while updating only the higher layers. It is also possible to freeze the entire architecture and add an additional final layer (or more) to be trained and updated.

2.5 Contextualized embeddings with Transformers

2.5.1 Introduction

The use of pretrained embeddings learned from neural architectures preexisted the existence of Transformer architectures. The idea of creating context-dependent word embeddings also predates deep contextualized embedding models from Transformers, made ubiquitous after the work of Radford et al. (2018) and Devlin et al. (2019). Distributional information can be combined with document occurrence, in order to create networks of related words. A first attempt to create word distributional representations with neural methods appeared in Gallant (1991). Hand-encoded semantic features are used alongside a context algorithm, in order to generate a dynamic context vector for a word at any position. The work by Bengio et al. (2000) is the first self-supervised method with a neural network, where the probability for word sequences is learned simultaneously with a distributed representation for each word. This approach outperformed state-of-the-art statistical methods of the time with trigrams since it allowed the generalization that similar words may be interchangeable in a known phrase. Reisinger and Mooney (2010) also developed context-dependent word embeddings that are capable of capturing polysemy and homonymy.

However, the Transformers' unique properties of multiple representations and parallel operations have generated dynamic embeddings that are sensitive to multiple contexts. These embedding models have been considered the most important advancement in NLP after word2vec models. In the following sections, we are examining ELMo, a deep contextualized word embedding model based on a neural architecture (CNN and LSTMs) that served as the mold for subsequent Transformer-based embedding models. We later focus on autoregressive models (GPT, XLNet) and autoencoder-like models (BERT, RoBERTa, ALBERT, CamemBERT, FlauBERT). Even though we are only using the latter in this doctoral research, it is necessary to briefly talk about the ELMo and GPT models, in order to understand the structure of more complex models and the motivation to discard direct output and only use the pretrained word embeddings.

2.5.2 ELMo

Peters et al. (2018) introduced Embeddings from Language Models (ELMo), a deep contextualized word representation model that is able to capture complex characteristics of word use (e.g. syntax and semantics) and their possible variations in different linguistic contexts. The model and its contextual embeddings significantly improved the state of the art—at the time—across a broad range of challenging NLP problems, including question answering, textual entailment, and sentiment analysis.

The model proposed in Peters et al. (2018) is similar to the TagLM model previously created by Peters et al. (2017) for sequence tagging (part-of-speech, text chunking, and named entity recognition). A graphic depiction of the architecture is shown in Figure 2.11. The training objective of the ELMo model is neighbor word prediction (either to the left or the right, independently).

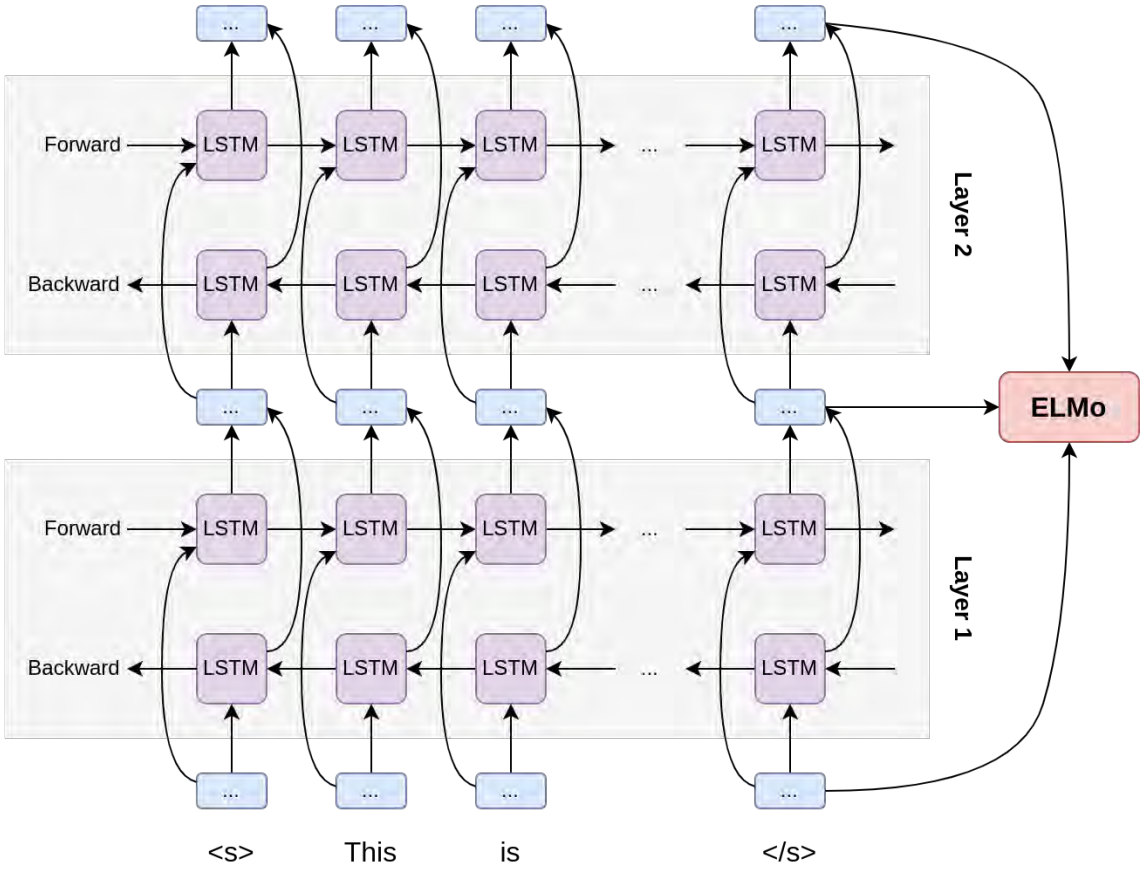


Figure 2.11: The structure of the biLM model to produce ELMo embeddings. Source: Hagiwara (2021)

A character-level CNN (inspired by Jozefowicz et al., 2016) is used to initialize token embeddings and character-level embeddings for the training process. A character/token representation $\mathbf{x}_k^{LM}$ is passed through a deep bidirectional language model (biLM) with $\mathbf{L}$ layers of LSTM neural networks. At each position k, each LSTM layer outputs a context-dependent representation $\overrightarrow{\mathbf{h}}_{k,j}^{LM}$ where j is the layer index. The top layer's LSTM output, $\overrightarrow{\mathbf{h}}_{k,L}^{LM}$, is used to predict the next token t_{k+1} with a softmax layer.

As seen in Figure 2.11, the input phrase is scanned twice in each layer, once with a forward and once with a backward pass. The internal states from the backward pass are calculated from the word itself and the future context, while the internal states from the forward pass are calculated from the word and its past context. Therefore, the calculation of the probability of token t_k is similar for both passes (see Equation 2.17).

$$\begin{aligned} p\left(t_1, t_2, \ldots, t_N\right) &= \prod_{k=1}^{N} p\left(t_k \mid t_1, t_2, \ldots, t_{k-1}\right) \\ p\left(t_1, t_2, \ldots, t_N\right) &= \prod_{k=1}^{N} p\left(t_k \mid t_{k+1}, t_{k+2}, \ldots, t_N\right) \end{aligned} \tag{2.17}$$

An intermediate word vector concatenates these two states with both contexts, which differentiates the biLM model from a traditional bidirectional architecture. (In a bidirectional architecture, the internal states of the forward and backward passes would be concatenated before being passed to the next layer). Every layer of the model computes its own internal states and the final representation is the weighted combination of the input word vectors and all intermediate word vectors, as shown in Equation 2.18. The variables Θ_x and Θ_s are the token and softmax parameters respectively, which are combined for both directions, while the $\vec{\Theta}_{LSTM}$ parameters of each LSTM are unique.

$$\sum_{k=1}^{N} \left(\log p\left(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \vec{\Theta}_{LSTM}, \Theta_s\right) + \log p\left(t_k \mid t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s\right) \right) \tag{2.18}$$

ELMo is, according to Peters et al. (2018), a "task-specific combination of the intermediate layer representations in the biLM". Each layer $\mathbf{L}$ computes a set of representations

based on the tokens' previously observed hidden states (see Equation 2.19).

$$\begin{aligned} R_k &= \left\{ \mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1, \ldots, L \right\} \\ &= \left\{ \mathbf{h}_{k,j}^{LM} \mid j = 0, \ldots, L \right\}, \end{aligned} \tag{2.19}$$

These models can be used directly for NLP tasks or adapted to a target task with additional data. The ELMo-specific task is formulated by Equation 2.20. R_k is the representation of the tokens t_k and Θ is the size of the context window. γ is the optimization task-specific parameter to scale the model and s_j^{task} is a softmax function to normalize weights $\mathbf{h}$ per token index k and model layer j (Becker, 2020).

$$\mathbf{ELMo}_k^{\text{task}} = E\left(R_k; \Theta^{\text{task}}\right) = \gamma^{\text{task}} \sum_{j=0}^{L} s_j^{\text{task}} \mathbf{h}_{k,j}^{LM} \tag{2.20}$$

The pretrained model is frozen and the task-specific weights are calculated, to produce task-specific representations for every token t_k, a linear combination of the internal representations.

The authors used ELMo pretrained embeddings from a trained biLM model and provide them as language representations to a new model, and also used the ELMo output as weights to the hidden states of the new model's output. With this method, they reported an accuracy improvement of 6-20% on NLP benchmark tasks: question answering, textual entailment, semantic role labeling, named entity extraction, co-reference resolution, and sentiment analysis. These pretrained embeddings can be further improved with the use of *feature extraction*.

The creators of the ELMo architecture have released different versions of pretrained word embeddings, a small and medium-sized model trained on 800 million tokens from the One Billion Word Benchmark (Chelba et al., 2013), and a large model trained on 5.5 billion tokens from Wikipedia dumps and the monolingual news crawl data from the Workshops on Statistical Machine Translation (WMT) from 2008 to 2012.

2.5.3 GPT

Generative Pre-trained Transformer (GPT) (Radford et al., 2018) was the first architecture to produce deep contextualized word representations using a Transformer decoder. Self-attention in a Transformer allows the model to focus on the most relevant parts of the input when processing each part of the input, and the Transformer itself is capable of parallelizing processes to compute weights faster. Word embeddings from the GPT model are trained in a two-step approach, with the generative pretraining (unsupervised) and the discriminate finetuning (supervised) being integral parts of the architecture.

The GPT model uses Byte-Pair Encoding (BPE) (Gage, 1994) for its input, a data-driven form of data compression which has been adapted for use in NLP applications (Sennrich et al., 2016). The vocabulary is encoded as a series of tokens, where common words will be encoded as a single token, while rare words will be segmented into subwords optimized by frequency. This method ensures the preservation of word-based encodings for frequent words while keeping the vocabulary size at a relatively reasonable size. It is also capable of handling unseen and rare words by decomposing them to (ideally) morphologically salient subwords.

The training objective of GPT is to calculate the probability of tokens $u_1, \ldots, u_n$ in the context vector $\mathcal{U}$ (see Equation 2.21), where k is the size of the context window, and the conditional probability P is modeled using a neural network with parameters Θ. These parameters are trained using stochastic gradient descent.

$$L_1(\mathcal{U}) = \sum_i \log P\left(u_i \mid u_{i-k}, \ldots, u_{i-1}; \Theta\right) \tag{2.21}$$

The internal architecture is a 12-layer decoder-only Transformer with 12 masked self-attention heads with 64-dimensional states each (for a total of 768). The architecture uses the Adam optimization algorithm, instead of stochastic gradient descent, and the Gaussian-error Linear Unit (GeLU) activation function instead of ReLU. This model applies a multi-headed self-attention operation over the input context tokens followed by position-wise feedforward layers. GPT employs the concept of *autoregression*; the output depends linearly on its own previous values, in a way reminiscent of RNNs, but harnessing the computational prowess of a Transformer architecture.

In Equation 2.22, $\mathcal{U}$ is the context vector of tokens, n is the number of layers, $\mathbf{W}_e$ is the token embedding matrix, and $\mathbf{W}_p$ is the position embedding matrix. GPT uses only one direction, calculating the Transformer's hidden state h_l based on the previous timestep h_{l-1} (forward pass).

$$\begin{aligned} h_0 &= \mathcal{U}\mathbf{W}_e + \mathbf{W}_p \\ h_l &= \text{transformer\_block } (h_{l-1}) \, \forall i \in [1, n] \\ P(u) &= \text{softmax}\left(h_n \mathbf{W}_e^T\right) \end{aligned} \tag{2.22}$$

Like ELMo, GPT can be finetuned for a specific task with labeled data, using the word embeddings from the previous pretraining process. With a labeled dataset $\mathcal{C}$ of input tokens $x^1, \ldots, x^m$ with a label y, the objective is to maximize the likelihood of the classification task. The inputs are passed through the pretrained GPT to obtain the final transformer block's activation h_l^m, which is then fed into an added linear output layer with parameters $\mathbf{W}_y$ to predict y (see Equation 2.23).

$$P\left(y \mid x^1, \ldots, x^m\right) = \text{softmax}\left(h_l^m \mathbf{W}_y\right) \tag{2.23}$$

In order to test GPT, the authors pretrained the model with a large corpus and finetuned the pretrained embeddings on benchmark NLP tasks: causal language modeling (CLM), natural language inference, question answering and commonsense reasoning, semantic similarity, and classification. A visualization of the finetuning process for the different tasks is provided in Figure 2.12.

Pretraining was performed with BookCorpus (Zhu et al., 2015) and around 1 billion tokens. According to Radford et al. (2018), the GPT pretrained embeddings were able to outperform traditional methods of neural networks and ELMo, in standardized tasks of semantic classification (entailment, contradiction), question answering, commonsense reasoning, and paraphrasing. The authors also highlighted the importance of transfer learning with finetuning on language-related tasks, which improves the generalization capacities of the model and accelerates computations.

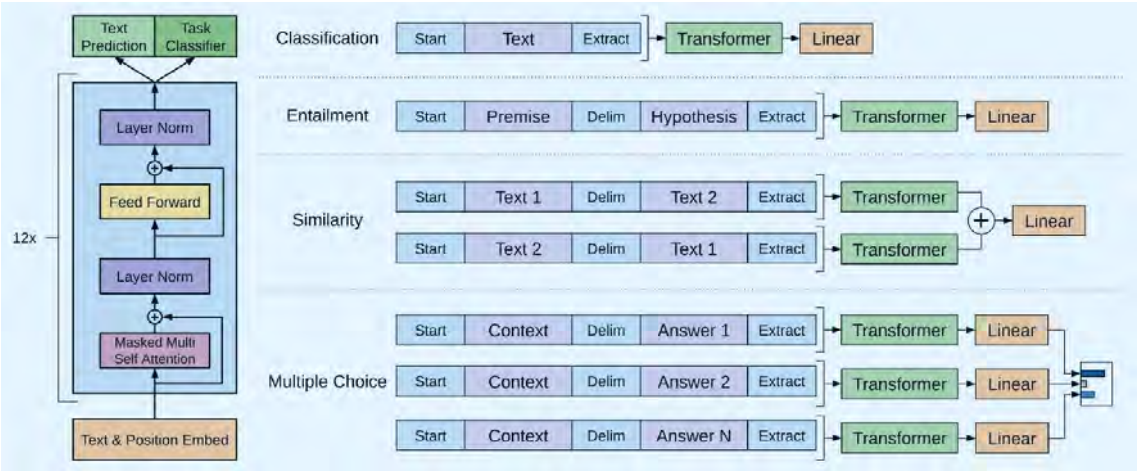


Figure 2.12: A visualization of the pretraining process of the GPT Transformer model (left) and of the task-specific inputs and finetuning processes. Source: Radford et al. (2018)

2.5.4 BERT

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) is a Transformer-based bidirectional encoder-decoder architecture, which shortly after its release became synonymous with deep contextualized word embeddings. One of the revolutionary aspects of BERT is its pretraining method, inspired by autoencoding (but not truly an autoencoder, as multiple sources support): the training process randomly samples positions in the input sequence and learns to fill the word in the masked position, while also learning to predict the next sentence given the first sentence. As Radford et al. (2018) have pointed out, the addition of language-motivated knowledge in a model is beneficial to its performance.

BERT's input is treated in three embedding layers that create three different representations, which are combined and passed to the training step. A visual is provided in Figure 2.13. In detail, the embedding layers composing the input are the following:

- The **token embedding layer** uses the WordPiece tokenization algorithm (Wu et al., 2016), a data-driven method similar to BPE, that creates words and subwords out of the input text for optimal vocabulary size and handling of rare/unknown words. At this stage, BERT's tokenizer also adds two special tokens, [CLS] ("classification") to annotate the start of a sequence and [SEP] ("separate") at the end of the sequence—

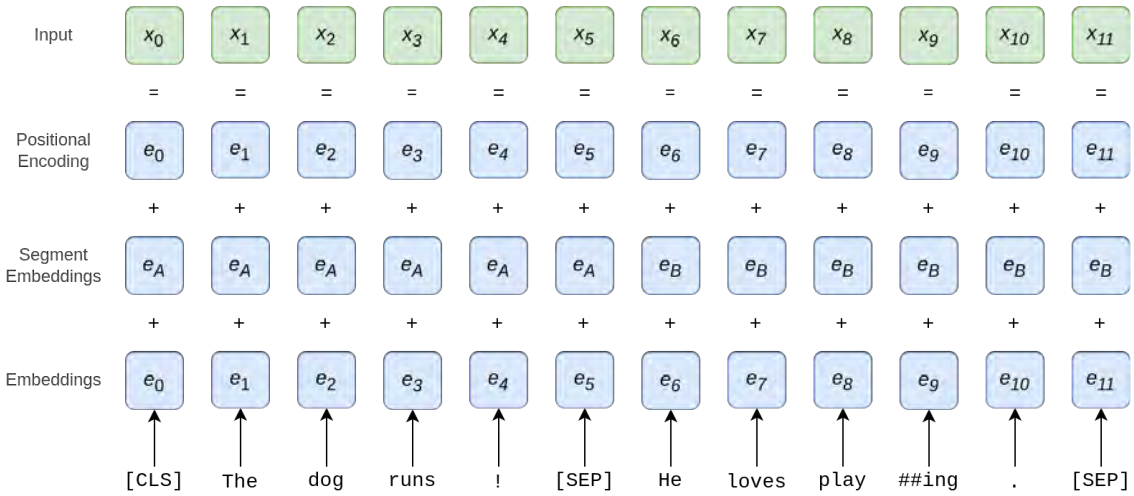


Figure 2.13: Visualization of converting the input into the embeddings layer and adding segment encoding and positional encoding to create the BERT input.

these tokens are necessary for the pretraining and finetuning processes when a sequence of multiple sentences is reduced to a single input vector. The original BERT pretrained word embeddings for English were built on a vocabulary of 30,522 words and subwords. The produced tokens are then encoded into a 768-dimensional vector representation, in a matrix of shape $(n, 768)$ (where n is the length of the input).

- The length of the input in BERT is fixed to the maximum length of 512; this means that longer sequences will be split and processed separately, and shorter sequences will have to be padded. In order to "block" the unnecessary padding special tokens from the attention mechanism (since they are not useful to the sequence), an additional layer called an **attention mask** may be used. It is a vector where useful tokens are mapped to 1 and tokens to be ignored are mapped to 0.

- The **segment embedding layer** is an additional method to separate the input's sentences, especially for classification tasks with two sequences. This layer only has 2 vector representations; the first vector (of index 0) is assigned to the tokens of the first sequence, and the second vector (of index 1) is assigned to the tokens of the second input. If there is no need to differentiate between the two sequences, the first vector can be assigned to all tokens.

- The **position embedding layer** serves the purpose of maintaining the positional in-

formation of the input sequence. They are positional embeddings calculated in the same way as in Vaswani et al. (2017)—see Section 2.4.3 and Equation 2.13. In a two-sentence input, the first sentence of length l will be modeled as $p(1), \ldots, p(l)$, and the second sentence of length j as $p(l+1), \ldots, p(l+j)$.

The representations of the three layers are summed element-wise to produce a single representation with shape $(1, n, d)$, which is then passed to the Encoder layer. The n variable is the maximum length of accepted input by the model, which in BERT's case is 512. d is the number of hidden states of the encoder (768 for base-sized models and 1024 for large-sized models). These representations can be added to a batch matrix of shape (b, n, d) (where b is the batch dimension).

The internal structure of BERT is a 12-layer Transformer encoder (24-layer for the large-sized models), with the architecture of the Vaswani et al. (2017) Transformer and 12-headed self-attention (or 16-headed for the large-sized models). A graphic demonstration can be seen in Figure 2.14.

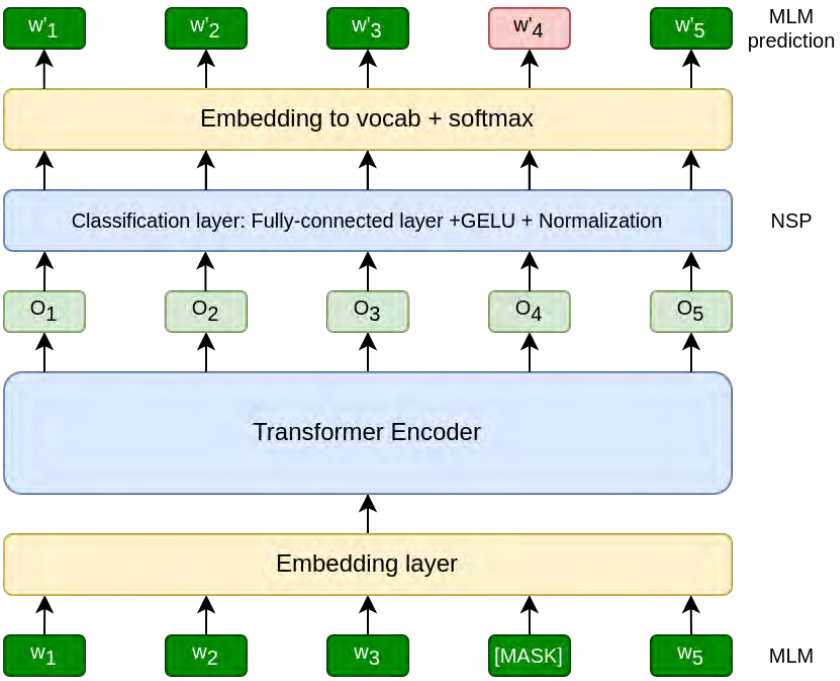


Figure 2.14: A visualization of BERT's pretraining process. In red is the prediction for the Masked Language Modeling task (based on the corresponding embedding output in the target position). The classification layer is responsible for the Next Sentence prediction task. Source: Horev (2018)

BERT was pretrained simultaneously on two training tasks: masked language modeling and next-sentence prediction. **Masked language modeling** (MLM) begins before the model is trained; 15% of the tokens of the input sequence are **masked**, i.e. replaced with the special [MASK] token, and the model is tasked to predict the original value of the masked token based on the context of the other non-masked tokens in the input sequence. This process involves the addition of a layer on top of the encoder output, multiplying the output vectors by the embedding matrix to transform them into the vocabulary dimension, and calculating the probability of each word in the vocabulary with the softmax function. While this method fits the description of an autoencoder (i.e. an architecture that creates input representations based on compressing structures observed in data) BERT differs in the sense that it only focuses on the prediction of masked tokens. Yang et al. (2019b) have described the BERT training process as "denoising autoencoding". Specifically, for a text sequence $\mathbf{x}$, BERT first constructs a corrupted version $\hat{\mathbf{x}}$ by randomly masking a portion of tokens in $\mathbf{x}$. The masked tokens are represented by $\overline{\mathbf{x}}$. The training objective is to reconstruct $\overline{\mathbf{x}}$ from $\hat{\mathbf{x}}$, as seen in Equation 2.24. $m_t = 1$ indicates x_t is masked, and H_θ is the Transformer block that maps a length- T text sequence $\mathbf{x}$ into a sequence of hidden vectors $H_\theta(\mathbf{x}) = [H_\theta(\mathbf{x})_1, H_\theta(\mathbf{x})_2, \cdots, H_\theta(\mathbf{x})_T]$.

$$\begin{aligned} \max_\theta \log p_\theta(\overline{\mathbf{x}} \mid \hat{\mathbf{x}}) &\approx \sum_{t=1}^{T} m_t \log p_\theta\left(x_t \mid \hat{\mathbf{x}}\right) \\ &= \sum_{t=1}^{T} m_t \log \frac{\exp\left(H_\theta(\hat{\mathbf{x}})_t^\top e\left(x_t\right)\right)}{\sum_{x'} \exp\left(H_\theta(\hat{\mathbf{x}})_t^\top e\left(x'\right)\right)} \end{aligned} \tag{2.24}$$

Next-sentence prediction (NSP) is a classification task in which the model is asked whether the second sentence of a two-sentence input is an appropriate continuation of the first one (as they appeared in the training set). The output vector of the special [CLS] token is used for the classification; it is passed to a single-layer feedforward neural network which is used as the classifier, and the result is a softmaxed probability.

The BERT architecture was originally trained with BookCorpus (Zhu et al., 2015) and the English Wikipedia (2.5 billion words), to produce (in the original paper) models with different parameter sizes: bert-base and bert-large (and each model having the variant

"cased" where the training set was unaltered, and "uncased" where it was lower-cased). The bert-base model has 12 layers, a hidden size of 768, and 12 attention heads for a total of 110 million hyperparameters, while bert-large has 24 layers, a hidden size of 1024, and 16 attention heads, for a total of 340 million hyperparameters. The size of the models is calculated not just by the size of the training set, but by the*hyperparameters* of the architecture. *Parameters* are the variables whose values are learned during training, but *hyperparameters* are the machine learning parameters whose value is determined before a learning algorithm is trained. For an architecture such as BERT, its hyperparameters are the variables that the architecture uses to configure the Transformer and multi-headed self-attention. These include the number of layers, heads, training size, learning rate, warmup steps, etc. A full list of the hyperparameters for the bert-base model is presented in Table 2.3, which also explains how the final number of hyperparameters is calculated.

These pretrained models can be used directly for NLP tasks (*feature-based approach*), achieving high accuracies without specialization being required (Peters et al., 2019). However, the models can also be finetuned on a specific task, with the use of one additional layer on top of the existing ones and a much smaller dataset than the pretraining one (according to Devlin et al. (2019), an annotated dataset of 100 thousand words is sufficient). Finetuning this additional layer takes 2-4 epochs, the whole process lasting a few hours on a GPU server compared to the days of pretraining on large processing units (e.g. around three days on 16 TPUv3 chips). Merchant et al. (2020) have shown that the update of weights is minimal on earlier layers of BERT and the update focuses on the last layers, therefore for most tasks and basic use there is no significant improvement with experimenting on freezing layers.

Soon after the open release of BERT by Google, multiple variations of word embeddings emerged; for example, Devlin et al. (2019) released, alongside the English BERT, multilingual BERT (mBERT), a BERT model pretrained on texts from multiple languages and sharing the same vocabulary. Multiple models in different languages appeared, either monolingual or multilingual, improving results in many tasks, especially in low-resource languages for which there were few available datasets and tools available (Wang et al., 2020b).

Embedding Matrices			
Word Embedding matrix size	Vocabulary size * embedding dimension	30,522 * 768	23,440,896
Position embedding matrix size	Maximum sequence length * embedding dimension	512 * 768	393,216
Segment (Token Type) Embedding	Matrix size	2 * 768	1,536
Embedding Layer Normalization	Weight + Bias	768 + 768	1,536
Total Embedding parameters			23,837,184 $\sim$ **24M**

Attention Head			
Query Weight	Matrix size (+bias)	768 * 64 + 768	49,920
Key Weight	Matrix size (+bias)	768 * 64 + 768	49,920
Value Weight	Matrix size (+bias)	768 * 64 + 768	49,920
Total parameters for the attention of one layer with 12 heads	No. attention heads * Sum of Query, Key and Value weights	12 * 3 (768 * 64 + 768)	1,797,120
Dense weight for projection after concatenation of heads	Weight + Bias	768^2 + 768	590,592
Layer Normalization	Weight + Bias	768 + 768	1,536
Position wise feedforward network weight matrices and bias	Weight + Bias [3072, 768]	2,359,296 + 3072 + 2,359,296 + 768	4,722,432
Layer Normalization	Weight + Bias	768 + 768	1,536
Total parameters for one attention layer		1,797,120 + 590,592 + 1536 + 4722432 + 1536	**7,113,216** $\sim$ **7M**
Total parameters for 12 layers of attention		12 * 7,113,216	**85,358,592** $\sim$ **85M**

Output layer of BERT Encoder			
Dense Weight Matrix and bias	Weight + Bias	768^2 + 768	590,592

Total Parameters in BERT-base	23,837,184 + 85,358,592 + 590,592	**109,786,368** $\sim$ **110M**

Table 2.3: How the number of parameters is calculated for the BERT-base model of Devlin et al. (2019).

2.5.5 RoBERTa

After the meteoric rise of BERT, there has also been a growing interest in how to adapt and improve the original architecture and implementation of Devlin et al. (2019) for faster and sturdier performance and higher accuracy. RoBERTa (Robustly Optimized BERT pretraining Approach) (Liu et al., 2019) is a popular successor to BERT, which uses the same architecture of BERT and improves on it by optimizing some training hyperparameters from Devlin et al. (2019): the learning rate, the warmup steps, and some optimizer biases. Additionally, they made small changes in the input and pretraining objectives. The RoBERTa architecture uses Byte-Pair Encoding (like GPT), yet increases the vocabulary size to 50,265 words and the batch size from 256 to 8,000. The input is passed as longer sequences, with the same token limitation as BERT (512 tokens), and there is no segment embedding layer, thus relying on special tokens (`</s>`) to denote the start and end of sequences. During pretraining, RoBERTa focuses only on the language masking modeling objective and has been trained with much larger mini-batches and learning rates. Unlike BERT, RoBERTa is trained only with full-length sequences[2]. Liu et al. (2019) also make use of *dynamic masking*; the masking pattern of 15% of the input tokens is different every time a sequence is fed to the model, as opposed to BERT which uses the same masking pattern. However, they have deemed that the Next Sentence Prediction task is not beneficial or stable as a pretraining objective.

For the English version, the training data was the same as BERT with an additional 144GB of text from Wikipedia, BookCorpus, CC-News (Mackenzie et al., 2020) and various annotated datasets for NLP (for a total of 160GB of data). The original release included two models, both exclusively lower-cased: roberta-base has 12 layers, hidden size of 768, 12 attention heads, and 125M parameters while roberta-large has 24 layers, hidden size of 1,024, 16 attention heads, and 355M parameters. The authors of RoBERTa claim that, at the time of its release, their model and embeddings achieved better downstream task performance compared to BERT, as a combination of its hyperparameter optimization and its longer and larger pretraining process. It has been tested on standardized benchmark tasks for NLP that examine complex linguistic phenomena and measure

[2]BERT was trained by randomly injecting short sequences, and with a reduced sequence length for the first 90% of updates.

success based on quantitative accuracy metrics, such as the General Language Understanding Evaluation (GLUE) benchmark that examines language understanding and linguistic competencies (Wang et al., 2018), and tasks such as natural language inference, textual entailment, and reading comprehension. At the time of its release, RoBERTa had achieved state-of-the-art for many of these benchmarks.

2.5.6 XLNet

XLNet (Yang et al., 2019b) is an architecture that uses autoregression similar to GPT, alongside BERT's bidirectional motivation, as a means to tackle the limitations of both architectures.

XLNet employs the *segment-level recurrence* mechanism and *relative positional encodings*, as introduced in Transformer-XL, a predecessor to XLNet (Dai et al., 2019). The representations calculated for a segment t are fixed and cached for later usage as the extended context for segment $t+1$. By allowing contextual information to now traverse segment borders, it is possible to extend the dependency length by N times, where N is the depth of the network, thus allowing more contextual information through the model. This technique may tackle BERT's independence assumption, i.e. that a masked token is only dependent on non-masked tokens and not other masked tokens, and that masked tokens are equally dependent on all the non-masked tokens (Yang and Le, 2019). An example from Yang and Le (2019) that illustrates the benefit of segment-level recurrence is the prediction of "New" in the masked sentence "[MASK] York is a city.". XLNet is capable of capturing the dependency pair "New York", while BERT will not model these words as a pair.

XLNet introduces *permutation language modeling*; by using a permutation operation during training time, the model is capable of capturing bidirectional context. For a sequence X of length T, there are $T!$ possible orders for a valid autoregressive factorization, which are permutated while the model traverses through sentences. A simplified example of the permutation language modeling objective is shown in Table 2.4: all the possible permutations for a sequence of length 3 (with index *0* being the model's memory of learned hidden states). If the model is asked to predict the token "dogs" (token with index 3), the model will observe the possible contexts and predict based on those

(or rely only on past knowledge, if there is no available context). With this objective, the model is able to efficiently capture a token's relations with the rest of the input tokens, regardless of distance.

Sentence	Permutation order	Permutated tokens	Seen tokens for idx=3
"I like dogs"	0,1,2,3	I like dogs	0, I, like
	0,1,3,2	I dogs like	0, I
	0,2,1,3	like I dogs	0, like, I
	0,2,3,1	like dogs I	0, like
	0,3,1,2	dogs I like	0
	0,3,2,1	dogs like I	0

Table 2.4: The input sequence "I like dogs" and its possible permutations by the XLNet architecture. When the word "dogs" is masked to be predicted, the model will learn the 3! permutated contexts.

A formalization of permutation language modeling can be found in Equation 2.25. For an input $\mathbf{x}$, a permutation order $\mathbf{z}$ is created, and the likelihood $p_\theta(\mathbf{x})$ according to the order. With all the possible permutation orders, x_t will have seen every possible element $x_i \neq x_t$ in the sequence, thus ensuring bidirectionality. θ is the model parameter that maximizes the expected value of the permutation order. $\mathcal{Z}_T$ is the set of all possible permutations of a sequence with length T. The current t-th token is z_t and the previous tokens of the permutation order $\mathbf{z}$ are $\mathbf{z}_{<t}$.

$$\max_{\theta} \quad \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\sum_{t=1}^{T}\log p_\theta\left(x_{z_t} \mid \mathbf{x}_{\mathbf{z}_{<t}}\right)\right] \tag{2.25}$$

The permutated order operations can be computationally costly, therefore the model does *partial predictions*, meaning that the model outputs predictions only for the last tokens in the factorization order with a cut-off point (He, 2020). The choice of tokens to predict is shown in Equation 2.26. A hyperparameter K is used such that about $1/K$ tokens are selected for predictions; i.e., $|\mathbf{z}|/(|\mathbf{z}| - c) \approx K$. In this equation, $\mathbf{z}_{>c}$ is chosen to be predicted, because it possesses the longest context in the sequence given the factorization order $\mathbf{z}$. The model does not compute query representations for unselected tokens (Yang et al., 2019b).

$$\max_{\theta} \quad \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\log p_\theta\left(\mathbf{x}_{\mathbf{z}_{>c}} \mid \mathbf{x}_{\mathbf{z}_{\leq c}}\right)\right] = \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\sum_{t=c+1}^{|\mathbf{z}|} \log p_\theta\left(x_{z_t} \mid \mathbf{x}_{\mathbf{z}_{<t}}\right)\right] \tag{2.26}$$

In order to combine permutated order with the segment-level occurrence, XLNet uses fixed embeddings with learnable transformations rather than learnable embeddings, this being able to capture longer sequences than fixed-size encodings.

XLNet utilizes *two-stream self-attention* (see Figure 2.15) in order to be able to attend to words in both directions, even if these words were not in the past of the token in the permutated word order.

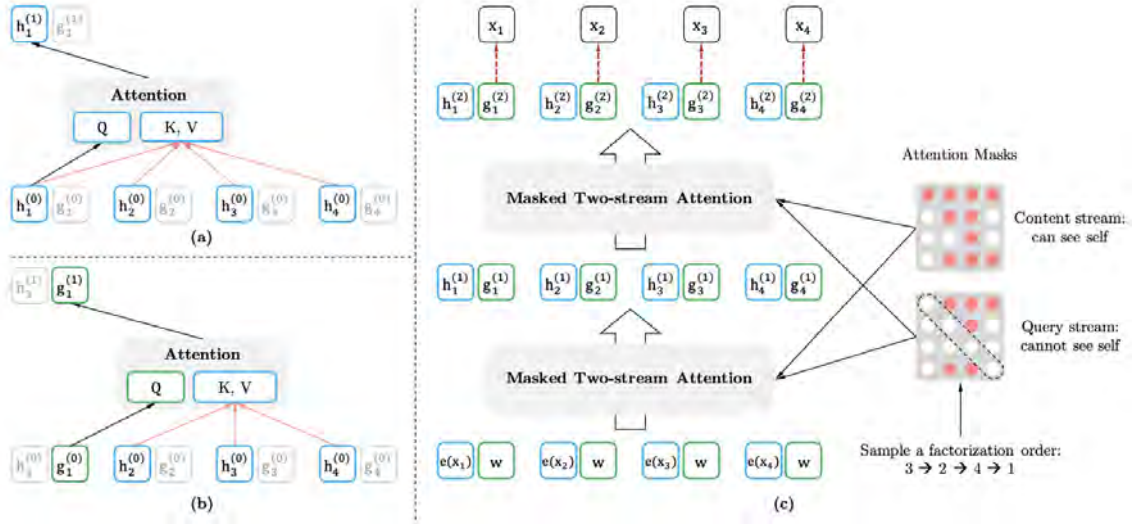


Figure 2.15: (a) Content stream attention, (b) Query stream attention, and (c) the two-stream self-attention. Source: Yang et al. (2019b)

The content representation $h_\theta\left(\mathbf{x}_{\mathbf{z}_{\leq t}}\right)$, or abbreviated as h_{z_t} is similar to the hidden states in a Transformer encoder. This representation encodes both the context and x_{z_t} itself. Meanwhile, the query representation $g_\theta\left(\mathbf{x}_{\mathbf{z}_{<t}}, z_t\right)$, or abbreviated as g_{z_t}only has access to the contextual information $\mathbf{x}_{\mathbf{z}_{<t}}$ and the position z_t, but not the content x_{z_t}. Thus, the model is able to encode both positional-sensitive and positional-independent contextual information. The calculations for the content representation are seen in Equation 2.27 and for the query representations in Equation 2.28.

$$h_{z_t}^{(m)} \leftarrow \text{Attention}\left(\mathbf{Q} = h_{z_t}^{(m-1)}, \text{KV} = \mathbf{h}_{\mathbf{z}_{\leq t}}^{(m-1)}; \theta\right) \tag{2.27}$$

$$g_{z_t}^{(m)} \leftarrow \text{Attention}\left(\mathbf{Q} = g_{z_t}^{(m-1)}, \text{KV} = \mathbf{h}_{\mathbf{z}_{<t}}^{(m-1)}; \theta\right) \tag{2.28}$$

The pretrained XLNet embeddings were trained on 32.89 billion words from various corpora: Wikipedia, BooksCorpus, English Gigaword Fifth Edition[3], ClueWeb 2012-B[4], and Common Crawl[5]. XLNet also uses the SentencePiece algorithm for tokenization. The 12-layer 12-attention head XLNet-base with a hidden size of 768 was trained only on the first two corpora for a total of 110 million hyperparameters, while the 24-layer 16-attention head XLNET-large has a hidden size of 1,024 and a total of 340M hyperparameters. According to Yang et al. (2019b), the benefits of autoregression and autoencoding set XLNet ahead of BERT on benchmark tasks such as question answering, natural language inference, sentiment analysis, and document ranking.

2.5.7 ALBERT

ALBERT (A Lite BERT) (Lan et al., 2020) is another successor to BERT, with the motivation to create an architecture that is smaller and more efficient; a smaller model can be trained and finetuned without the need for large processing units and high energy consumption. ALBERT uses 18x fewer parameters than BERT and can be trained 1.7 times faster, without a trade-off in performance. It keeps the masked language modeling objective, accompanied by the *sentence order prediction* classification task. ALBERT uses *factorized embedding parameterization*, a process that divides the embedding matrix into two halves: the vocabulary embeddings E and the hidden layer embeddings H. The vocabulary embeddings maintain context-independent token representations, while the hidden layer embeddings learn context-dependent representations through pretraining. In a model like BERT and RoBERTa, the embedding parameters have a size of $(V \times H)$, while in ALBERT, the embedding parameters are reduced to $(V \times E + E \times H)$.

[3]https://catalog.ldc.upenn.edu/LDC2011T07
[4]https://lemurproject.org/
[5]https://commoncrawl.org/

ALBERT's input includes the relative positional encoding vectors and the segment encoding vectors. After decomposing the embedding matrix, ALBERT applies a linear fully connected layer to the embedding matrices in order to map its dimensions to the hidden layer.

Additionally, with *cross-layer parameter sharing*, the encoder blocks of ALBERT are able to share weights with each other, thus do not have to calculate them individually, a process which helps with computations and also with regularization of the model (Wright, 2019). Finally, ALBERT removes dropout layers, i.e. does not randomly ignore some neurons, relying on the Transformer's parallelization skills to avoid overfitting. Furthermore, following the example of Liu et al. (2019) and Yang et al. (2019b), ALBERT does not include the Next Sentence Prediction classification task.

ALBERT was pretrained on Wikipedia and BookCorpus, as BERT was, and uses the SentencePiece algorithm (Kudo and Richardson, 2018) for tokenization. The second version of the pretrained embeddings released included no dropout, additional training data, and longer training, and came in four different training sizes (with the cased/uncased variations):

- albert-base-v2 (12 layers, embedding size of 128, hidden size of 768, 12 attention heads, 11M parameters)
- albert-large-v2 (24 layers, embedding size of 128, hidden size of 1024, 16 attention heads, 17M parameters)
- albert-xlarge-v2 (24 repeating layers, embedding size of 128, hidden size of 2048, 16 attention heads, 58M parameters)
- albert-xxlarge-v2 (12 layers, embedding size of 128, hidden size of 4096, 64 attention heads, 223M parameters)

At the time of its release, ALBERT outperformed BERT on NLP benchmarks such as textual entailment and reading comprehension, proving that better exploitation of contextual representations could be more beneficial than larger training and parameter sizes.

2.5.8 CamemBERT

CamemBERT (Martin et al., 2020) is a monolingual Transformer-based deep contextualized word embedding model, built with the RoBERTa architecture described in Section 2.5.5. The model uses SentencePiece for tokenization and employs Whole Word Masking, in which the model has to predict a word instead of a subword artifact (Joshi et al., 2020). As the RoBERTa architecture does, CamemBERT uses dynamic masking for 15% of the input, and only uses the Masked Language Modeling task for pretraining. The pretrained embeddings are based on the OSCAR corpus (Ortiz Suárez et al., 2019), the French version of CommonCrawl, and Wikipedia. There are several models (exclusively lowercased) available: the base models have 12 layers, a hidden size of 768, 12 attention heads, and a total of 110M hyperparameters while the large models have 24 layers, hidden size of 1,024, 16 attention heads, and 335M parameters. Some of the base models were also trained only with a subset of the training datasets (4GB of data instead of 135GB), and the authors did not notice a deterioration in results.

At the time of its release, the only deep contextualized word embeddings came from multilingual models that included French, e.g. mBERT (Devlin et al., 2019), $\text{XML}_{\text{MLM-TLM}}$ (Conneau and Lample, 2019). CamemBERT managed to significantly outperform them in tasks of POS tagging, dependency parsing, named entity recognition, and natural language inference.

2.5.9 FlauBERT

French Language Understanding via Bidirectional Encoder Representations from Transformers (FlauBERT) (Le et al., 2020) was the second monolingual French model to be released, up to this day. The FlauBERT model is based on BERT, but uses only its pretraining task of masked language modeling; it has already been observed from previous architectures that the next sentence prediction task does not affect performance on downstream tasks. FlauBERT uses the Moses tokenizer (Koehn et al., 2007) and sources data from multiple datasets, mainly Wikimedia projects, French text corpora offered in the OPUS collection[6] and monolingual data for French provided in WMT19 shared

[6] https://opus.nlpl.eu/

tasks[7]. They have released 3 different sizes of their models, with the base model being also available in lower-cased: flaubert-small with 6 layers, 8 attention heads, hidden size of 512 and 54M parameters, flaubert-base with 12 layers, 12 attention heads, hidden size of 768 and 137M parameters, and flaubert-large with 24 layers, 16 attention heads, hidden size of 1024 and 373M parameters.

Alongside their architecture, the authors introduced FLUE (French Language Understanding Evaluation), a French benchmark of various tasks (text classification, constituency parsing and part-of-speech tagging, dependency parsing, word sense disambiguation, paraphrasing, natural language inference). They reported that FlauBERT models were marginally better than CamemBERT in most of these tasks, with CamemBERT having very similar accuracies and the multilingual embeddings only outperforming them in natural language inference. In Table 2.5 we are comparing the parameters of the two architectures.

	CamemBERT	**FlauBERT**
Train size	138 GB	71 GB
Pretraining objectives	MLM	MLM
Parameters (base-large)	110 M / 335 M	138 M/ 373 M
Tokenizer	SentencePiece 32K	BPE 50K
Masking strategy	Dynamic & Whole word masking	Dynamic & Sub word masking

Table 2.5: A comparison of the CamemBERT and FlauBERT architectures, from Le et al. (2020).

[7]https://www.statmt.org/wmt19/translation-task.html

Chapter 3

Explainability of Transformer-based architectures

3.1 Introduction

Transformer models have overtaken the field of NLP since their contextual word embeddings have proven to yield high accuracies in multiple tasks, some of which were difficult to tackle with traditional machine learning methods. These models quickly outperformed traditional methods and neural network approaches in standardized tests called *benchmarks*; these tests measure accuracy on an NLP task (e.g. machine translation) with a predetermined dataset and accuracy metrics and are widely accepted as proof of competence in NLP applications. However, success in a benchmark does not guarantee a complete mastery of the given task. The benchmarks themselves tend to be short-lived and (understandably) very limited in the scope of the phenomena and datasets they can test (Srivastava et al., 2022). Unfortunately, in recent years there have been overzealous attempts to create systems that achieve a high ranking in benchmark scores, without a guarantee that the winning models will be successful on this task in a different setting or with a different dataset. Based on the reported high accuracies in many NLP tasks, these models have also been released for public and commercial use, without a true understanding of their capacities, limitations, and potential dangers (from the perpetuation of harmful stereotypes to their considerable carbon footprint) (Bender et al., 2021).

Hence, it is paramount to understand and interpret how Transformer models achieve such good performances; can these models learn linguistic structures (e.g. agreement, dependency structure), or is their performance based on skillful exploitation of artifacts

in their massive training data (Bender and Koller, 2020)? Fortunately, the scientific community is conscious of these shortcomings and aims to explore the capabilities of these models, even though the increasingly complex architectures become harder to decipher. The interest in understanding and interpreting these powerful models is so widespread that the study on explainability has been unofficially named "BERTology", made popular after the paper of Rogers et al. (2021).

Explaining a neural architecture model can either be a process of introspection or generalization. Insights into how a specific model processes information may help improve a model's performance, by changing factors such as the number of labeled data, the value of the hyperparameters, and the model selection. On the other hand, generalizing sheds light on model predictions; the goal is to explain, typically in terms of model input, why a certain prediction was generated by the model and ideally identify patterns of behavior in the model's predictions.

3.2 Linguistic Evaluation & Explainability

3.2.1 Probing methodology

In order to perform linguistic analysis, a combination of probing methods and attention analyses have been proposed. They permit us to observe how much linguistic information and of what type has been learned by a model. When possible, the scientific community opts for qualitative analysis, either by creating specialized datasets for probing or by examining specific phenomena and sentences.

A method of studying the capabilities of a model is with *probing*. In NLP research, probing methods that use the encoded representations of one system to train another classifier on a different task of interest. These classifiers are also known as *diagnostic classifiers* (Hupkes et al., 2018). The probing tasks are usually linguistically motivated and focus on simple and complex linguistic properties of a predefined input. They may involve surface-level features, syntactic information, semantic information, etc. ideally isolated from structures and phenomena that could interfere with the study. If the probing classifier performs well on the probing task, it is implied that the system has encoded the linguistic phenomena in question (Conneau et al., 2018).

For example, Shi et al. (2016) used the encoder input of a system performing neural machine translation to train a logistic regression model to perform syntactic labeling and reported positive findings. Ettinger et al. (2016) proposed a classification task using Word2Vec embeddings and prompt sentences with semantic differences that are easily distinguishable for humans but were occasionally confusing to embedding models (e.g. a non-animate verb subject as agent). Gulordava et al. (2018) experimented with embeddings generated from RNN architectures and their ability to predict long-distance number agreement. They concluded that the models were not relying only on frequent morphosyntactic sequences, but were able to construct patterns akin to syntactic structure. Giulianelli et al. (2018) also explore LSTM embeddings for their knowledge of subject-verb number agreement and observed that this information is learned by the model dynamically, in each timestep of the learning process. Zhu et al. (2018) create sentence-level embeddings by averaging static word embeddings and reported that the classifying process was able to distinguish between negation and synonymy, but not between synonymy with different word order.

For transformer models, probing is performed on pretrained models for a downstream task, e.g. natural language inference, or with multiple training objectives like BERT. The goal is to observe if the pretraining process suffices to capture enough linguistic information to show a degree of syntactic and semantic competence (Kim et al., 2019). An important point by Hewitt and Liang (2019) is that the probing process should not necessarily aim for high accuracy, but for providing linguistic insights, with simple (for the intended observation) yet efficient probes. They are critical of probing tasks, claiming that it is possible that the target observation is learned during classifying and not truly encoded in the source model.

3.2.2 Assessing Transformer models' linguistic knowledge

Raganato and Tiedemann (2018) were some of the first to investigate encoder representations and the attention mechanism of a Transformer model (trained for the task of Machine Translation) for its learned information. They observed that there is some specialization on syntactic information by some attention heads and that lower layers tend to syntactic information and higher levels to semantic information. Goldberg (2019) has

found that BERT (a Transformer model) is more robust in syntactic tasks than a simple LSTM architecture (a Recurrent Neural Network). With a series of probing tasks on different datasets, they proved that there is some syntactic knowledge beyond semantic and contextual relations, in subject-verb pairings. According to its creators, BERT performed better on syntactic tasks, compared to older neural network models and other Transformer architectures, because it was able to avoid distractors (Wolf, 2019).

Further research on learned syntactic information showed that BERT captures different types of information on different levels. Jawahar et al. (2019) claim that BERT captures phrase-level information in the early layers, surface levels, syntactic and semantic features at the middle layers, and makes use of the final layers to track long-distance dependencies. Coenen et al. (2019) found that the attention matrices output by bert-base-uncased contain syntactic representations, with certain directions in space representing specific relations, and they were also able to locate similar sub-spaces for semantic relations. Ravishankar et al. (2021) focus on attention heads in multilingual BERT and observe that even single attention heads are able to recreate dependency syntax tree structures (a fining also found in Jawahar et al. (2019)) and that frozen models demonstrate interesting attention patterns akin to linguistic structures.

The specialization of layers is also reported in Vig and Belinkov (2019), who analyzed GPT-2 and observed that attention heads show a great variety of attention to tokens in different layers and heads. They also report that, in middle layers, attention may follow dependency relations, and attention heads focus on different parts of speech. Petroni et al. (2019) report that BERT contains enough relational knowledge to compete with knowledge-based methods on tasks such as open-type questions, which leads them to the conclusion that the model has acquired a certain level of semantic knowledge.

However, McCoy et al. (2019) question the ability of BERT—and similar pretrained models—to truly capture deep linguistic structures and semantic information, as past bibliography has suggested. Tenney et al. (2019) also investigated pretrained models on their performance on both syntactic and semantic phenomena. They concluded that simple syntactic phenomena were successfully identified, but phenomena that primarily relied on semantic relations were not as easily learned. Ettinger (2020) presents a number of experiments on syntactic-semantic knowledge, where in many cases BERT

makes good predictions with regard to semantic pertinence, such as hypernyms and subject-object nouns. However, they are critical of the semantic competencies of BERT compared to human performance and highlight that the model does not perform well with truth statements and negations. Yu and Ettinger (2020) examined contextual word embedding models against human evaluations of phrase similarity and meaning shift. They compare the findings before and after controlling for word overlap, in order to distinguish between lexical effects and composition effects. They claim that there is minimal indication of nuanced composition in these models' phrase representation and that the representation mainly depends on the content of the context words. They examine several models and discover that, when using the sentence as input, the middle layers of most architectures offer the most accurate predictions. They also propose that the use of an averaged embedding of a sentence's embeddings offers a better sentence representation than the use of special tokens (e.g. [CLS] for BERT). Zhang et al. (2019c) created SemBERT, a BERT model with integrated explicit contextual semantics, supporting the fact that external semantic knowledge was more useful than manipulating inherent model knowledge to achieve better results in semantics-related tasks. Mickus et al. (2020) delve further into exploring the embeddings of BERT and report that it is uncertain whether the embeddings are able to properly represent semantic similarities on a word-base level (as the theory of distributional semantics would suggest), due to the influence of the context sentence on the distributional semantics space (even without meaning correlates).

Lasri et al. (2022) examine whether BERT can make correct number agreement associations, between a verb and its nominal subject in English, in many variations of syntactic structures. They claim that BERT may favor meaningful lexical combinations because they are more frequent. Weissweiler et al. (2022) study how pretrained language models treat sentences with comparative correlative in English, a construction that requires syntactic and semantic competencies to be understood. The models were able to recognize sentences as examples of the construction, even in challenging situations, suggesting that the syntactic aspect is captured in pretraining. However, they showed weaknesses in comprehending the sentence's meaning and using its context for complex processes as inference, thus demonstrating weakness in semantic-driven tasks.

Kim et al. (2022) focus on the treatment of auxiliary clauses and the phenomenon of ellipsis and note that the models show sensitivity to the connection of auxiliary verbs to the main clause, but this preference stems from superficial frequency factors rather than principled discourse rules.

3.2.3 Self-attention and psycholinguistics

The interpretability of attention as an analogy to human attention has also been exploited by literature; Chrupała and Alishahi (2019) use *Representational Similarity Analysis* to correlate the way a Transformer model encodes and processes representations to the way humans process a sentence, and found some modest connections. Abdou et al. (2019) conduct similar experiments, with the addition of gaze fixation as a measure of human attention. Brandl et al. (2022) continue experiments with the mean attention vector of the final layer heads of Transformer models and with psycholinguistics methods of measuring human gaze, and observed a high correlation between human- and machine-highly attended tokens. However, human attention is not an infallible measure of human language processing, as the ways of measuring human reaction can—and probably are—simultaneously affected by biological, psychological, environmental, and other linguistic processes (Lindsay, 2020).

Chang and Bergen (2021) conduct a study of neural networks and transformer architectures, BERT and GPT-2, inspired by language acquisition research; they draw similarities between the frequency-based learning of language models and the chronological stages of language acquisition in humans, for which frequency of words and features are important factors—but not the driving force. They identified similarities in the learning of frequent word forms during the training process of models (by examining word surprisal) and during human acquisition.

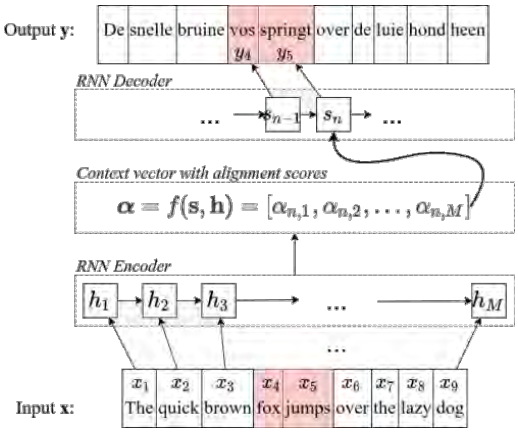
3.3 Interpretability of self-attention

3.3.1 Is self-attention explanation?

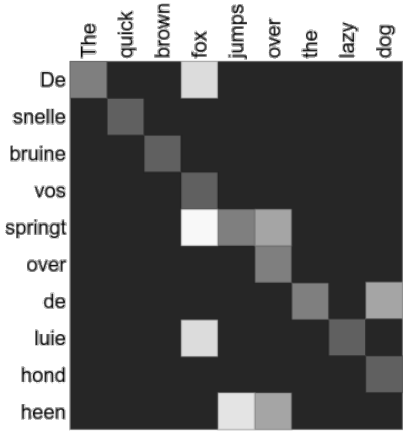
Neural networks use sub-symbolic structures, meaning that the information they acquire is kept in numerical elements that cannot be interpreted on their own, making it difficult, if not impossible, to identify the causes of a neural architecture's output. The attention mechanism not only improves performance but also serves as a tool for deciphering the behavior of neural architectures, which is notoriously challenging to do. Even if it cannot be regarded as a trustworthy method of explanation (Jain and Wallace, 2019), attention may be an interesting way to partially analyze and explain neural network behavior (Guidotti et al., 2018; Wiegreffe and Pinter, 2019). For example, the weights calculated by attention could point us to important data that the neural network missed or unimportant components of the input source that have been taken into account and could explain an unexpected result of the neural network (Galassi et al., 2020).

A commonly used method to explain the behavior of the attention mechanism is visualization with grid *heatmaps*, which originated from data science to summarize findings and main components in data. An example of how Bahdanau et al. attention can be visualized is shown in Figure 3.1, which presents an example of how attention is visualized, for an RNN neural network with Bahdanau et al. attention, performing neural machine translation of English to Dutch. In the attention mechanism, the alignment score function f produces the alignment scores α (i.e. the attention weights), which align the different parts of the input and output. These alignment scores can be directly used to create a heatmap; in black-and-white heatmaps, lighter colors correspond to higher alignment scores and darker to lower attention. Through this matrix, it is possible to observe that the attention mechanism focuses mostly on each input token's direct translation, while also capturing some syntactic relations (verb-subject, noun-determiner).

As explained in Section 2.4.2, the internal structure of a Transformer architecture is based on parallel operations, multiple layers, and self-attention. The high performance in multiple NLP tasks is attributed to the use of the self-attention mechanism, a structure far more complex than traditional attention mechanisms previously used in NLP. Extensive research and discussions have been carried out, to assess whether these self-



High-level overview of Bahdanau et al. attention



Attention heatmap

Figure 3.1: An example of attention as interpretation in neural machine translation, in a neural network with Bahdanau et al. (2014) attention for English-Dutch translation. Source: Ras et al. (2022)

attention mechanisms are interpretable, that is, whether they produce results—correct or incorrect—which can be traced back to the way they tend to the input sequence. This task is not as straightforward as with traditional neural architectures; multi-headed attention in a multi-layer model means that every head, in every layer, computes its own weights and attends to input in a unique way.

Jain and Wallace (2019) investigated whether the input tokens which were attributed the highest attention weights were, in fact, the most important tokens of the input and for the output. They conducted experiments on the correlation between input attention weights and feature importance methods (gradient-based and leave-one-out) and on prediction outcomes by randomly (and adversarially) shuffling attention weights. They observed that there are no correlations between the assigned attention weights and the chosen explanation methods and that the models' output was not affected even by shuffled attention weights. Serrano and Smith (2019) conducted similar experiments to Jain and Wallace (2019), by removing attention weights of the input sequence and observing the results of the text classification models, and noted that there were some cases where attention to the most important input constituents was important, e.g. the other binary label was predicted. However, they do not conclude with solid and definitive evidence about the attention's role and interpretability. Michel et al. (2019) conducted pruning ex-

periments on machine translation and multi-headed attention and stated that big parts of the network can be removed and some layers may be reduced to one attention head, without loss in performance. This means that the influence of certain attention heads on the output sequence may be more significant than other heads.

However, Wiegreffe and Pinter (2019) recreated the experiments of Jain and Wallace (2019)'s and claimed that permutations of attention weights are not an efficient way to test attention. Their experiments showed that an adversarial permutation of attention weights is, in fact, detrimental to model performance. Also, shuffling attention weights—in an inconsistent way—cannot lead to truly meaningful insights on the role of attention. Brunner et al. (2019) have explored whether the final attention weights of a model using self-attention can offer interpretations of the relations of input-output tokens. They observed that there is the possibility that different variations of optimized attention weights could produce the same output since the inner workings of the attention mechanism do not treat tokens and input in an identifiable, analogous way. They propose that the study of attention can be fruitful, with their method of removing the attention weights that do not influence the model's predictions (*effective attention*). Kobayashi et al. (2020) support the findings of Brunner et al. (2019) by examining BERT's embeddings, and they observe that meaningful alignments are created between the inputs and the output tokens in the pretraining process. However, they do support that attention vectors alone are not an adequate explanation, since, at first glance, the model's attention does not focus on important tokens. They suggest that attention vectors should be studied alongside the transformed input vectors, in order to gain insight into the behavior of Transformer-based models.

Vashishth et al. (2020) added that attention may be influential and interpretable for specific tasks, such as natural language inference and neural machine translation, but trivial for other tasks such as text classification (which was used by Jain and Wallace (2019) and by Wiegreffe and Pinter (2019)). Bastings and Filippova (2020) also agree that using attention as an explanation might not be the best course of action to interpret the inner workings of Transformer models on various NLP tasks. They examine gradient-based, propagation-based, and occlusion-based methods, and observe that these models are more efficient in identifying the important parts of the input sequence. Galassi et al.

(2020) propose that attention can be explainable and interpretable, but its study should be more thorough with a combination of input analysis, and a clear distinction of the methodology in which it may be extracted from the model, as the bibliography might not be consistent.

In their study, Vashishth et al. (2020) used human evaluation to assess whether the top three attention-weighted words were in fact the most significant of the sequence, introducing the notion of *plausibility* of model outputs as a correlation of human performance/assessment. However, Jacovi and Goldberg (2020) are opposed to human evaluation and rephrase the quest for interpretability of attention as an *evaluation of faithfulness*, i.e. whether a model's output can be interpreted based on the models' decisions. They claim that the criteria of evaluating attention for its contributions and choices, framed as *plausibility*, can be unreliable and produce anecdotal interpretations of a model's inner workings. They support that the success of a model should be measured by the stability and reproducibility of its results, rather than human evaluation and comparison.

Bibal et al. (2022) conducted extensive and exhaustive bibliographic research on findings and methods of evaluating self-attention, and they concluded that methods that combine the analysis of attention with proactive methods of selecting only relevant weights, such as *effective attention* (Brunner et al., 2019; Kobayashi et al., 2020), can potentially make self-attention more interpretable.

3.3.2 Visualizing self-attention for interpretation

As previously discussed, visualization can be a powerful tool for examining the strengths and weaknesses of models, however, visualizing multi-headed attention over several layers is a challenging feat. Vig (2019) aims to explain attention in a Transformer model with a multi-scale visualization tool that can visualize attention per attention head, per model, and per neuron. This tool follows the traditional methods of visualizing attention in neural networks while adapting to the challenge of visualizing the deep architecture of Transformer models with multiple layers and heads. Clark et al. (2019) have also developed a visualization tool for Transformer models, with many visualizations of attention weights per attention head and entropy changes in attention, as commonly studied in

neural machine translation interpretation. Their tool focuses on pretrained BERT models. An example from their work can be seen in Figure 3.2, in which they load pretrained embeddings and compute attention head entropies for each token to the other tokens of the sequence (including itself). Instead of creating a heatmap, they are using lines with varying thicknesses, to show how each token attends to the other tokens in a sentence. They create a separate visualization, for each head in one layer, in order to study each attention head's weights and how they evolve.

Abnar and Zuidema (2020) visualize attention by using *attention flow* (by treating the attention graphs as flow networks) and *attention rollout* (by computing the amount of information that is propagated from each node). They support that this work can provide better insights on the distribution of attention, since it is challenging to follow in multi-layer, multi-head models, and because deeper layers and heads are more contextualized and may all carry similar information.

Hoover et al. (2020) developed exBERT, a tool that visualizes BERT's token-level attention with a wider context scope. Alongside the attention heatmap visualization per head, it also displays linguistic metadata for a selected token. The tool retrieves the sentences of the training datasets of BERT that contain the selected token, thus providing more insight into how the contextual information was captured by the model, given various inputs and contexts. The evolution of this tool, LMdiff (Strobelt et al., 2021), is capable of examining and comparing multiple models' outputs on a sentence at the same time, whether they are pretrained, finetuned, distilled, or have varying hyperparameters.

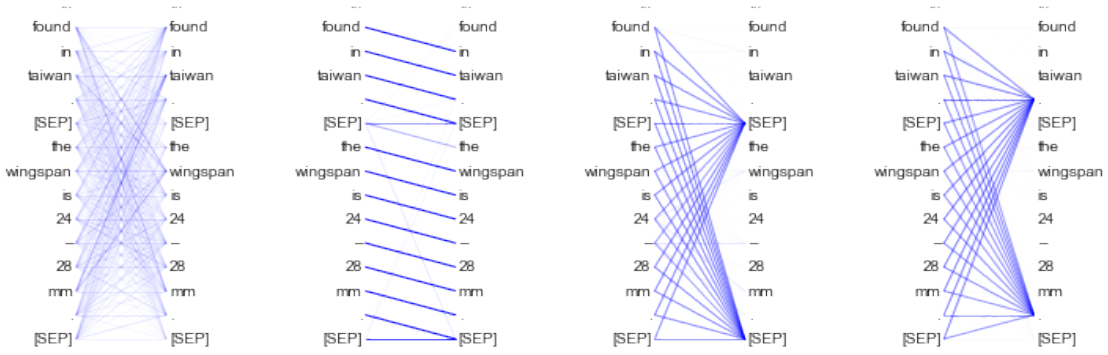


Figure 3.2: A visualization produced by the tool of Clark et al. (2019). Source: Clark et al. (2019)

Chapter 4

Selectional preferences in contextual word embeddings

4.1 Introduction

The first experiment we planned and conducted focused on the BERT transformer architecture. The linguistic knowledge of BERT in multiple NLP applications has been probed with methods traditionally used in attention-based neural networks, as discussed in Section 3.2. Our contribution to these endeavors is the investigation of knowledge on a syntactic and semantic level; relevant work supported the presence of syntactic competencies, but little was known about semantics.

In order to examine both syntactic and semantic abilities, we used the *selectional preferences* of a word, i.e. the type of arguments and meanings a word prefers to be related with. We investigated whether BERT embeddings contain information on the selectional preferences of words, by examining the probability it assigns to the dependent word, given the presence of its head word in a sentence. In early experiments, we examined the probability itself, with quantitative and qualitative methods; however, this does not suffice in the study of selectional preferences, since they are quite complex to determine. Even in literature, linguists cannot fully define selectional preferences in every possible context and language use by native speakers, and have made use of computational methods for assistance in selectional preference induction. A proper understanding of this phenomenon is important within various NLP applications, and selectional preferences have indeed been used as an additional knowledge source for various NLP tasks, such as word sense disambiguation (McCarthy and Carroll, 2003) and semantic role labeling

(Gildea and Jurafsky, 2002).

For our experiments, we made use of an existing dataset of selectional preference, which has been annotated by humans on word pair preferences (Zhang et al., 2019b). Word pairs are annotated with an average plausibility score, an analogy to the felicity of the head word choosing a word as its argument. We used word pairs of head-dependent words in five different syntactic relations from the SP-10K corpus of selectional preference, as found in real sentences from the ukWaC corpus (Ferraresi et al., 2008). We calculated the correlation of the plausibility score and the model's assigned probabilities for the dependent word, as retrieved by the masked language modeling version of bert-base-uncased.

Our results show that overall, there is no strong positive or negative correlation in any of the proposed syntactic relations. However, we do find that certain head words have a strong correlation. Additional experiments with attention masks (on the self-attention mechanism of BERT) showed that masking all words but the head word yields the most positive correlations in most scenarios, which indicates that the semantics of the predicate is indeed an integral and influential factor for the selection of the argument.

4.2 Linguistic background

Native speakers have structural preferences and constraints on how to speak and write their language, which they learn during language acquisition and later on may potentially influence or adapt to their needs (while maintaining mutual understanding in their communities). The conventions that have been established as a way to model language, at a specific time and for a specific speech variety (dialect, idiom), are called the *grammar* of the language[1]. The notion of *grammaticality* explains the creation of utterances that are well-formed and adhere to the rules of the native speakers' grammar, and sentences that do not follow these rules are deemed ungrammatical (Fromkin et al., 2013). For example, as seen in Table 4.1, grammatical sentences in English are those that follow the syntactic rules of subject–verb–object word order, phrase structure, and agreement,

[1]This definition conforms to *generative grammar*, a concept introduced by Chomsky (1957). Grammar is seen as a cognitive function, rather than a set of fixed, moored rules that should be taught and strictly followed by all speakers.

	Acceptable	**Unacceptable**
Grammatical	(1) John said he likes Mary.	(3) *The banana cried all room.
Ungrammatical	(2) ?More people have been to Russia than I have.	(4) *Mary likes he John said.

Table 4.1: Examples of grammaticality and acceptability judgments. Sentences adapted by Montalbetti (1984) and Leivada and Westergaard (2020).

without necessarily examining the semantic content of the utterance.

However, following the rules of building a sentence does not guarantee that an utterance can be understood and used for human communication. A speaker may consciously decide if a sentence belongs in their language, but this acceptance should not be confounded with a direct adherence to the status quo of grammar (Schütze, 2016). *Acceptability* is aligned with the intuition of a native speaker on how comprehensible and well-formed an utterance is (Greenbaum, 1977). It makes use of cognitive capacities and the knowledge of grammatical conformisms, in order to interpret an utterance (Bard et al., 1996). Grammars are created with the intention of creating rules that capture the largest number of utterances in a language, but cannot take into account all possible uses and intentions (Laporte, 2004). While linguists and non-native language learners benefit from the existence of grammar, for the native speaker acceptability holds more significance. For example, as seen in Table 4.1, it is possible for an utterance to be acceptable, without following all grammatical rules, as long as there is intelligible meaning and that some core structural elements of the language are preserved (in Example (2), the word order is respected but the pronoun agreement is incorrect). However, if the combination of words in an utterance is nonsensical (e.g. Example (3)), or too many rules are disrespected (e.g. incorrect word order and incorrect agreement in Example (4)), it will be rejected even if it follows the conventions of grammar.

Selection is the capacity of words to choose the semantic content of their arguments. A word's selectional preferences are defined by its propensity to occur in a syntactic relation with words belonging to specific semantic classes (Katz and Fodor, 1963). For example, the verb "eat", when used in a literal sense in the English language, requires a subject that is a living organism capable of digestion, and its direct object must be of the

Food class (Schütze, 2016). Selectional preferences are based on pragmatics and cannot be easily defined as strict rules, but as tendencies to favor particular arguments within a certain linguistic context, and reject others that result in conflicting or implausible meanings. For these reasons, it is common to refer to selectional preferences with a scale of *felicity*, rather than correctness or acceptability. However, the semantically felicitous combinations of words should respect the syntactic framework of a language, and in the case of a verb its subcategorization frame (i.e. the syntactic arguments that can occur with a specific verb in a predicate).

Selectional preferences commonly refer to the choices that a predicate makes for its arguments rather than the argument's preferences, following the syntactic hierarchy of a sentence (Light and Greiff, 2002). The meaning of a word can sometimes be used to explain its preferences and constraints, for example in the case of the verb "eat" in its literal sense (see Table 4.2). In other instances, however, a word's selectional preferences appear to be less rooted in the pragmatics of the real world; for example, "join" and "enlist in" are synonymous in the phrases "join/enlist in the army" (meaning Registration). While "join" can accept "a political party" in its predicate, "enlist in a political party" is considered an *infelicitous* phrase –neither ungrammatical nor unacceptable, but unnatural to the native speaker of English. The presence of homographs or metaphorical speech may also create pairs that are felicitous but contradict the preferences of a word; for example, the metaphorical use of "eat" as Torment means that the verb can accept an inanimate subject (see example (3) in Table 4.2). Idioms, i.e. language-specific phrases that were created to be intentionally nonsensical or whose original meaning is lost in time, also tend to create felicitous pairs that may not be evident to non-native speakers or in direct translation to different languages (see examples (5) and (6a) in Table 4.2). The presence of additional context may also shift the felicity of an utterance, with more details that clarify the meaning (see examples (4) and (5) in Table 4.2) (Měchura, 2008). Much like the concept of acceptability, whether a combination of words is felicitous or not rests upon the discretion of the native speakers, and is a complicated affair dependent on communicational needs, conversational cues, and language evolution.

		Animate S?	Edible DO?	Felicitous?
(1)	The goat eats an apple.	Yes	Yes	Yes
(2)	The house eats a marathon.	No	No	No
(3)	My thoughts are eating me alive.	No	No	Yes
(4)	John ate dust for breakfast.	Yes	No	No
(5)	John ate dust at the race.	Yes	No	Yes
(6a)	*Tu as mangé du lion?*	Yes	Yes	Yes
(6b)	Did you eat lion?	Yes	Yes	No

Table 4.2: Examples with the verb "eat" and different arguments, that produce felicitous/infelicitous utterances. Even when following the selectional preferences of a verb, it is possible to create semantically infelicitous sentences and vice versa. Example (6a) contains the original idiom in French *manger du lion* "to have a lot of energy", and (6b) is the direct translation of the sentence.

4.3 Selectional Preferences and NLP

As presented previously, selectional preferences are difficult to capture with rules, due to the speakers' needs for communication, expression, and creativity. Despite the linguistic community's best efforts to map verbs to their preferred arguments, it is difficult to take into account all possible uses. The use of statistical methods for NLP, however, provide the possibility for automatic extraction of relations and patterns between words, either from lexical databases like WordNet or from large corpora.

Resnik (1993, 1996) proposed the automatic selectional preference induction, for noun clusters (i.e. phrases), with the use of WordNet synsets and prior-posterior probabilities. The selectional preference strength of a specific verb v (Resnik, 1993, 1996) in a particular relation is calculated by computing the Kullback-Leibler divergence between the posterior class distribution of the verb and the prior cluster distribution, as seen in Equation 4.1:

$$S_{R(v)} = \sum_{c} \mathrm{p}(c|v) \log \frac{\mathrm{p}(c|v)}{\mathrm{p}(c)} \tag{4.1}$$

where c stands for a noun cluster, and R stands for a given predicate-argument relation. The *selectional association* of a particular noun cluster is the contribution of that cluster

to the verb's preference strength; for example, the best argument noun class for the verb "drink" are nouns of the Beverage class (see Equation 4.2).

$$A_{R(v,c)} = \frac{\mathrm{p}(c|v) \log \frac{\mathrm{p}(c|v)}{\mathrm{p}(c)}}{S_{R(v)}} \tag{4.2}$$

The model's generalization relies entirely on WordNet, focusing on generalization for noun classes.

Since then, there has been a lot of research on creating generalizations for word classes from WordNet based on probability distributions from large corpora (for example, Li and Abe (1998); Clark and Weir (2001); Ó Séaghdha and Korhonen (2012)). Alishahi and Stevenson (2007) proposed a probabilistic model derived from WordNet that predicts a predicate's preferences based on the semantics of the verb. However, research interest has gradually shifted from hand-crafted resources to acquiring selectional preferences from large corpora, which are superior to the WordNet generalizations as discussed in Zapirain et al. (2013). For example, Rooth et al. (1999) propose an Expectation–Maximization (EM) clustering algorithm (seen in Equation 4.3) for automatic induction of verb constraints, based on a probabilistic latent variable model. Their model generates both predicate and argument from a latent variable, where the latent variables represent clusters of tight verb-argument interactions.

$$\mathrm{p}(v, o) = \sum_{c \in C} \mathrm{p}(c, v, o) = \sum_{c \in C} \mathrm{p}(c)\mathrm{p}(v|c)\mathrm{p}(o|c) \tag{4.3}$$

The use of latent variables allows the model to generalize to predicate-argument tuples that have not been seen during training. The latent variable distribution and the probabilities of predicates and arguments are automatically induced from data using EM.

Erk (2007) and Erk et al. (2010) describe a method that uses corpus-driven distributional similarity metrics for the induction of selectional preferences, with the possibility of generating generalizations from domain-specific corpora. The key idea (seen in Equation 4.4) is that a predicate-argument tuple (v, o) is felicitous if the predicate v appears in the training corpus with arguments o'.

$$S(v, o) = \sum_{o' \in O_v} \frac{wt(v, o')}{Z(v)} \cdot sim(o, o') \tag{4.4}$$

where O_v represents the set of arguments that have been attested with predicate v, $wt(\cdot)$ represents an appropriate weighting function (in its simplest form the frequency of the (v, o') tuple), and Z is a normalization factor.

Van de Cruys (2009) proposes a tensor factorization-based approach that can simulate multiple selectional preferences. A latent tensor factorization model is used to generalize to unseen instances for three-way co-occurrences of subjects, verbs, and objects, that are represented as a three-way tensor (the generalization of a matrix). Bergsma et al. (2008) provide a discriminatory method for generating selectional preferences. Positive instances are drawn from observed predicate-argument pairs, while negative instances are created from unobserved combinations. The positive occurrences are distinguished from the negative ones using an sc-SVM classifier. Other proposed models are built on the topic modeling framework, either for the selectional preference of a predicate and a single argument (Ó Séaghdha, 2010), or multi-way selectional preferences for bi-transitive predicates (Ritter et al., 2010).

Approaches based on neural networks have also been employed, e.g. a feed-forward neural network with predicates as embeddings, to produce a single selectional preference value (Van de Cruys, 2014). Le and Fokkens (2018) use a neural network in order to extract one-way and multi-way selectional preferences for predicates –but make note that the use of automatically-extracted selectional preferences is not sufficient for downstream tasks. Zhang et al. (2019a) propose multiplex word embeddings for selectional preference modeling, with "relational" embeddings to capture how each word interacts with other words inside a given syntactic connection.

Evaluating the performance of selectional preference induction can be performed by humans or automated. Many researchers have used the pseudo-disambiguation task for evaluation, in which the model is asked to disambiguate between existing selectional preference pairs from a corpus and randomly constructed, corrupted pairs (Rooth et al., 1999; Ritter et al., 2010; Van de Cruys, 2014). It is also possible to perform human evaluation, i.e. the selectional preference judgments of the model are compared to labeled

datasets of human judgments, using a correlation measure. The process of collecting human judgments however is quite time-costly, therefore hasn't been favored by the researchers of automatic preference induction (McRae et al., 1998; Zhang et al., 2019b).

4.4 Experimental Setup

4.4.1 Datasets

4.4.1.1 SP-10K corpus

In our experiments with transformer-based architectures, we aim to analyze the English BERT-based models' competence in capturing selectional preferences and compare this performance to the human standard. There are several datasets of syntactic-semantic relations, released throughout the years for linguistics and NLP research, but not necessarily annotated with human evaluations of preferences and constraints (e.g. F-Inst (Ferretti et al., 2001), P07 (Padó, 2007) and GDS-all (Greenberg et al., 2015)). We were looking for a dataset with a sufficient number of entries and multiple human evaluations (in order to ensure that the felicity judgments were not skewed by one speaker's idiolect). Out of the two datasets we found in research, McRae et al. (1998) was not openly accessible, so we opted for the SP-10K dataset (Zhang et al., 2019b).

SP-10K is the largest dataset openly available to date for evaluating the selectional preference abilities of natural language processing tasks. It has been annotated by human Mechanical Turk workers, who were presented with word pairs without any other context and asked to evaluate the plausibility of the second word being dependent on the first with a specific syntactic relation. The dataset is composed of (slightly over) ten thousand pairs of words [2], evenly split into five different types of syntactic relations:

1. **nsubj**: verb and noun as *verb + nominal subject*
2. **dobj**: verb and noun as *verb + direct object*
3. **amod**: noun and adjective as *noun and modifier to the noun*
4. **nsubj_amod**: verb and adjective as *verb + (subject) + modifier to the subject*

[2] A few tens of sentences were added for the two-hop relation pairs, for the authors' tests on the Winograd Schema Challenge.

5. **dobj_amod**: verb and adjective as *verb + (direct object) + modifier to the direct object*

The first three categories include one-hop syntactic relations, i.e. direct connections between two nodes of a syntactic dependency tree [3]. The two latter represent higher-level, two-hop dependencies, i.e. two nodes that are not immediately connected but need an intermediate node. To better illustrate the meaning of dependency distance, we present a dependency parse in Figure 4.1. In the sentence "The lazy dog eats tasty treats.", the connections between two nodes (i.e. sentence tokens) with one arrow are one-hop relations (e.g. eat - treats), and relations that require two arrows and an intermediate node are two-hop relations (e.g. eat - tasty). Even though selectional preferences are usually built on direct semantic connections, Zhang et al. claim that, in certain cases and contexts, the constraints of the verb are strong enough to influence the choices made by their constituents.

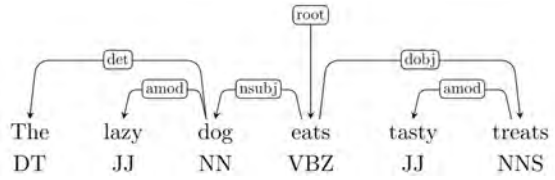


Figure 4.1: A dependency parse for the sentence "The lazy dog eats tasty treats.".

The words composing the word pairs are 2,500 frequent words, lemmatized, and all of the word pairs are annotated with a *plausibility score*, a concept similar to the felicity of a pair. The human annotators were presented with the word pairs and the syntactic relation and asked to rate the fit of the dependent word on that role, on a scale of 0 to 10. The speakers were asked to evaluate both the syntactic and semantic appropriateness, even though the words were not presented in the context of a sentence; this perhaps permitted more or less metaphorical interpretations of the context, but might have also misled some annotators.

The dataset is currently publicly available on the authors' Github page [4], in a simple text format, in separate files per syntactic relation.

[3]The syntactic formalism followed in the construction of this dataset is *dependency parsing*, the most commonly used in NLP applications.

[4]https://github.com/HKUST-KnowComp/SP-10K/tree/master/data

4.4.1.2 Prompt sentence corpus

Our goal is to investigate the relative importance of selectional preference information on BERT's predictions for a masked word within the context of a complete sentence. Therefore, we need to discover relevant sentences that contain the word pairs from the SP-10K corpus, in the proper syntactic positions and relations. To investigate the circumstances in which selectional preferences have an impact on the prediction of the masked word, we need grammatical sentences with a variety of contexts. We made the decision not to create our own prompt phrases, since doing so would have required a massive amount of work and could have produced unintended biases. Conversely, the SP-10K pairs were not sufficiently represented in available datasets of prompt phrases, either because they are too small (such as the Corpus of Linguistic Acceptability, Warstadt et al. (2018)) or too specialized on semantic relations (such as the LPAQA corpus, Jiang et al. (2020)).

As a result, we made the decision to extract sample sentences for each word pair from a large corpus. The ukWaC corpus (Ferraresi et al., 2008) contains about 2 billion words and a range of English texts (articles, titles, user reviews, etc.) that were collected by crawling websites in the .uk domain. We used a syntactically annotated version of the corpus, parsed with the Malt parser (Nivre et al., 2006). The dependency parser generates syntactic dependency trees in CoNLL-U format with lemmas, extended part-of-speech tags, and dependency labels (the same labels were used for the SP-10K corpus). This allowed us to identify the SP-10K word pairs in the ukWaC sentences in the correct syntactic positions and relations.

Among the 85 million sentences in the ukWaC corpus, we sought short sentences (4 to 15 tokens), in order to stay well under BERT's limit of 512 tokens per sequence and to ensure that the sentences were not mistakenly composed of multiple sentences (due to segmentation errors), multiple clauses, or complex and long-distance dependencies. We considered excluding some specific dependency labels, such as *xcomp* (open clausal complement) and *acl:rel* (for relative subclauses), but our selected sentences were already short enough to exclude the more complex syntactic phenomena. Automated parsing frequently fails to accurately identify passive structures, but this did not pose a problem for the syntactic relations we were examining. Thus we decided not to eliminate them

or take measures to correct the parsing. Additionally, we came to the conclusion that the distance between each pair of words in the phrase should be between one and five words, leaving enough room for determiners and modifiers without making the sentence excessively complicated.

We made the decision to assess the quality of the chosen sentences, at this point in the creation of our selectional preferences dataset. Parsing errors are inevitable in automatic dependency parsing, but when the same word pair is repeatedly incorrectly tagged, the result is many false prompt sentences. Additionally, the quality of the word pairs in the SP-10K corpus was problematic in some cases. For instance, we observed that several of the word pairs, when placed in the context of the defined syntactic relation, should have received the lowest possible score (zero), but were deemed felicitous or believable to some extent by the human annotators. We are aware that some of the word pairs were intentionally designed to be of low frequency or low plausibility; we are referring to falsely tagged syntactic structures. For these reasons, we decided it was necessary to perform a quick and non-exhaustive manual evaluation of the SP-10K word pairs and the resulting extracted sentences.

First of all, we noticed some problematic word pairs in the SP-10K corpus, which were included in a group with a certain syntactic relation, which they could not possess. For example, some word pairs under the verb-direct object relation included intransitive verbs such as “laugh”, “walk”, “smile”, or verbs that could not accept the dependent word as a direct object such as “look way”, “think time”. For many pairs, the words of a pair belong in similar contexts, hence these pairs were still assigned plausible scores when the plausibility should have been zero because of the proposed syntactic relation (e.g. “look way”, where “way” had a score of 6.5). We are unaware if these errors were caused by careless reading of the annotation instructions or by the annotators’ lack of knowledge of the exact syntactic relation between these words. These word pairs with problematic head or dependent words were removed from our query for sentences, in order to make sure that they were not accidentally found in a sentence with a wrong parse tree.

On the other hand, some word pairs, especially the ones which were by design very infelicitous and had a low plausibility score, are not found in the ukWaC corpus – almost half of the word pairs for all types of syntactic relation. However, some word pairs are

very common in the corpus and are found in disproportionately more sentences. In addition, several word pairs are parts of idiomatic, lexicalized phrases, and are very frequent in the ukWaC corpus and almost exclusively found in the context of these idiomatic phrases, but were assigned a low score. As an example, for the **nsubj** relation, in the pairs "weather permit" (4.06) and "study find" (4.0), the subjects are inanimate (whereas the verbs generally require an animate subject) but in this specific metaphorical use, they are acceptable.

These problems in the correct and consistent annotation of word pairs and their plausibility scores might have negatively affected the correlation results. However, we believe that the extent is limited, after our manual evaluation of the word pairs.

The number of sentences that we extracted from the ukWaC corpus is presented in Table 4.3. These sentences are organized per type of SP-10K word pairs, for the pairs found in at least one sentence with the given parts of speech and dependency relation. These sentences have been counted after implementing our criteria of length and distance that we previously determined, and our manual evaluation.

Type	Word pairs in ukWaC	Found sents	Final sents	Score
nsubj	958 / 2,000	38,613	30,526	6.64
dobj	980 / 2,000	70,250	56,777	7.39
amod	1,030 / 2,000	29,403	23,110	7.62
nsubj_amod	956 / 2,061	15,265	12,911	5.75
dobj_amod	922 / 2,063	28,336	21,839	6.32
Total	4,846 / 10,124	181,867	145,163	

Table 4.3: The number of SP-10K word pairs which were found in sentences of the ukWaC corpus (out of the total number of word pairs), the initial number of found sentences, and how many of those sentences include the word pairs in the correct syntactic positions (after our evaluation). The last column documents the average score of the SP-10K annotated plausibility scores of all pairs in the category.

4.4.2 Transformer models

Our experiment makes use of the pretrained contextual word embeddings from English monolingual BERT. One of the training objectives for the BERT models is *masked language modeling*, i.e. the prediction of a masked word in a sequence. Our experiment

was modeled after this training objective, as well: for a sentence with an SP-10K word pair, we mask the dependent word of the pair. The model receives the sentence with a masked word, and the probability of the dependent word being in its original masked position is retrieved. The experimental setup will be further explained in the following sections.

We used the bert-base-uncased model for English, as provided by HuggingFace's transformers Python library (Wolf et al., 2020), and specifically its version for masked word prediction. The BERT base model has been exploited and extensively analyzed for its semantic and syntactic competencies, for example in Goldberg (2019) and McCoy et al. (2019). Some preliminary experiments we performed with bert-large-uncased did not show significantly different results, thus we used the computationally lighter base model. We do not perform any finetuning of the encoder weights or pruning of any heads but use the pretrained model as it is made available.

On the sentences of our ukWaC & SP-10K dataset, we add the special BERT tokens [CLS] (to indicate the start of a sentence) and [SEP] (to mark the end of it). After masking the dependent word, we use the BERT tokenizer, so that the sentence is tokenized to words and subwords that can be matched to BERT's embeddings. However, for our masking experiment, we could only preserve the dependent words that were not tokenized to subwords, as the pretrained model from the transformers library only supports one masked token. Out of the 250 unique words of the SP-10K word pairs, only 27 were split into subword segments and therefore were not eligible for our experiments (since computing simultaneously the probabilities of multiple adjacent subwords is not possible with the BERT models we employed)[5].

4.4.3 Methodology

4.4.3.1 Correlation of SP-10K score and probability

For each example sentence in our corpus, we mask the dependent word of the word pair using a [MASK] token, and we retrieve the probability that is assigned to the target

[5] "grandparent", "hightech", "indiscreet", "cholesterol", "allegation", "pant", "grandchild", "africanamerican", "tasty", "socalled", "tshirt", "fulltime", "wellknown", "carbohydrate", "guideline", "symptom", "oldfashioned", "tablespoon", "lawmaker", "youngster", "cede", "shortterm", "wellbeing", "longterm", "stereotype", "respondent", "nosy"

word in the focal position. The probability is computed by passing the last hidden state through a *softmax* function.

We are making the assumption that this result is to be treated as the conditional probability of a bi-directional language model, similar to what a traditional language model would return.

We compute the correlation of the masked word's probability and the plausibility score of the word pair, using the Kendall rank correlation coefficient as implemented by the scipy Python library. Kendall τ (tau) correlation is a non-parametric measure of the monotonicity of the relationship between two datasets. The p-value roughly indicates the probability of an uncorrelated system producing datasets that have a correlation at least as extreme as the one computed from these datasets. [6] Values close to 1 indicate a strong positive correlation, while values close to -1 indicate strong disagreement. Intuitively, we are looking for a strong positive correlation, meaning that the higher the plausibility score of the word pair, the higher the probability of the dependent word in the context of the head word.

We created batches of 128 sentences and used CUDA to accelerate our calculations.

4.4.3.2 Prediction with attention masks

In order to determine the relative importance of selectional preference information, we retrieve probabilities in different attention settings by using **attention masks**. As explained in Section 2.5.4, the attention mask is an array of 1s and 0s indicating which tokens we do not wish to incorporate in the way the model attends to the sequence. By using this feature, we are able to "block" certain tokens of the sentence from BERT's self-attention mechanism, and examine the impact it brings to the probability scores and the correlation. These masks do not completely block parts of the input, only mask them from the attention mechanism. We use four different settings:

- The *standard* setting does not involve any masking, thus the model can attend to the whole sequence.
- The *head* setting blocks attention to the head word of the pair, so the attention can only

[6] In the remainder of this doctoral work, significant results are defined as $p < 0.01$.

focus on the rest of the context (other arguments of the head word and non-dependent words).

- The *context* setting masks the context except for the head word, so the attention can only focus on the head word (and BERT's special tokens).
- The *control* setting masks all the words of the sequence (except for the special tokens), so that the attention mechanism is "sabotaged" and no adequate prediction should be possible (as a sanity check).

A graphical example of the four different attention settings is given in Figure 4.2.

Mask Type		The	film	tells	the	story	of	that	trial	.	
standard	[CLS]	the	film	tells	the	[MASK]	of	that	trial	.	[SEP]
head	[CLS]	the	film	-	the	[MASK]	of	that	trial	.	[SEP]
context	[CLS]	-	-	tells	-	[MASK]	-	-	-	-	[SEP]
control	[CLS]	-	-	-	-	[MASK]	-	-	-	-	[SEP]

Figure 4.2: Illustration of the four attention mask settings, for the sentence "The film tells the story of that trial." with the word pair "tell story" (as a **dobj** relation). Dashes indicate blocked attention.

The correlation scores between the plausibility scores and the model's probabilities are calculated per sentence, then the results are both micro- and macro-averaged. As previously indicated, there are considerable differences in the amount of extracted sentences for each word pair. The micro-averaged findings are calculated over the whole collection of sentences, without accounting for this varying amount of sentences. For the macro-averaged results, we first calculate the average for each word pair's sentences, before providing a total average for all the pairs (hence treating all word pairs as equally important). A value above 0.4 will be regarded as a strong positive correlation, and a value below -0.4 will be regarded as a strong negative correlation.

4.5 Results

4.5.1 Quantitative results

In Table 4.4, the correlation results between the human assessments of plausibility and the probability of the dependent word in the sentence are presented. Our results do not demonstrate a strong positive or negative correlation for any of the five syntactic relation categories, neither without masking the attention nor with the attention masks. However, we observed interesting differences between the different settings, that point to assumptions about selectional preferences and attention.

Mask type	*standard*	*head*	*context*	*control*
nsubj	0.03	-0.02	0.16	-0.01
dobj	0.05	-0.07	0.05	-0.05
amod	0.04	-0.06	0.24	-0.04
nsubj_amod	-0.01	-0.13	0.29	-0.00
dobj_amod	0.06	0.01	-0.03	0.02

(a) Micro-averaged results

Mask type	*standard*	*head*	*context*	*control*
nsubj	0.19	0.15	0.29	0.08
dobj	0.16	0.04	0.27	0.05
amod	0.15	0.03	0.35	0.03
nsubj_amod	0.01	-0.04	0.22	0.06
dobj_amod	0.14	0.10	0.20	0.07

(b) Macro-averaged results

Table 4.4: Kendall τ (tau) correlation coefficient of masked word probability and word pair plausibility score.

The *context* attention mask scenario (masking attention for the entire sequence aside from the head word and BERT tokens) exhibits the highest positive values (positive correlation up to +/-.30), whereas the *head* attention mask scenario (masking attention for the head word and attending to the context for prediction) displayed no strong correlation but was biased toward negative values. Prediction with attention to the entire sequence is successful, because both the head word and the context are important, but the presence of context slightly lowers the probability of the dependent word. Attention

solely to the head word worked the best, as the relationship of the word pair dictates. However, the absence of attention to the head word acts negatively, even in the case of two hop-relations. This finding lends credence to the claim that, in syntactic relations where verbs serve as the head word, the verb's selectional preferences are comparatively significant and sufficiently influential in the choice of constituents. BERT seems to have captured these preferences and constraints in its encodings and makes use of them to assign the dependent word a proportionate probability.

The **nsubj** relationships exhibit slightly stronger positive correlations than the **dobj** relationships for direct objects, possibly due to the restriction of animacy for some subjects (word pairs included a variety of animate and inanimate subjects). The two-hop relations, **nsubj_amod** and **dobj_amod**, exhibit lower correlations but are still quite strong with attention only to the head word (the *context* attention mask), supporting the claim of Zhang et al. (2019b) that selectional preferences extend beyond one-hop relations.

4.5.2 Analysis of head words

Taking a closer look at the head words of the word pairs, we searched for strong positive or negative correlations for each head word that exists in at least two different word pairs, per syntactic relation. We examined whether specific verbs and nouns affected positively or negatively the correlation of probability and plausibility, and whether there were common features between these head words (e.g. semantic similarity, common semantic class). We grouped the probabilities and scores of sentences per head word, and calculated the correlation coefficient for head words that were present in at least two different word pairs. Overall, for all five syntactic categories of our experiment, we do not notice distinct classes, semantic or syntactic, that the words with strong correlations could be grouped with.

For **nsubj** relations, verbs with semantic similarity (in at least one of their meanings) did not demonstrate similar patterns of probability and correlation; for example, the verbs of violence (in some contexts) "kill", "strike", "grab", "fire" show a strong positive correlation, while the verbs of the same semantic class "shoot" and "confront" have a strong negative correlation – this could be caused by the different metaphorical mean-

ings that these two words might have, or the dependent words that they were paired with in the SP-10K dataset (favorable for "kill", detrimental for "shoot"). Concerning the type of subjects, the animate subject "man" had a high plausibility score in the SP-10K dataset and high probability scores for "kill" and "shoot", causing a strong positive correlation. The inanimate subjects had mid-range plausibility scores ("earthquake", "explosion") or low scores ("film", "tragedy") but the probability varied based on the sentence and metaphorical use; for the word pair "strike tragedy" which existed in many sentences of our dataset, the plausibility score was 5.25 and the assigned probability for "tragedy" was relatively low, even though the idiomatic phrase "tragedy struck" is fairly common. Likewise, we noticed that in the standard head attention mask scenarios, the word pair "kill explosion" had a strong positive correlation while "shoot man" 8.0 had a strong negative correlation; interestingly enough, the former had a plausibility score of 6.25 while the latter had a score of 8.0.

Examining the **dobj** relations, verbs (head words) showed inconsistent correlations among the different attention mask scenarios; out of the few verbs that showed consistently positive or negative correlation, we were not able to identify semantic clusters of verbs or differences based on verb transitivity (monotransitive/ditransitive). The presence of a strong correlation relied more on BERT's semantic knowledge rather than world knowledge or utterance plausibility; for example, the word pair "blame customer" has (correctly) a moderate plausibility score (6.75), is found twice in the ukWaC corpus, but the assigned probability by BERT of the word "customer" is very low in the *standard* and *context* attention mask scenarios. The word pair "blame management", on the other hand, with slightly lower plausibility (6.25) is assigned a proportionally good probability. This leads us to the conclusion that, even though both syntactic pairs are grammatically correct and have commonly used words, the pretrained model has learned that "management" (someone in control and responsible of a service) is a more probable direct object for the verb "blame" than the word "customer" (the receiver of a service), especially when the only given context is the verb. When attention to the head word was removed, there was no strong negative correlation between "blame" and the given plausibility score.

Concerning the **amod** word pairs, again no semantic class of nouns appear consis-

tently in the positive or relative correlation groups. An interesting observation is that high-frequency adjectives of size and age, such as "small", "big", "old" and "new" were almost always assigned a high probability by BERT, but the variations in plausibility score (from 8.25 to 4.25) led to strong positive or negative correlations, especially since word pairs with these adjectives are quite frequent in our corpus, for example "new house", "small bird" and "new face" had many occurrences in the corpus and a strong positive correlation (high plausibility/high probability), "new material" (6.5) and "old daughter" (4.25) had lower plausibility scores and subsequently lower probabilities, in all attention mask scenarios. Unlike in the other syntactic relation groups, we do not notice shifts from strong to weak correlation, based on the attention mask scenario, though in most cases blocking the head word attention yielded a slightly weaker correlation and blocking the context word attentions a stronger one.

In the **nsubj_amod** word pairs, again we see that high-frequency descriptive adjectives (dependent words) are still assigned higher probabilities, even though the plausibility scores are more mediocre for the word pairs of this relation, therefore high-frequency adjectives can be found in both the strong positive and negative correlation groups. We also do not notice distinctive semantic classes among the verbs (head words), and neither can we make assumptions based on the animacy of the subject, since the adjective modifiers do not follow such a constraint ("new", "local", "national", "exact", "different") and the given verbs do not have the animacy constraint either ("bring", "attract" had a strong positive correlation, "increase", "reflect" a strong negative). Some verbs that do prefer animate objects were found to have a strong positive correlation (e.g. "compare", "operate"), others to have a strong negative one (e.g. "lift"). Concerning the different attention scenarios, there is a noticeable positive shift in correlations ($+.30$, $+.20$) with the *context* attention mask compared to no mask or masking the head word, which hints at the influence that the verb had in the predictions, and how the context (including the one-hop dependency to the verb subject) produced less polarizing probabilities.

Finally, for the **dobj_amod** word pairs, as in the direct object word pairs, we do not notice verb grouping based on semantics or transitivity. Many of the verbs with strong positive ("teach", "promise") or negative correlation ("claim", "confirm") are verbs with varied subcategorization frames. In this syntactic category, we observe the smallest pos-

itive shifts with the use of the *context* attention mask, and even a decrease in correlation (-0.03) in the micro-averaged results. However, the results still show a weak positive correlation similar to the ones of the other syntactic relations, for the most part; this observation supports the fact that the role of the verb is quite important for predictions.

4.5.3 Correlations and attention per layer

We conducted several experiments in order to visualize the behavior of BERT per model, and to better peer into the attention layers. Focusing on the standard setting without the use of attention masks, we froze the model on each layer (see Section 2.4.4 for details on freezing layers) and asked the model to predict the probability of the dependent word. Then, we calculated the correlation to the human assessments. By freezing the model on each layer, we can only assess the predictions per layer, therefore the results are different than those presented in Table 4.4. Overall, there are no strong correlations, with layers being slightly positively or negatively biased (± 0.10) for all five syntactic relations categories. As seen in Figure 4.3, the performance of layers is not always uniform among the different relation types and neighboring layers demonstrate quite different behavior (e.g. from layers 5 to 8). We observe that for most types of syntactic relations, the models achieve the highest positive correlations in layers 7-9, with layers 2 and 5-6 also having a tendency for positive correlations. The final layers do not always demonstrate the highest positive correlations. This finding is in accordance with the literature on the linguistic capabilities of the models (see Section 3.2), where it is observed that BERT models show some linguistic specialization in their layers, with early middle layers being focused on syntax and late middle layers on semantics. However, this experiment does not provide solid proof of syntactic and semantic competencies, since there were no strong positive or negative correlations in any category or layer.

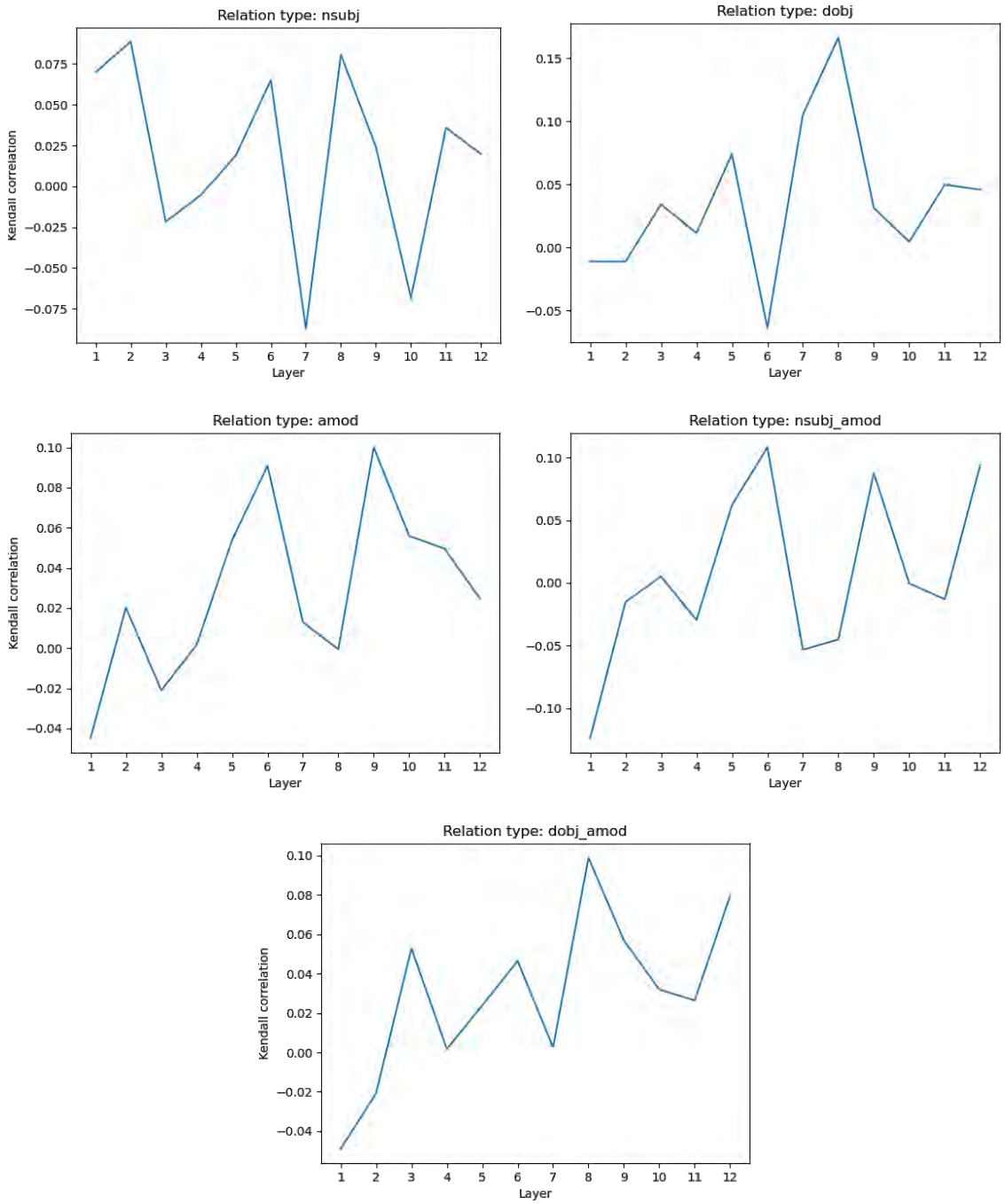


Figure 4.3: Correlation of masked dependent word probability and human assessments from SP-10K. The graphs are separated per syntactic category.

Focusing on attention, we examined the way the tokens of a sentence attend to the other tokens of the sentence, in the case of unmasked attention. The attention weights of the 12 heads of the bert-base-uncased model were aggregated per layer and plotted in a heatmap. The different plots per layer allow us to observe how layers attend to tokens, and whether the specialization we observed in the probability correlations was reflected in attention, too. Every square in a plot represents how a token in a sequence attends to the rest of the tokens of the sequence, and the lighter the shade, the higher the attention weight to the target token. In Figure 4.4 the attention heatmaps are shown for the sentence "The event took months to plan." which includes the **nsubj** word pair "take event". Additionally, the attention heatmaps for the sentence "Give us the chance to teach loving." for the **dobj** word pair "give chance" is shown in Figure 4.5.

This traditional method of depicting attention may not be perfectly suited for multi-headed self-attention, since it doesn't offer insights into how each head behaves and attends to the input. However, it does visualize the behavior of each layer in a human-readable way and offers insights into the choices of the model in each layer. For example, we notice that some layers show the same patterns of self-attention even with different sentences; layer 3 shows a monotonic focus of attention of a token to its right neighbor, and layer 12 shows that tokens aggregate their attention to the end-of-sequence token. Examining each sentence we selected to present, we notice that the token "took" does not strongly attend to the masked position of the "event" token in any of the layers, apart from some attention in layer 2, and prefers to attend to the subordinate clause. Similarly, "give" attends to the masked position of the "chance" token mostly in layer 2 and also prefers the subordinate clause. The determiners "the" and "a" still attend to the masked positions, however.

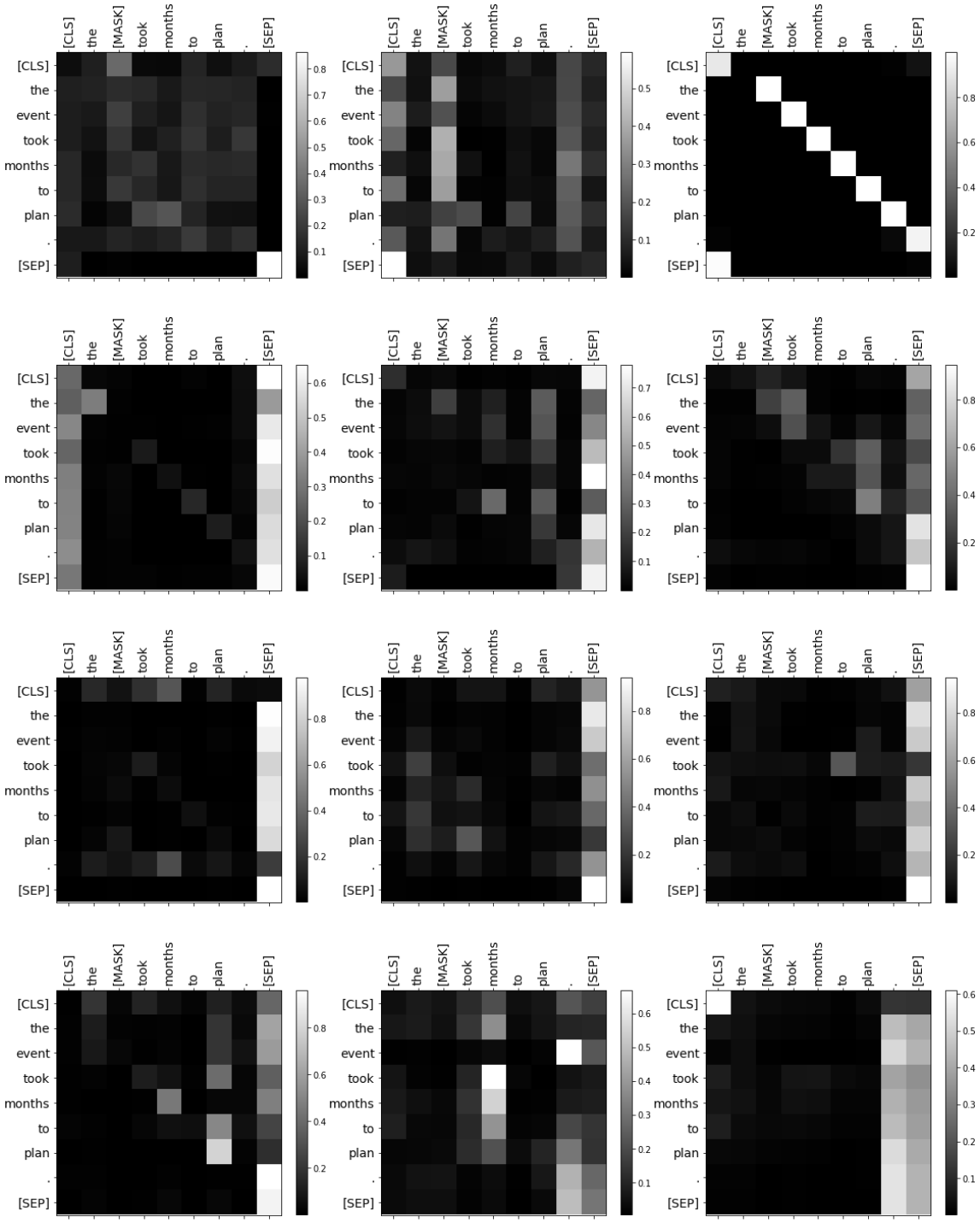


Figure 4.4: Attention heatmaps for the sentence “The event took months to plan.”, for the **nsubj** word pair “take event”, for layers 1-12 of bert-base-uncased.

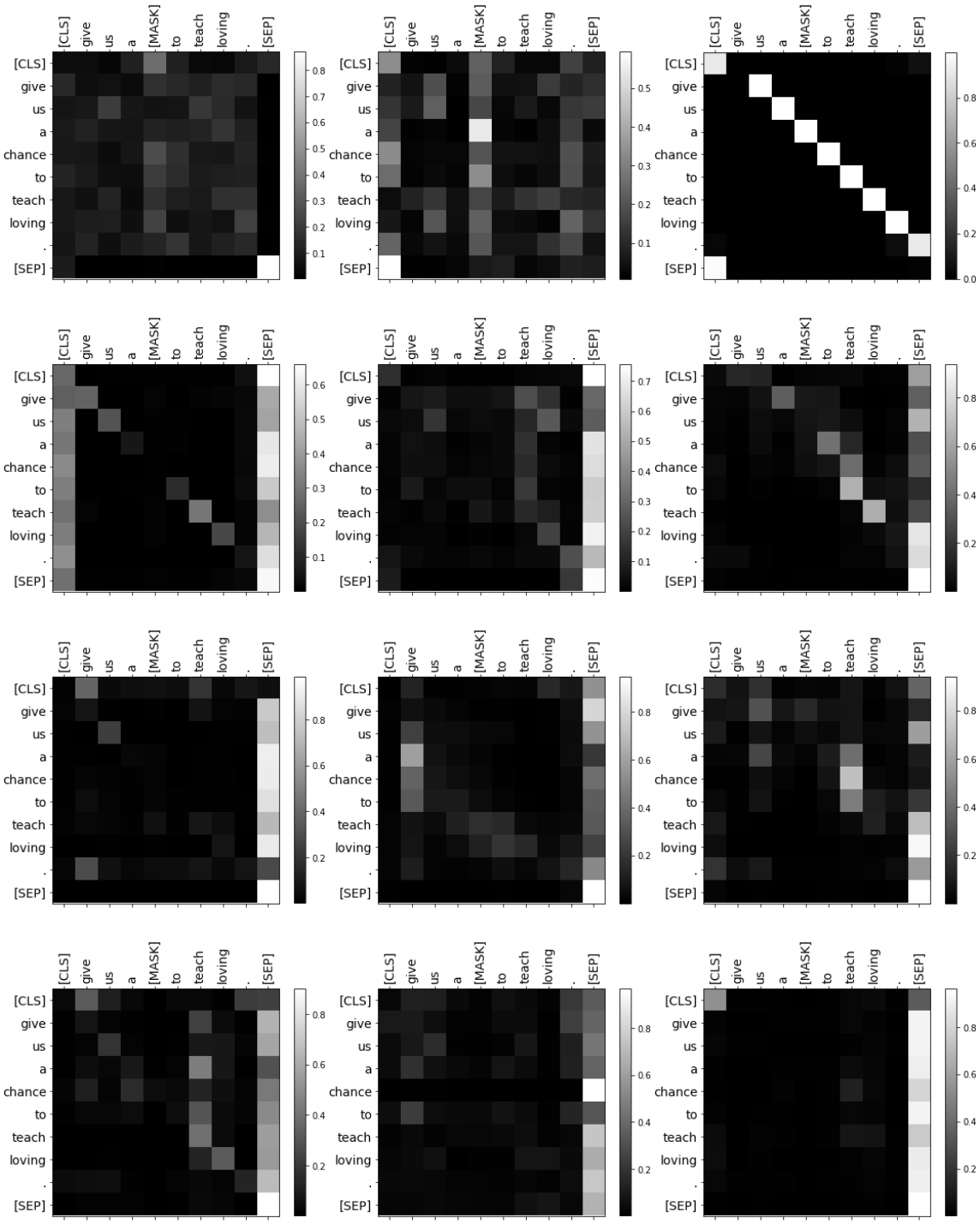


Figure 4.5: Attention heatmaps for the sentence "Give us a chance to teach loving.", for the **dobj** word pair "give chance", for layers 1-12 of bert-base-uncased.

4.6 Discussion

Selectional preferences are hard to define, and as seen by the SP-10K annotating process, challenging to identify even to humans. In our experiment, we investigated whether the contextual word embeddings of BERT (specifically, the bert-base-uncased version) contain such information, by comparing their knowledge to human annotations. We studied the head word-dependent word pairs in the SP-10K corpus, in real sentences; the process of extracting real sentences from ukWaC with these word pairs was already a method of filtering out the infelicitous pairs that existed in SP-10K (for research purposes). The correlation coefficient between human judgments and BERT probabilities did not show a strong positive or negative correlation; this is caused by two factors. First, the average plausibility of the pairs in SP-10K was sometimes quite low for common pairs, especially those found in idioms and lexicalized phrases. Meanwhile, BERT assigns high probabilities in frequently seen pairs of words and contexts. Second, BERT had the advantage of accessing full sentences, while the annotation in SP-10K was performed on word pairs. Additionally, as mentioned already, SP-10K pairs were created out of frequent words, and BERT tends to favor the prediction of frequent words (e.g. adjectives of size), sometimes to a fault. However, we were not demanding a human-like performance by BERT, but rather we explored its learned preferences and whether they coincide with human intuition to some extent.

The ability to use attention masks allowed us to study how the likelihood of the target word can change, depending on how the input sequence is processed by the self-attention mechanism. Our objective was to determine how much the head word influenced the probability of the dependent word and if the context was more significant than the head word alone. The fact that the strongest positive correlation values nearly always resulted from focusing attention exclusively on the head word (and the non-lexical BERT tokens) suggests that the head word is recognized as an essential and significant component of the sequence when it comes to choosing a masked word. Blocking the attention mechanism from accessing the head word also showed the head word's importance to the assigned probability to the dependent word, even in two-hop relations where the one-hop relation was masked. Further experiments with self-attention, how-

ever, did not pinpoint this observation to a specific layer in the model, as the literature has suggested.

Comparing the different syntactic relations of the SP-10K corpus, the lowest correlation scores came from the **amod** syntactic relation category, even though some nouns also have strong lexical preferences. This stems from BERT's favoritism of high-frequency adjectives, which in some cases may not be very felicitous to nouns. Interestingly enough, the verb and adjective (as the modifier to the subject or the direct object) categories showed, for the most part, similar positive correlations as the one-hop syntactic relations of verb and noun. As Zhang et al. (2019b) have also mentioned, these two-hop relations also fall under the influence of a word's selectional preferences. The head word in these cases is the head of the sequence and the subject or direct object are its arguments, therefore its selectional preferences could have impacted the selection of a modifier to a greater extent than the context could.

Our overall results did not show a strong correlation that would definitively prove or disprove the presence of selectional preferences, but there are indications that BERT's embeddings have captured enough syntactic-semantic information to be able to assign probability based on "the right fit" for a head word. BERT is able to capture, to some extent, a verb's preferences and constraints, and can make predictions based on them, when the use is not metaphorical and conflicting with usual, literal cases.

Chapter 5

Classification of lexical aspect in English and French

5.1 Introduction

Lexical aspect is the property of a verb describing the temporal qualities of the verb's action, event, or state. Unlike grammatical aspect and verb tense, it is a semantic property that is innate to the verb, and could only change in the presence of different meanings and contexts. The comprehension of lexical aspect is crucial for many tasks where semantic knowledge is required, since aspect conveys details on temporal relations (Costa and Branco, 2012), textual entailment (Hosseini et al., 2018; Kober et al., 2019) and event ordering (Chambers et al., 2014).

Our goal was to discover whether deep contextual word embeddings can learn and encode lexical aspect information in their encodings. We focused on the properties of *telicity* (the existence of an endpoint or not) and *duration* (the presence of an action or a state). We used transformer-based architectures by finetuning the sequence classification language models on a dataset of telicity and duration annotations.

We conducted two rounds of experiments, one with the dataset of Friedrich and Gateva (2017) and another with the addition of the dataset of Alikhani and Stone (2019), and finetuned the English pretrained models. The second iteration of our experiments was deemed necessary, first of all, because the results of the first round of experiments did not produce sufficient explanations, and second, because of some annotation problems we identified in the Friedrich and Gateva dataset. In both rounds of experiments, our main experiment was conducted as follows; We trained the models on a binary sequence

classification task of telicity or duration (*telic-atelic* and *stative-durative*) in two different ways: by either providing the verb position or not. We tested on a held-out test set from the datasets, and two smaller hand-crafted test datasets with simple sentences. Additionally, we performed experiments with attention masks, attention visualization methods, and the knowledge of the pretrained word embeddings. We were also able to extend some experiments in French, with translations of our datasets and the French transformer models.

Even when trained on limited datasets, transformer models were quite effective in classifying data. Adding the verb position as extra information enhanced performance in telicity and duration categorization for English, but not for French. Even without finetuning, the pretrained word embeddings contain knowledge of lexical aspect, inside the verb representation. From the analyses of our qualitative test sets, we observed that the models classified based on verb before context, meaning that they are able to distinguish the most important part of the sequence. However, when required to, they failed to capture more precise information, for complex sentences where the verbal aspect contradicted the temporal information in the context.

5.2 Linguistic overview

The internal temporal organization of the events that verbs (predicates or sentences) describe has been the subject of continuous research. The linguistic concept of *aspect* describes the temporal characteristics of a verb's reported action, occurrence, or state, beyond the scope of the verb's tense. Aspect communicates information like frequency, duration, and completeness, but this information can either be dynamic or inherent to the verb's meaning. *Lexical aspect* (or *aktionsart*) relies on the meaning of the verb (a described event, state, action, or accomplishment), and these meanings cannot change regardless of how they are placed on a timeline. For example, in Table 5.1, the sentence "I eat an apple." is presented in different grammatical tenses of present. These tenses change the time that the action of Eat occurs but cannot change the lexical aspect: the action of Eat is bound to finish once the Food is consumed.

The endeavor of defining lexical aspect is a complex one, and apart from the semantic meaning of a verb, its perception is the outcome of the entire verbal phrase and

not solely the verb's features (Krifka, 1998). However, it should not be confused with *grammatical aspect*; grammatical aspect defines temporal properties that can change in different contexts and can be expressed with syntax and morphology. This distinction isn't always clear for all contexts and languages; even the first definition of telicity was built on the grammatical feature of *perfectivity*. According to Garey (1957), telic verbs express actions that are directed towards a goal that is thought to be realized in the perfective tense but contingent in the imperfective tense. In contrast, atelic verbs express actions that are realized as soon as they begin, and lack any goal or endpoint in their semantic structure.

		Lexical aspect		
Tense	**Sentence**	**Telicity**	**Duration**	**Frequency**
Present simple	*I eat an apple.*	telic	durative	non-repeated
Present continuous	*I am eating an apple.*	telic	durative	non-repeated
Present perfect	*I have eaten an apple.*	telic	durative	non-repeated
Present perfect continuous	*I have been eating an apple.*	telic	durative	non-repeated

		Grammatical aspect	
Tense	**Sentence**	**Progressivity**	**Perfectivity**
Present simple	*I eat an apple.*	not progressive	imperfect
Present continuous	*I am eating an apple.*	progressive	imperfect
Present perfect	*I have eaten an apple.*	not progressive	perfect
Present perfect continuous	*I have been eating an apple.*	progressive	perfect

Table 5.1: Features of lexical and grammatical aspect, of the present tense, in English.

Since then, there have been multiple proposals to define the different features included in lexical aspect and correlate them to verb classes. Vendler (1967) divides verbs (more accurately, predicates) into four categories: state, activity, accomplishment, and achievement. *States* (*know, believe, own*) are homogeneous, non-dynamic, and continuous circumstances (Dowty, 1979; Kearns, 1991; McClure, 1994). *Activities*, such as *running, walking, swimming* are dynamic activities that carry on continually with no obvious endpoint (Smith, 1997). *Accomplishments*, such as *draw (a picture), run (a mile), build (a house)*, are dynamic and durative events with an inherent endpoint. *Achievements* (*recognize, arrive, die*) are dynamic and near-instantaneous events with an inherent endpoint. *Semelfactives*, added by Comrie (1989) and Smith (1997), refer to punctual events

such as *cough, knock, wink*. They are similar to activities, but they have an instantaneous endpoint and can be iterated. *Scalar* verbs (Hovav and Levin, 2010) include activity verbs, e.g. *cool, widen*, with a degree of achievement, thus they may have a varied endpoint in different contexts (and depending on the context, the presence of an endpoint may be irrelevant).

Olsen (1994, 1997) and Kearns (2000) chose a feature-based representation for aspectual classes, in order to accurately distinguish between the various aspectual classes. The three binary properties ([±dynamic], [±durative], and [±telic]) and their existence or not in an aspectual class can be found in Table 5.2. It is also important to highlight that these classifications usually include a verb in a specific grammatical form (e.g. *running*) or with specific arguments (e.g. *draw a picture*), since these factors can possibly affect the event structure of a verb (Siegel, 1998).

	Verb example	**[±dynamic]**	**[±durative]**	**[±telic]**
State	*know*	-	+	-
Activity	*running*	+	+	-
Accomplishment	*draw (a picture)*	+	+	+
Achievement	*recognize*	+	-	+
Semelfactive	*cough*	+	-	–
Scalar	*cool*	+	+	±

Table 5.2: Binary properties of lexical aspect and how aspectual classes include them. Table adapted by Peck et al. (2013).

However, lexical and grammatical aspects are intertwined, in that the semantic constraints of lexical aspect can influence how grammatical aspect can be expressed on a verb. As a result, the grammatical aspect can be a surface-level indicator of lexical aspect; for example, in Czech and other Slavic languages, an atelic verb can be created from a telic verb with a related meaning and the addition of a morpheme of perfectivity (see Table 5.3). By adding the prefix *na-* to the imperfective verb, it is transformed into a perfective verb, and one or another meaning specialization is very often superimposed on the imperfective meaning (Šimandl et al., 2016). The derivative verb expresses a different meaning from its base verb, therefore may express a different degree of telicity (e.g. *napsat* "to write" is a telic verb, as opposed to its imperfective base form *psát* "to write" which is atelic). In these cases, the context is chosen by the verb to complement

Imperfective	**Perfective**
psát "write"	*napsat* "write (something)"
Pořád píš-u dopisy. still write-PRS.IND.ACT.1SG letters I still write letters.	*Na-psa-l o tom článek.* PF-write-PST.IND.ACT.3SG.M about it article He has written an article about it.

kreslit "to draw"	*nakreslit* "to draw (something)"
Kresl-i-l-a obrázky. draw-THEME-PST.IND.ACT-3SG.F pictures She drew pictures.	*Na-kresl-i-l-a květinu.* PF-draw-THEME-PST.IND.ACT-3SG.F flower She drew a flower.

Table 5.3: Examples of deriving an imperfective verb from a perfective base form with the prefix *na-* in Czech, with corresponding sentences. Sources: Šimandl et al. (2016); Ševčíková et al. (2017)

the verb's degree of telicity, e.g. when the action of drawing has an endpoint with a direct object of a known, quantifiable size, the verb form *nakreslit* will be chosen over *kreslit*.

For the purposes of this doctoral work, and in order to comply with the annotations of our datasets, we are focusing on two features of lexical aspect, in a binary manner. **Telicity** denotes the potential presence of an endpoint or not; if the verb's action can be completed in the past, present, or foreseeable future the verb is *telic*, but if the verb describes a state or action whose completion is either indefinite, impossible, or irrelevant the verb is *atelic*. **Duration** distinguishes between a state (*stative* verbs) and an action (*durative* verbs), regardless of the existence of a possible endpoint or not – with a possible distinction of the duration of the action (*punctual* verbs). Examples of telic/atelic and stative/durative sentences will be presented in the following sections, discussing our experiments and datasets.

5.3 Identifying and learning aspect with NLP

In modern NLP research, Siegel and McKeown (2000) were the first to propose natural language processing methods for aspectual classification. They located linguistic indicators of stativity and completedness using decision trees, genetic programming, and logistic regression, and reported that supervised methods performed better than unsu-

pervised approaches.

In order to predict a verb's stativity and duration, Friedrich and Palmer (2014) use a semi-supervised strategy of learning lexical aspect, by incorporating linguistic and distributional variables. This work has also provided two datasets of annotated sentences for stativity. Friedrich and Pinkal (2015) extended this approach by classifying verbal lexical aspects into multiple categories of duration and features of habitual/episodic/static, with the use of Brown clustering (Brown et al., 1992). Friedrich et al. (2016) expanded their datasets and classes, achieving 76% accuracy on supervised classification of stativity and duration with Brown clustering, compared to their human baseline of 80%. Friedrich and Gateva (2017) published two datasets in English with gold- and silver-level annotations of telicity and duration as part of their most recent study. Gold annotations come from humans, while silver annotations are obtained from parallel English—Czech corpora, where aspectual features were inferred by Czech morphological markers of perfectivity. They claim that automatic telicity classification has significantly improved, with the use of these datasets and their L1-regularized multi-class logistic regression model. They report results on training and testing with their gold-annotated dataset of 86.7% accuracy of telicity classification –and up to 86.2 accuracy on the gold test set, with the addition of their silver-annotated data.

Loáiciga and Grisot (2016) exploit telicity annotations in order to improve on French–English machine translation. They observe that tense is better translated with the use of verb classification of telicity (defined as *boundedness*). Falk and Martin (2016) also use machine learning techniques in addition to morpho-syntactic and semantic annotations to predict the aspect of French verbs in different contexts (*verb readings*). Peng (2018) employ the dataset of Friedrich and Gateva and two different compositional models (PLF, LSA) to classify telicity. They emphasize the significance of the verbal phrase and the verb's dependents in the interpretation of telicity, and they report accuracies of up to 89%. Kober et al. (2020) also propose modeling the aspect of English verbs with regard to their context. They use compositional distributional models and confirm that the context of a verb and closed-class words of tense (e.g. prepositions, auxiliary verbs) are important features for aspect classification.

Alikhani and Stone (2019) created a multi-lingual annotated dataset of image cap-

tions, in which they included annotations of grammatical and lexical aspects. They conducted quantitative and qualitative analyses of these captions, in order to explore how temporal qualities depicted in an image are interpreted and expressed linguistically by human annotators. An extension of this work was presented in Alikhani et al. (2022), in which they perform (lexical) aspectual classification and zero-shot learning of aspect, with the corpus of Alikhani and Stone and pretrained word embeddings (fastText, mBERT, ELMo). They concluded that aspect can be predicted with distributional representations in a monolingual setup, but also learned even with cross-lingual information.

5.4 First round of experiments

5.4.1 Methodology

As discussed previously, lexical aspect is a feature of the verb but relies heavily on the predicate, the verb's dependents, and meaning on a sentence level. Thus, the task we decided would be most appropriate for identifying telicity and duration is *binary classification*. The process of using contextual word embeddings for sentence classification is streamlined with the use of the Huggingface library transformers; the pretrained model is loaded and has to be finetuned to the classification task. As mentioned in Section 2.4.4, finetuning is the strategy of adapting a pretrained model to a specific task, by adding an extra layer on top of the existing ones and specializing it on the given task. We are finetuning each transformer model for binary sequence classification of telicity and duration separately. Thus, we can exploit the existing model's knowledge from its contextual word embeddings, without the need for a large annotated corpus. We can also avoid large computational power and long training times. We are testing the accuracy of the finetuned model in predicting telicity and duration, both with quantitative measures and also with the manual evaluation of specific cases we created.

The telicity and duration annotations of Friedrich and Gateva annotate the sentence's verb as a carrier of telicity/duration, and we decided to use this additional information. We are finetuning each model in two ways: by providing an embedding that points to the position of the verb in the input sequence (by using the token_type_ids vectors when available, see Table 5.4), or by training only with the inputs and labels.

tokens	He	**worked**	well	and	earned	much	.	[SEP]
token_type_ids	0	1	0	0	0	0	0	0

Table 5.4: Tokens and corresponding token_type_ids vector of a sentence in our dataset. The sequence is followed by padding with special tokens/zeroes to 128 tokens.

5.4.2 Dataset

For this round of experiments, we used the gold- and silver-annotated datasets that have been developed and made publicly available by Friedrich and Gateva (2017)[1]. The gold annotations are based on the MASC dataset (Ide et al., 2008), while the silver annotations were crafted from the InterCorp parallel corpus of English and Czech (Čermák and Rosen, 2012), extracting the annotations from the Czech morphological markers of telicity and duration and applying them to the English translations. From the dataset, we extracted 6,354 sentences with their verbs annotated for telicity and/or duration. Table 5.5 presents the total number of sentences per tag. Telicity and duration-annotated sentences were used as two separate datasets. It is important to mention that, during our preprocessing of the dataset, we made note of the quality of annotations was questionable. The silver annotations were based on grammatical aspect in Czech, which should not be equated to lexical aspect as previously explained. Also, we noticed conflicts among the annotators in some sentences. For this round of experiments, we tried to not eliminate many problematic cases, as our dataset was already small. Besides, the dataset has been already successfully used in telicity classification—reportedly (Friedrich and Gateva, 2017; Peng, 2018).

In order to prepare the sentences for tokenization by the transformer models, we pre-processed them and fixed inconsistent annotations, e.g. annotations referencing a word form of the verb which did not exist in the sentence. As indicated for finetuning, we additionally padded, lower-cased, and truncated the sequences to 128 tokens. Since only one sentence in the whole dataset was longer than 128 characters and the annotated verb was in the first 128 tokens, this did not lead to issues with our dataset.

For finetuning and testing, we are splitting each dataset into training, validation,

[1] https://github.com/annefried/telicity

Type	Label	No. sentences	Training	Validation	Testing
telicity	telic	3,220	5,083	635	636
	atelic	3,134			
duration	stative	1,861	4,095	512	512
	dynamic	3,258			
TOTAL		**6,354**			

Table 5.5: Number of sentences tagged for telicity and duration in Friedrich and Gateva's dataset.

and test sets with a ratio of 80-10-10%, since our telicity and duration datasets are rather small. We have also prepared a second small testing dataset of 40 sentences annotated on telicity, and 40 sentences annotated on duration, evenly split between the telic-atelic tags and stative-durative. We hand-crafted these sentences, with the help of aforementioned bibliography and online resources[2]. A sample of the dataset is seen in Tables 5.6 and 5.7. The entirety of these datasets can be found in Section 5.7.1, Table 5.27 for telicity, and Table 5.28 for duration.

Finally, we prepared a third dataset of telicity-annotated sentences, aiming to create "minimal pairs" of telic-atelic sentences. The creation process was to form a sentence with a certain verb and degree of telicity, and then create another sentence with (preferably) the same verb and context, and the smallest amount of changes that could lead to a different degree of telicity. For example, the sentence "The boy is eating an apple." is telic, because the action is telic and the determined end is established by the direct object. However, in the sentence "The boy is eating apples." the action of the verb in present continuous implies a repetition or continuity, and the non-finite amount of the direct object does not force an endpoint to the action. The entire dataset of minimal pairs is presented in Section 5.7.1, Table 5.29.

[2]https://www.perfect-english-grammar.com/support-files/stative-verbs-list.pdf

label	sentence	label	sentence
telic	I **ate** a fish for lunch.	atelic	Cork **floats** on water.
telic	The cat **drank** all the milk.	atelic	The Earth **revolves** around the Sun.
telic	John **kicked** the door shut.	atelic	Kim is **singing**.
telic	I **opened** the juice bottle.	atelic	We **live** in a democratic age.

Table 5.6: A sample of the manually annotated sentences for telicity.

label	sentence	label	sentence
stative	She didn't **agree** with us.	durative	The boy **kicked** the ball hard.
stative	I don't **believe** the news.	durative	The dogs **bark** all night.
stative	Do you **hear** music?	durative	The snow **melts** every spring.
stative	This box **contains** a cake.	durative	I **slept** all morning.

Table 5.7: A sample of the manually annotated sentences for duration.

label	sentence	label	sentence
telic	I **drank** the whole bottle.	atelic	I **drank** juice.
telic	I read the book in an hour.	atelic	I read the book for an hour.
telic	The boy is **eating** an apple.	atelic	The boy is **eating** apples.
telic	I **put** on my red dress.	atelic	I **wore** my red dress.

Table 5.8: A sample of the manually annotated minimal pairs of telicity.

5.4.3 Models and finetuning

We are using the pretrained models from the Huggingface Python library transformers[3], and specifically the models for sequence (binary) classification. In preliminary experiments, we used the Python library simpletransformers;[4] even though it is endorsed by the transformers development team (Wolf et al., 2020), it did not offer the flexibility we needed, in order to compare the effect of the verb in the finetuning process. Therefore we followed the implementation of the transformers' team on finetuning the library models for sequence classification[5]. We are using selected models of the following transformer architectures: BERT, RoBERTa, XLNet, and ALBERT. These models have already been introduced in Section 2.5 and in Table 5.9 we list the models that we used and their hyperparameters.

[3]https://huggingface.co/models

[4]https://simpletransformers.ai/

[5]https://github.com/huggingface/transformers/blob/5bfcd0485ece086ebcbed2d008813037968a9e58/examples/run_glue.py

Model	Layers	Embedding size	Hidden	Heads	Hyperparemeters
bert-base-cased	12	-	768	12	109M
bert-base-uncased	12	-	768	12	110M
bert-large-cased	24	-	1024	16	335M
bert-large-uncased	24	-	1024	16	336M
roberta-base	12	-	768	12	125M
roberta-large	24	-	1024	16	355M
xlnet-base-cased	12	-	768	12	110M
xlnet-large-cased	24	-	1024	16	340M
albert-base-v2	12	128	768	12	11M
albert-large-v2	24	128	1024	16	17M

Table 5.9: The pretrained models and their parameters used for our experiments.

We finetune the models as Devlin et al. (2019) have recommended, with some modifications; we use a batch size of 32 and a learning rate of 2×10^{-5}. We apply dropout with probability p = 0.1 and weight decay with $\lambda = 0.01$. We use the PyTorch Adam as our optimizer (AdamW) without bias correction. We are finetuning each model for a maximum of 4 epochs, and we select the parameters between epochs 2-4 with the best accuracy, following the recommendation of Devlin et al. (2019) and McCormick and Ryan (2019) to train for 2-4 epochs for finetuning on a specific task. For base models, each training epoch took 25 minutes, and for large models 85 minutes, on one GPU system with CUDA acceleration of the IRIT computing cluster OSIRIM.

5.4.4 Results

5.4.4.1 Quantitative test set

The results are presented in Table 5.10 for telicity and Table 5.11 for duration. We calculated the accuracy metrics with the Python library scikit-learn (Pedregosa et al., 2011), which returns the metrics of precision, recall, and F1-score for the classification process and for each binary tag.

On classifying **telicity**, the best performing models were bert-base-cased and bert-large-cased. Overall, BERT models outperformed the other architectures significantly, and the large models were more successful than the base ones. RoBERTa models were moderately successful, although they were not able to make use of the verb position information in an additional embedding. Some XLNet and ALBERT models (xlnet-large-

cased, albert-base-v2, albert-large-v2) completely failed to classify the sentences and predicted the same label for all testing instances, switching their decision in each epoch. For our training objective with the addition of the verb position information in the sentence, the accuracy for most models (especially the more successful ones) improved significantly, e.g. $65\% \rightarrow 76\%$ for bert-base-cased, $68\% \rightarrow 79\%$ for bert-large-cased. However, there was rarely a significant improvement in the unsuccessful models, sometimes even a decline in accuracy.

Our findings on the best performing models in **duration** classification were similar to the ones on telicity. Despite the dataset being smaller and unbalanced, the BERT models performed overall better on this classification task. The bert-base models outperformed the bert-large ones (86% and $74 - 77\%$ respectively), while roberta-large, xlnet-large-cased and albert-large-v2 failed to make predictions. In this classification task, we noticed a significant improvement in accuracy when providing the model with the verb position information, especially in the leading models: $73\% \rightarrow 85\%$ for bert-base-cased, $74\% \rightarrow 84\%$ for bert-base-uncased.

5.4.4.2 Qualitative test sets

We examined further the incorrect predictions made by the models, focusing on BERT models since they were the ones consistently producing the best results. Instead of measuring accuracy with quantitative methods, this type of analysis permits the study of the type of errors that the models are prone to, thus revealing their weaknesses. The sentences that were consistently mislabeled by the BERT models can be found in Table 5.12. The classification errors in almost all the models could be found in certain specific sentences where the verb or verbal phrase has a strong preference of telicity, yet elements of the context define the temporal aspect of the sentence in the opposite manner. This could be due to a temporal prepositional phrase, for example, "I eat a fish for lunch on Fridays."; "eat" with a finite direct object would be considered telic, but the prepositional phrase "on Fridays" implies a degree of repetition, turning the action of the verb semelfactive (and therefore atelic). The presence of a grammatical tense is, as previously stated, not the vessel of lexical aspect, but can influence the temporal quality of the telic action; for example, in the sentence "The inspectors are always checking

every document very carefully." the continuous tense and the presence of the adverb "always" also render this sentence atelic. The adverb "always" was also ignored in the case of the atelic sentence "I always spill milk when I pour it in my mug.", leading to incorrect labeling.

The test set of minimal pairs of telicity provided an even better insight into the learned aspectual information of the model. The two sentences of the pair are as similar as possible, therefore the model may be biased towards a certain degree of telicity, which is respected by one sentence and not by the other. For example, the sentence "The boy is eating an apple." is a telic sentence, however, the presence of a continuous tense has misled all the models to incorrectly classify the sentence as atelic. Similarly, the sentence "The Prime Minister made that declaration for months." would be telic, if not for the presence of the prepositional phrase "for months". However, all models incorrectly classified this sentence. The atelic sentence "They have been building the house." has also been challenging for all models. We could hypothesize that the models give too much importance to the verb and not enough to the verb tense or the context, even if it contains prepositions and prepositional phrases of time.

Concerning the qualitative test set of duration, we observed fewer classification mistakes than the telicity questions, in all models. This improvement was expected since all models performed better on the duration classification task, and additionally because duration is harder to be misinterpreted or to be altered from a state to a durative action from context. In Table 5.13, the sentences that were consistently mislabeled by the BERT models are presented. The most commonly misinterpreted sentence by the models as *stative* was "She's playing tennis right now."; this was unexpected, as "play" is a verb of action in all of its possible meanings. However, the sentence "Do you hear music?" was incorrectly labeled as "durative" by some models, This may not be due to a deeper understanding of the state having an eventual endpoint, but due to frequent contexts dictating that "hear" has a short duration (but is, in fact, a state).

Model	Verb position	Accuracy	Precision	Recall	F1-score
bert-base-uncased	yes	0.72	0.72	0.72	0.72
	no	0.66	0.66	0.66	0.66
bert-base-cased	yes	0.76	0.76	0.76	0.76
	no	0.65	0.65	0.65	0.65
bert-large-uncased	yes	0.64	0.64	0.64	0.64
	no	0.66	0.66	0.66	0.66
bert-large-cased	yes	**0.79**	**0.79**	**0.79**	**0.79**
	no	0.68	0.68	0.68	0.68
roberta-base	no	0.64	0.64	0.64	0.64
roberta-large	no	0.66	0.66	0.66	0.66
xlnet-base-cased	yes	0.59	0.59	0.59	0.59
	no	0.61	0.61	0.61	0.61
xlnet-large-cased	yes	0.59	0.59	0.59	0.59
	no	0.51	0.26	0.51	0.34
albert-base-v2	yes	0.49	0.24	0.49	0.33
	no	0.6	0.6	0.6	0.6
albert-large-v2	yes	0.49	0.24	0.49	0.33
	no	0.49	0.24	0.49	0.33

Table 5.10: Results for the Friedrich and Gateva test set, for telicity classification.

Model	Verb position	Accuracy	Precision	Recall	F1-score
bert-base-uncased	yes	0.86	0.85	0.86	0.85
	no	0.71	0.71	0.71	0.71
bert-base-cased	yes	**0.86**	**0.86**	**0.86**	**0.86**
	no	0.7	0.7	0.7	0.7
bert-large-uncased	yes	0.77	0.77	0.77	0.77
	no	0.7	0.69	0.7	0.7
bert-large-cased	yes	0.74	0.73	0.74	0.73
	no	0.71	0.71	0.71	0.71
roberta-base	no	0.72	0.71	0.72	0.71
roberta-large	no	0.64	0.41	0.64	0.5
xlnet-base-cased	yes	0.7	0.69	0.7	0.68
	no	0.71	0.7	0.71	0.69
xlnet-large-cased	yes	0.64	0.41	0.64	0.5
	no	0.64	0.41	0.64	0.5
albert-base-v2	yes	0.8	0.8	0.8	0.78
	no	0.68	0.66	0.68	0.66
albert-large-v2	yes	0.64	0.41	0.64	0.5
	no	0.64	0.41	0.64	0.5

Table 5.11: Results for the Friedrich and Gateva test set, for duration classification.

label	sentence	bert-base-uncased		bert-base-cased		bert-large-uncased		bert-large-cased	
		yes	**no**	**yes**	**no**	**yes**	**no**	**yes**	**no**
telic	I ate a fish for lunch.		x		x				
telic	The cat drank all the milk.			x					
telic	The classes lasted one hour and took place twice a week over a four-week period.	x	x	x	x	x	x	x	x
telic	I can swim from one coast to another in less than an hour.	x	x			x	x	x	x
telic	Last night I slept like a baby.	x	x	x	x	x	x	x	x
telic	Jean was born in 1993 in Lyon.				x				
telic	The advancements in technology have changed the world.				x	x	x	x	x
atelic	I eat a fish for lunch on Fridays.						x	x	x
atelic	I always spill milk when I pour it in my mug.		x	x	x	x	x		x
atelic	The inspectors are always checking every document very carefully.					x	x	x	
atelic	I am working on a big project now.								x
atelic	The damage may last for many years.					x			
atelic	In the summer months James sleeps in every morning.			x	x				
atelic	Kim is writing a song.			x	x	x	x	x	x
atelic	Grandma is making pancakes for breakfast.			x	x		x	x	
atelic	He is constantly changing his script.			x					

Table 5.12: The sentences which were predicted with the wrong label of **telicity**, from the BERT models. The 'yes' and 'no' labels refer to whether the model was trained with the verb position vectors or not.

label	sentence	bert-base-uncased		bert-base-cased		bert-large-uncased		bert-large-cased	
		yes	**no**	**yes**	**no**	**yes**	**no**	**yes**	**no**
stative	Bread consists of flour, water and yeast.						x		
stative	Do you hear music?	x	x	x			x		
stative	I suppose John will be late.						x		
stative	I've known Julie for ten years.	x							
stative	The noise surprised me.				x				
stative	I didn't realise the problem.	x							
stative	I suppose John will be late.		x					x	
durative	She plays tennis every Friday.				x				x
durative	She's playing tennis right now.	x	x		x	x		x	x
durative	The snow is melting right now.		x		x	x			
durative	We talked for hours on our trips.		x			x			x
durative	She runs ten kilometers a day.		x						
durative	The dogs bark all night.		x						
durative	He grew potatoes in his farm.				x	x			
durative	I slept all morning.	x							
durative	She runs ten kilometers a day.	x				x			
durative	They ate their dinner in silence.							x	

Table 5.13: The sentences which were predicted with the wrong label of **duration**, from the BERT models. The 'yes' and 'no' labels refer to whether the model was trained with the verb position vectors or not.

label	sentence	bert-base-uncased		bert-base-cased		bert-large-uncased		bert-large-cased	
		yes	**no**	**yes**	**no**	**yes**	**no**	**yes**	**no**
atelic	I drank juice.	x							
telic	She fell asleep at 8 pm.		x						
telic	I read the book in an hour.		x						
atelic	I stopped reading the book at 5 pm.	x	x	x	x	x		x	x
atelic	I will receive new stock on [UNK].	x		x	x	x	x	x	x
telic	She ate that sandwich.								
telic	**The boy is eating an apple.**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
atelic	She has been eating that sandwich.						x		x
telic	She noticed him.		x	x	x	x	x	x	x
atelic	She looked at him.								x
atelic	She slept at 8 pm.								x
atelic	The artist studies a painting.						x	x	
telic	The girl walked a kilometer yesterday.	x	x						
atelic	The girl walked yesterday.		x	x	x		x		x
atelic	The hunters chased the deer.			x					
atelic	The hunters chased the deer.								x
telic	The pond is freezing over.		x	x	x				
atelic	**The Prime Minister made that declaration for months.**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	
telic	The workers painted the house in an hour.		x						x
atelic	The workers painted the house for an hour.			x		x	x	x	
atelic	**They have been building the house.**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**

Table 5.14: The sentences of **minimal pairs** which were predicted with the wrong label of **telicity**, from the BERT models. The 'yes' and 'no' labels refer to whether the model was trained with the verb position vectors or not.

5.4.4.3 Additional experiments: Attention Masks

Inspired by the attention mask experiments of the selectional preferences experiment in Section 4.1, we conducted a final experiment with classification and the use of attention masks. Since the verb position was already indicated, and since the verb possesses lexical aspect, we could create attention masks that allow attention only to the verb or obstruct the verb from the attention mechanism. The attention masks were applied to the testing input sentences, not to training input sequences, therefore the same finetuned classification models were used as previously. We experimented only with the BERT models since they were the most successful.

When the models were asked to classify telicity or duration only attending to the verb of the sentence, the classification results drastically declined (see Tables 5.15 and 5.16). This demonstrates that the finetuned models additionally take into consideration the context and the dependents of the verb, in order to classify the telicity and duration properties of the verb since they can do better predictions when this information is available. However, it is not clear whether this is information acquired from the pretrained embeddings or the finetuning process.

Model	Verb position	Accuracy	Precision	Recall	F1-score
bert-base-uncased	yes	0.50	**0.76**	**0.75**	**0.75**
	no	**0.52**	0.53	0.52	0.52
bert-base-cased	yes	0.50	0.49	0.50	0.49
	no	**0.51**	**0.51**	**0.51**	**0.51**
bert-large-uncased	yes	0.49	0.48	0.48	0.48
	no	**0.53**	**0.53**	**0.53**	**0.53**
bert-large-cased	yes	0.51	0.51	0.51	0.51
	no	**0.53**	**0.53**	**0.53**	**0.53**

Table 5.15: The classification results for **telicity** for the Friedrich and Gateva test set, with BERT models with an attention mask on the context.

Model	Verb position	Accuracy	Precision	Recall	F1-score
bert-base-uncased	yes	**0.46**	**0.58**	**0.46**	**0.51**
	no	0.45	0.54	0.45	0.49
bert-base-cased	yes	0.38	**0.66**	0.38	0.48
	no	**0.46**	0.54	**0.46**	**0.50**
bert-large-uncased	yes	**0.60**	0.48	**0.60**	**0.53**
	no	**0.60**	**0.51**	**0.60**	0.55
bert-large-cased	yes	0.45	**0.58**	0.45	**0.51**
	no	**0.46**	0.50	**0.46**	0.48

Table 5.16: The classification results for **duration** for the Friedrich and Gateva test set, with BERT models with an attention mask on the context.

5.5 Second round of experiments

5.5.1 Methodology

The first round of experiments on lexical aspect offered us some interesting insights, but more experiments were required in order to understand the extent of lexical aspect knowledge in contextual word embeddings. The occasional failure of the RoBERTa, XL-Net, and ALBERT-based models to perform binary classification, even in the relatively easy task of duration, was puzzling and led us to conduct a second round of experiments, with the same methodology but improved training datasets. We performed additional experiments, with the dataset of Friedrich and Gateva (2017) and additionally the dataset of Alikhani and Stone (2019). We explored the inner workings of the attention mechanism and the capacities of the pretrained word embeddings. Additionally, we recreated our finetuning and classification experiments in French, with the translated versions of the English datasets (with adaptations to the qualitative sets, when needed).

5.5.2 Datasets in English

Similarly to the first round of experiments, the telicity and duration-annotated sentences were used as separate datasets for separate experiments. We used the dataset of Friedrich and Gateva (2017) as previously, and we added sentences from the "Caption" dataset of Alikhani and Stone (2019). This addition of sentences allowed us to remove many problematic sentences from the first dataset: multiple occurrences of the same sentence, annotations with conflicting labels, and sentences that were too long to be useful.

The "Captions" dataset[6] (Alikhani and Stone, 2019) was built from five image–text caption corpora, with the intention to study inference between sentences. Based on the aspect of the verb and the context, it has been annotated with human annotators for telicity (telic/atelic) and duration (stative/durative/punctual). Even though the focus of the original work was on the head verb of each sentence, the verbs were not separately annotated in the original study. However, the sentences were all short descriptions of an image, with a simple syntactic structure usually containing one verb. As a result, we employed dependency parsing with spaCy (Honnibal et al., 2020) to extract the verb and

[6] https://github.com/malihealikhani/Captions

its position for our studies. Since there were not enough sentences to support a third category with the *punctual* label, we eliminated these sentences. We also found some annotation errors that we fixed.

In Table 5.17 we present the sizes of the two datasets and our final dataset. We split this dataset into training, validation, and test sets with a ratio of 80-10-10%. We are also using the qualitative datasets from the first round of experiments, for telicity (Table 5.27), duration (Table 5.28), and minimal pairs for telicity (Table 5.29). In addition, we created an extra test set for telicity, with variations of some challenging sentences, by changing the verb tense or the prepositional phrase word order, to the extent that English permits. This could allow an even deeper insight into the context's importance in deciding the telicity degree of a sentence — possibly leading to mistakes. This additional dataset can be found in Table 5.30.

Type	Label	Friedrich and Gateva	Captions	Current	Total
telicity	telic	1,831	785	2,885	**6,173**
	atelic	2,661	1,256	3,288	
duration	stative	1,860	419	2,036	**4,081**
	durative	38	1,843	2,045	
	punctual	-	355	-	

Table 5.17: Number of sentences and annotations in each dataset, and our final dataset sizes.

5.5.3 Improvements on technical methods

We are using the same models and the same finetuning setting as the first round of experiments, with improvement on the overall technical methodology of our work. First of all, it was necessary to establish baselines with non-transformer methods, in order to test how difficult the classification task of telicity and duration is, for traditional NLP methods, and compare the transformers to a trusted baseline. We used two standard binary classification models trained and tested on the same sets as the transformers. First, a simple bag-of-words logistic regression model, implemented with the Python library scikit-learn (Pedregosa et al., 2011) with default parameters and data scaling. Then, a one-layer convolutional neural network model (CNN) implemented with Py-

torch (Paszke et al., 2019) and trained for 50 epochs, which is commonly used for text classification tasks (Kim, 2014). The CNN model is trained without pretrained word embeddings (such as word2vec, fastText), embedding dimension of 300, filter size of $[3, 4, 5]$, 100 filters per dimension, a dropout rate of 0.5, a learning rate of 0.01 and the Adadelta optimizer.

Despite the criticism on the efficacy of finetuning, we tried to ensure, as much as possible, the success of our methodology. We are aware that our training sets were small (6K for telicity, 4K for duration). We took into consideration the proposition of Dodge et al. (2020) to not shuffle the datasets before splitting them into the train, test, and validation sets. Instead of using BERTadam which was proposed by Devlin et al. (2019) and Wolf et al. (2020), we are using the PyTorch Adam as our optimizer as recommended by Zhang et al. (2020), because they report that BERTadam omits debiasing, and directly uses the biased estimates in the parameters update. Unlike the proposition of Dodge et al. (2020) and Mosbach et al. (2020) for multiple training epochs, which has not been definitively proven beneficial for all tasks (Zhang et al., 2020), we followed the advice of Devlin et al. (2019) and McCormick and Ryan (2019) for fewer training epochs and picking the best epoch based on validation results. We did however perform several training runs, with the same (80%) or more (90%) or fewer (75%) training data, and we did not notice significantly different behaviors or results in loss or validation metrics.

While studying the classification results, we also examined the probability distribution of the predicted labels. In addition, we planned some smaller experiments, in order to observe how the context is interpreted and attended to by the model—based on previous work by Clark et al. (2019) and Subudhi (2019). Finally, moving from the finetuned results, we explored the knowledge of the pretrained contextual word embeddings on lexical aspect with a classification experiment.

5.5.4 Results for English

5.5.4.1 Quantitative results

In this section, the results of the finetuning experiment of telicity and duration classification in English are presented. The results for telicity are found in Table 5.18 and for duration in Table 5.19. The probability scores are found in Figures 5.1 and 5.2 respec-

tively.

On classifying **telicity**, bert-large-cased model had the best performance. In general, BERT models performed better in this round of experiments as well, although all models showed relatively high accuracy of $> 80\%$, compared to the 64% accuracy of the logistic regression baseline and the 75% of the CNN baseline. Accuracy increased for all models ($+1/4\%$) when trained with the additional knowledge of the verb location in the sentence, but not statistically significant. Looking at the probability distributions of the predicted labels, the BERT models, both base and large, were the most confident in their predictions, with the probability of each label being 90%, whereas the large versions of other models were the ones whose probability distribution included more instances with lower label probability. Overall, the models were more confident when making accurate predictions, and only slightly less confident when making incorrect predictions (with a few labels closer to $40 - 60\%$ but still a majority above 90%).

Our results on classifying **duration** were comparable to those for classifying **telicity**, with the models generally performing better on this classification task despite the smaller dataset –even the CNN baseline performed very well, with 88% accuracy. Although all models reached an accuracy of 93%, the BERT models were the most successful, with an accuracy of up to 96%. Since most models either improve or worsen by $\pm 1\%$ in this classification task, it is difficult to determine the impact of using verb position information. Regardless of how accurate they were, all models were quite confident in categorizing phrases and had high confidence in both correct and wrong predictions when looking at the probability distribution of the two labels (erroneously).

Model	Verb position	Accuracy	Precision	Recall	F1-score
bert-base-uncased	yes	0.86	0.86	0.86	0.86
	no	0.81	0.81	0.81	0.81
bert-base-cased	yes	0.87	0.87	0.87	0.87
	no	0.81	0.80	0.80	0.80
bert-large-uncased	yes	0.86	0.86	0.86	0.86
	no	0.81	0.80	0.80	0.80
bert-large-cased	yes	**0.88**	**0.87**	**0.87**	**0.87**
	no	0.81	0.81	0.80	0.80
roberta-base	no	0.84	0.84	0.84	0.84
roberta-large	no	0.80	0.81	0.79	0.79
xlnet-base-cased	yes	0.82	0.82	0.82	0.82
	no	0.81	0.81	0.81	0.80
xlnet-large-cased	yes	0.82	0.82	0.82	0.82
	no	0.80	0.80	0.80	0.80
albert-base-v2	yes	0.84	0.84	0.84	0.84
	no	0.81	0.80	0.80	0.80
albert-large-v2	yes	0.80	0.80	0.80	0.80
	no	0.82	0.81	0.81	0.81
CNN (50 epochs)	no	0.75	0.75	0.75	0.75
Logistic Regression BoW	no	0.61	0.61	0.61	0.61

Table 5.18: Results of classification accuracy on the telicity test set.

Model	Verb position	Accuracy	Precision	Recall	F1-score
bert-base-uncased	yes	**0.96**	**0.96**	**0.96**	**0.96**
	no	0.94	0.94	0.94	0.94
bert-base-cased	yes	**0.96**	**0.96**	**0.96**	**0.96**
	no	0.96	0.95	0.96	0.96
bert-large-uncased	yes	**0.96**	**0.96**	**0.96**	**0.96**
	no	0.95	0.95	0.94	0.94
bert-large-cased	yes	**0.96**	**0.96**	**0.96**	**0.96**
	no	0.95	0.95	0.95	0.95
roberta-base	no	0.95	0.95	0.95	0.95
roberta-large	no	0.95	0.95	0.95	0.95
xlnet-base-cased	yes	0.94	0.94	0.94	0.94
	no	0.95	0.95	0.95	0.95
xlnet-large-cased	yes	0.94	0.94	0.94	0.94
	no	0.95	0.95	0.95	0.95
albert-base-v2	yes	0.95	0.95	0.95	0.95
	no	0.95	0.95	0.95	0.95
albert-large-v2	yes	**0.96**	**0.96**	**0.96**	**0.96**
	no	**0.96**	**0.96**	**0.96**	**0.96**
CNN (50 epochs)	no	0.88	0.88	0.88	0.88
Logistic Regression BoW	no	0.70	0.70	0.69	0.69

Table 5.19: Results of classification accuracy on the duration test set.

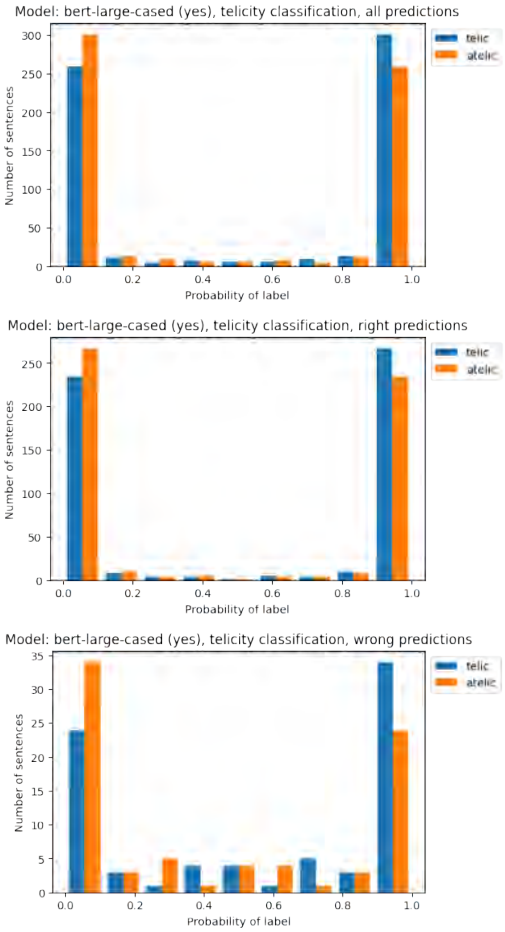


Figure 5.1: Probability distribution for the telicity labels, for the most successful model (bert-large-cased with verb position).

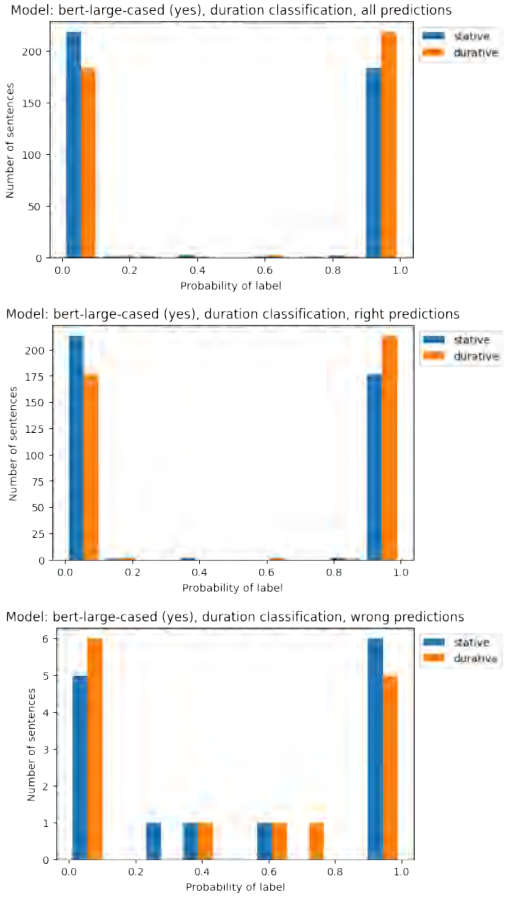


Figure 5.2: Probability distribution for the duration labels, for the most successful model (bert-large-cased with verb position).

5.5.4.2 Qualitative results and analysis

As previously stated, in order to examine aspectual features outside the scope of classification measures, we additionally produced our own annotated datasets of telicity and duration. The cases that were easier or more challenging for the models to categorize were identified by a deeper examination of the right and inaccurate predictions made by them. Our objective was to manually examine the models' strengths and weaknesses in challenging and contradictory classification cases, hence the smaller qualitative datasets and the inclusion of the most interesting examples.

For **telicity**, overall, models were quite successful in classifying the sentences of our qualitative dataset, and they were more successful than the models of the first round of experiments thanks to our improvements on the dataset and methodology. However, we noticed that the common mistakes came from a preference of the models for the *atelic* label than the *telic*. For example, all models were able to identify that sentences with statements are atelic, such as "Cork floats on water." and "The Earth revolves around the Sun.", but they mistakenly labeled the sentence "The advancements in technology have changed the world." as atelic as well. Sentences with an action were correctly classified almost all the time: "I spilled the milk." was correctly classified as *telic*, and "I always spill milk when I pour it in my mug." was also correctly classified as *atelic* (except for the xlnet models). However, the sentences "Yesterday I ran a mile in under 10 minutes.", "The classes lasted one hour and took place twice a week over a four-week period.", and "Louise made the biggest progress of everyone this year." were almost always incorrectly classified as atelic, despite the presence of prepositional phrases of time and past tenses. Some sentences with conflicting verbal lexical aspect and context, such as "I eat a fish for lunch on Fridays." (atelic) and "The inspectors are always checking every document very carefully." (atelic) were still incorrectly annotated by the finetuned models, as they were in the first round of experiments.

Moving to our minimal pairs of telic-atelic sentences, we observe that, in most cases, most models are able to classify correctly a sentence based both on the verb action and the context. The sentences "I drank the whole bottle." and "I drank juice." were properly categorized as telic and atelic, respectively, despite the existence of the identical verb and tense. However, the sentence "The cat drank all the milk." was mistakenly categorized

by all the models in our qualitative dataset as *atelic*. Another intriguing error we found was classifying the pair "The boy is eating an apple." and "The boy is eating apples." as *atelic*; in the first, the action is telic, but the tense is continuous for pragmatic reasons.

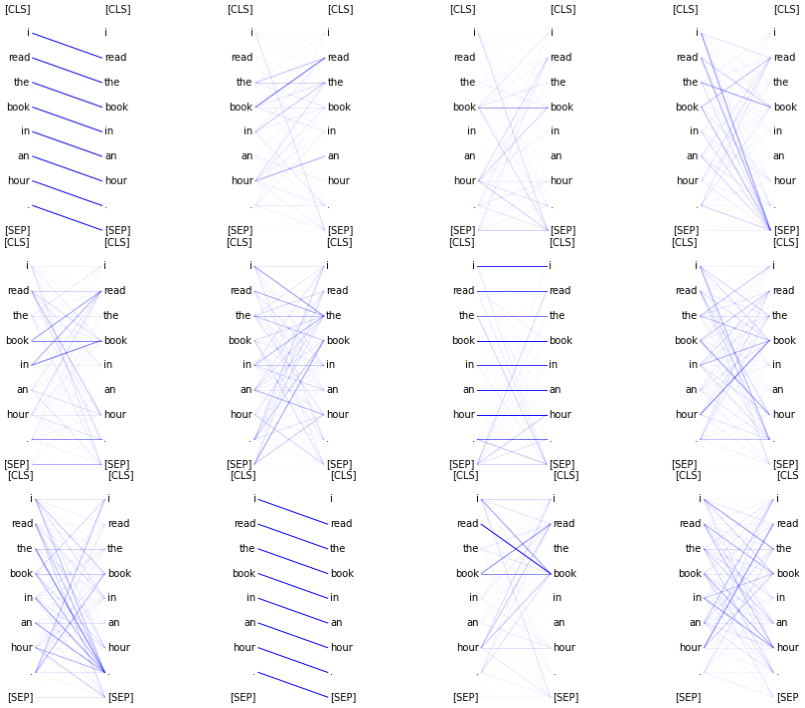
In order to observe specific tenses, word positions, and context more extensively, we can examine the variations of sentences (some of which were already challenging cases of the previous test sets). Word order did not affect the models' wrong predictions in some sentences, such as "I ate a fish for lunch at noon." and "The classes lasted one hour and took place twice a week over a four-week period.". In some complex cases, such as the sentence "The Prime Minister made that declaration for months." we notice that most models fail to classify it as atelic in all its variations, except for when the prepositional phrase is at the start and the tense is present perfect continuous ("has been making"). We noticed that even sentences with a more obvious degree of telicity ("John Wilkes Booth killed Lincoln on 1865." – telic) were sometimes labeled incorrectly when the prepositional phrase was at the end rather than the start. However, the presence of a perfective tense over past simple, especially past perfective, led the models to correctly label the specific variations of telic sentences, e.g. "Louise (had) made the biggest progress out of everyone this year."

Regarding **duration**, the models were less successful at classifying stative sentences than durative. Statements such as "I like reading detective stories.", "I love chocolate.", "I prefer chocolate ice cream." were incorrectly labeled as *durative* by almost all the models. However, statements such as "I disagree with you." were correctly classified. Another weakness was the sentences with world knowledge and facts, which tend to be *stative*, even some sentences with intransitive verbs, such as "Bread consists of flour, water and yeast." and "This cookbook includes a recipe for bread.". Durative sentences, despite verb tense and context, were almost always correctly classified, e.g. "She plays tennis every Friday." and "She's playing tennis right now.". Some notable examples are "The noise surprised me." and "He screamed for help.", incorrectly classified as *stative* by xlnet and albert models, while "Do you hear music?" was classified as *durative*.

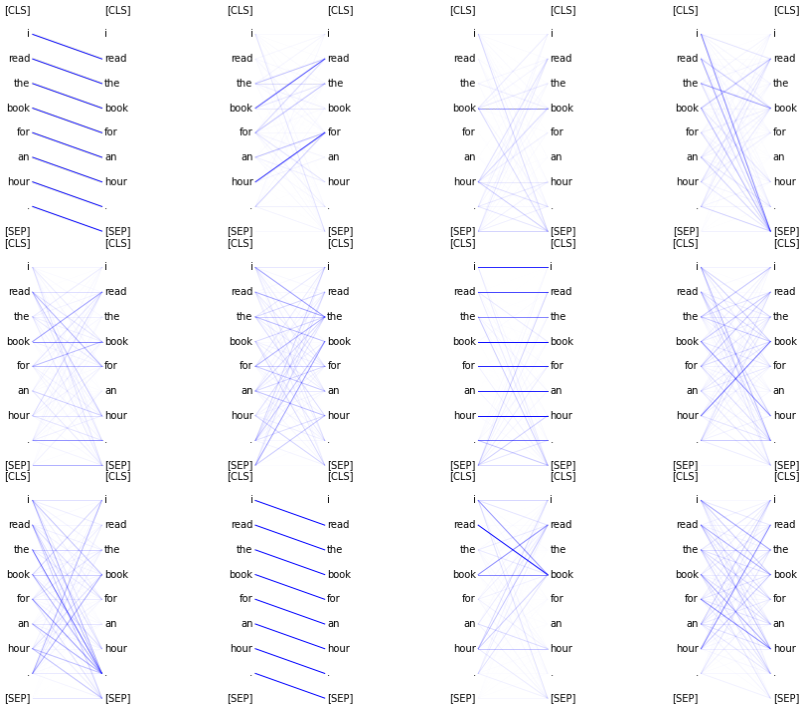
5.5.4.3 Additional experiments: A look at attention

It has been extensively discussed how the analysis of the self-attention mechanism has been a preferred method of explaining transformer architectures' results and abilities. For our analysis, we studied sentences from the qualitative test sets and their visualizations per layer and per attention head. We observed that, among the models we tested in our experiments, BERT models in earlier levels exhibited more "focused" attention to particular tokens and "diffused" attention on later layers. Meanwhile, the other architectures exhibited more "diffused" attention even from earlier layers. Most tokens attended to all tokens or to the special tokens in the final layers (the special tokens of start and end of the sequence). In Figure 5.3, we are comparing a minimal pair of telicity, on layer 3 of the bert-base-uncased model (with information on verb position): "I read the book in an hour." (telic) and "I read the book for an hour." (atelic).

The most interesting were the earlier layers, while the middle layers specialized on syntactic dependencies (verb attended to subject and object, prepositional phrase attended to its tokens) and the last layers did not focus saliently on any word tokens. For the presented example, there was a tendency for the verb to slightly attend more to "for" than "in", before moving to focus on its subject and direct object. Looking even closer at the attention of the verb token in Figure 5.4, we located this tendency in head 4 but generally, the verb prefers to pay attention to its neighboring words and its closer syntactic dependents.

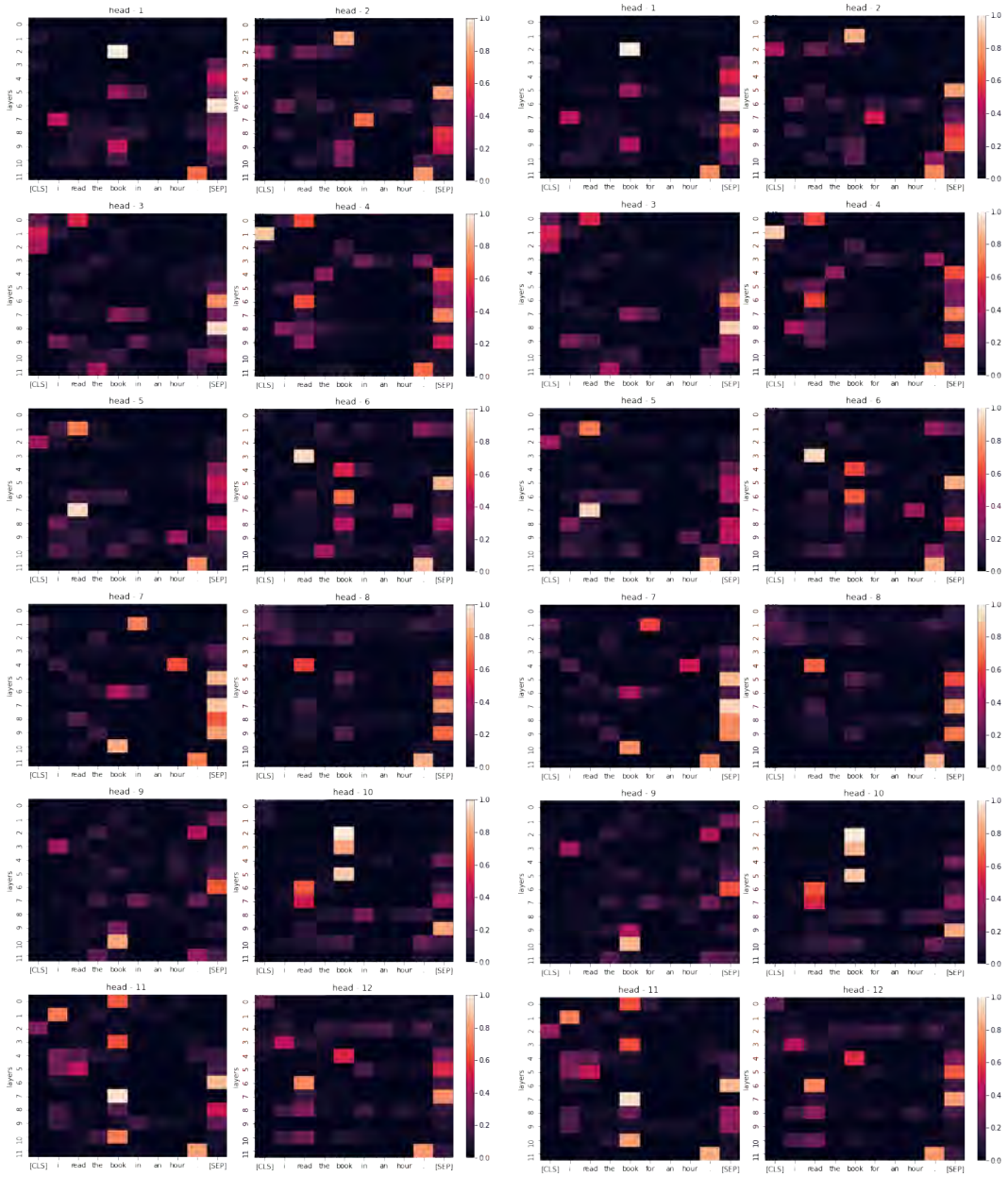


"I read the book in an hour."



"I read the book for an hour."

Figure 5.3: Visualization of attention for the sentences "I read the book in an hour." (telic) and "I read the book for an hour." (atelic), from the model bert-base-uncased (with verb position information), on the 3rd layer of the model, for all heads (1-12).



"I read the book in an hour." "I read the book for an hour."

Figure 5.4: Visualization of attention of the verb token to all other sentence tokens (x axis), from the model bert-base-uncased (with verb position information), on all layers (y axis), for all heads (per plot).

5.5.4.4 Additional experiments: Classification with layer embeddings and logistic regression

The contextualized word embeddings of pretrained models already include linguistic information, that could be extracted and employed with task-specific models for classification. The wide use of these models is a direct outcome of the embeddings' learned knowledge since they enable quicker computations and access to this information without the need for specialization (even though specialization with finetuning is recommended). In order to determine how much knowledge has been already been acquired in the pretraining process by each layer of a transformer model about lexical aspect, we extracted the contextual word embedding (for the annotated verb) from each layer and trained a logistic regression model to classify telicity and duration. The inspiration for these experiments came from probing experiments, such as Jawahar et al. (2019); Coenen et al. (2019).

In Figure 5.5, we present the accuracy for each layer of the *base* models. Similar to the performance of the finetuned models, models were successful up to 79% for telicity classification and up to 90% for duration classification. However, as the layers increase, accuracy does not increase proportionally. For example, for telicity, some models attain high accuracy in the intermediate layers, then again in the last layers, with accuracy occasionally declining in the final layer.

5.5.4.5 Additional experiments: Unseen verbs

In our training and test datasets, there was a large variety of verbs (as the root of sentence), which allowed us to test the classification success on sentences where the verb has not been observed by the model. For telicity, 267 verb forms which were the head of their phrase were not "seen" by the model in the training set (and 146 of them were not split into subwords), and for duration, 117 verbs (and 80 not split). We investigated whether the associated sentences had been erroneously labeled, as well as the average likelihood of the assigned label in the models. Overall, a few phrases were classified wrongly for both classification tasks, with labels of either category. This implies that even when the verb form has not been detected by the model, context plays a significant impact in the decisions made.

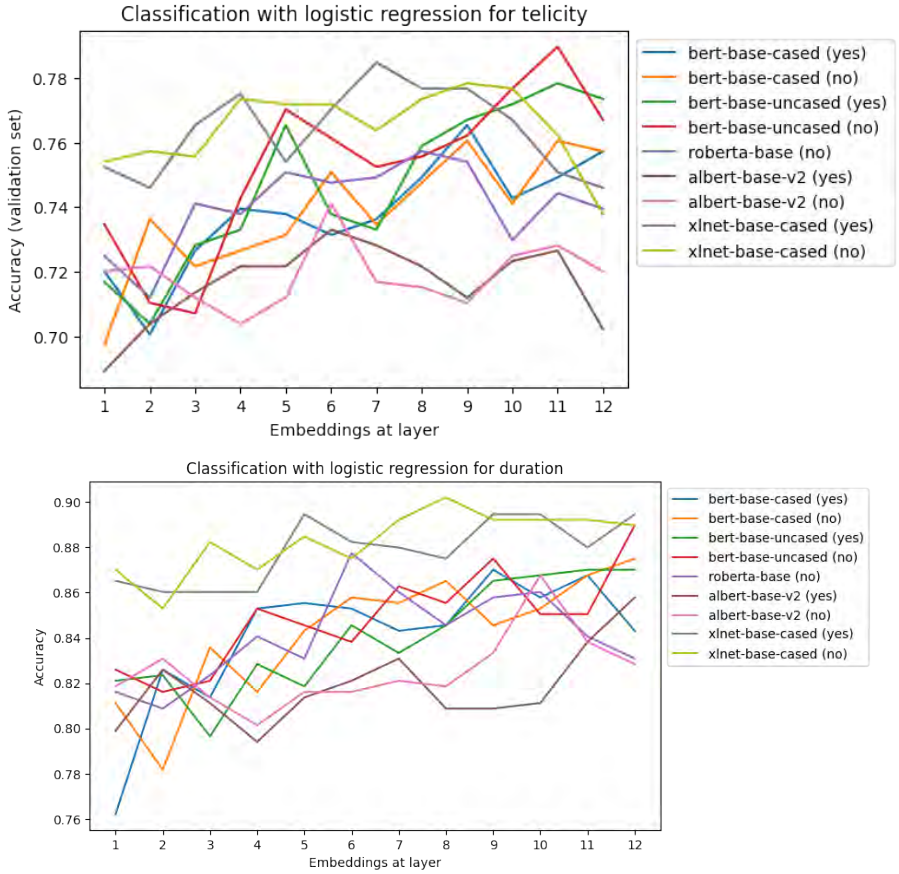


Figure 5.5: Accuracy of classification of logistic regression, per layer of embeddings, for base models.

Model	Verb	Telicity						Duration					
		Seen verbs			Unseen Verbs			Seen verbs			Unseen Verbs		
		Correct	Wrong	Acc.	Correct	Wrong	Acc.	Correct	Wrong	Acc.	Correct	Wrong	Acc.
bert-base-uncased	yes	1286	240	0.84	180	41	0.81	681	26	0.96	142	6	0.96
	no	1194	336	0.78	170	50	0.77	678	29	0.96	143	5	**0.97**
bert-base-cased	yes	1290	218	**0.86**	169	31	**0.85**	665	17	**0.98**	129	5	0.96
	no	1169	342	0.77	162	37	0.81	661	21	0.97	128	6	0.96
bert-large-uncased	yes	1292	234	0.85	190	31	0.86	687	20	0.97	142	6	0.96
	no	1191	339	0.78	177	43	0.8	688	19	0.97	143	5	**0.97**
bert-large-cased	yes	1308	200	0.87	168	32	0.84	666	16	**0.98**	128	6	0.96
	no	1167	344	0.77	153	46	0.77	667	15	**0.98**	127	7	0.95
roberta-base	no	1243	291	0.81	185	41	0.82	662	19	0.97	126	8	0.94
roberta-large	no	1157	377	0.75	176	50	0.78	667	14	**0.98**	127	7	0.95
xlnet-base-cased	yes	1196	327	0.79	174	43	0.8	651	30	0.96	127	8	0.94
	no	1175	350	0.77	171	45	0.79	656	25	0.96	129	6	0.96
xlnet-large-cased	yes	1190	333	0.78	174	43	0.8	653	28	0.96	127	8	0.94
	no	1182	343	0.78	169	47	0.78	652	29	0.96	125	10	0.93
albert-base-v2	yes	1281	271	0.83	186	44	0.81	698	16	**0.98**	138	5	**0.97**
	no	1194	362	0.77	187	42	0.82	696	18	0.97	137	6	0.96
albert-large-v2	yes	1204	348	0.78	174	56	0.76	690	24	0.97	137	6	0.96
	no	1212	344	0.78	184	45	0.8	698	16	**0.98**	137	6	0.96

Table 5.20: The results on the test set, for sentences with seen/unseen verbs in the training set, for telicity and duration. The ratio of correct/incorrect labels is similar, with seen and unseen verbs, both for telicity and duration.

5.5.5 Telicity and duration classification in French

Additionally, we were interested in determining whether transformer models could classify telicity and duration in a different language. French morphology differs from English enough to constitute an interesting antagonist to the English experiments. French verb tenses are formed and employed in a different manner; for example, the inflected forms of present simple and present continuous in French are the same, while English has two separate forms (see Table 5.21). However, French does not include morphological markers of perfectivity as Czech does, thus the lexical aspect remains "morphologically hidden" in the semantics of the verb and the context.

The added benefit of using French is that it is a high-resource language for NLP and there exist monolingual transformer models ready for use in the transformers library.

Tense	French	English
Present simple	Il parle.	He speaks.
Present continuous	Il parle. / Il est en train de parler.	He is speaking.
Present with emphasis	Il parle.	He does speak.

Table 5.21: Examples of how present simple and present continuous are homographs in French, while English differentiates between the two — with an additional structure for emphasis. However, there is a construction in French that is used to express continuity, which is not included in the conjugation paradigm of verbs: *Il est en train de parler.* "He is in the process of speaking.".

Since there are no French datasets with annotations of telicity and duration, we used the DeepL translator[7] to translate our English datasets and manually reviewed 200 sentences from the datasets to ensure that the translation and annotation were accurate. Our average accuracy rating for the machine-translated sentences was 88.6%, whereas the annotated label accuracy rating was 73.5%. In order to train the classifier with the verb position information, we also extracted the verb-head word from each sentence using the spaCy dependency parser. However, because dependency parsers are flawed, the compound verb tenses of French have led to many mistakes. Therefore, we decided to only perform the classifying experiment, and not the additional experiments we per-

[7] https://www.deepl.com/translator

formed for English.

We use the resulting datasets to finetune the French transformer models, and assess their abilities in aspectual classification. We are using the two monolingual French transformer models available from the transformers library, CamemBERT (Martin et al., 2020) and FlauBERT (Le et al., 2020). Details on the models can be found in Section 2.5.8.

In addition, we manually translated our qualitative test sets and made appropriate changes (when verb tense did not convey the desired telicity, for example), and in lieu of the English sentences on variations of word order and verb tense, we created more minimal pairs with variations on prepositional phrases. These qualitative sets can be found in Table 5.31 for the telicity set, Table 5.32 for duration, Table 5.33 for the minimal pairs and Table 5.34 for the additional sentences. A sample is shown in Tables 5.22 and 5.24 (for telicity) and 5.23 (for duration).

label	**sentence**	**label**	**sentence**
telic	J'ai **renversé** le lait.	atelic	Le liège **flotte** sur l'eau.
telic	Kim a **écrit** une chanson.	atelic	La Terre **tourne** autour du Soleil.
telic	J'ai **accroché** le tableau au mur.	atelic	Kim **chante**.
telic	La soupe a **refroidi** en une heure.	atelic	Jean **regarde** la télévision.

Table 5.22: A sample of the manually annotated sentences for telicity.

label	**sentence**	**label**	**sentence**
stative	Le bruit m'a **surpris**.	durative	Elle **joue** au tennis tous les vendredis.
stative	Cette chemise me **va** bien.	durative	Elle **joue** au tennis en ce moment.
stative	Je **connais** Julie depuis dix ans.	durative	Ils ont **mangé** leur dîner en silence.
stative	Cette boîte **contient** un gâteau.	durative	Il a **crié** à l'aide.

Table 5.23: A sample of the manually annotated sentences for duration.

label	**sentence**	**label**	**sentence**
telic	Le garçon mange une pomme.	atelic	Le garçon mange des pommes.
telic	J'ai **bu** toute la bouteille.	atelic	J'ai **bu** du jus de fruit.
telic	Les chasseurs ont **chassé** le cerf.	atelic	Les chasseurs **chassaient** le cerf.
telic	J'ai mis ma robe rouge.	atelic	Je portais ma robe rouge.

Table 5.24: A sample of the additional manually annotated sentences for telicity.

5.5.6 Results for French

5.5.6.1 Quantitative analysis

The results of the classification for telicity and duration are presented in Tables 5.25 and 5.26. Overall, accuracy is lower than in English. The models performed better than the CNN classifier baseline, but marginally. However, the fact that the extra verb position information was nearly always harmful is likely a fault with dependency parsing incorrectly identifying an auxiliary verb as the main verb, since French uses compound tenses more frequently than English does.

Model	Verb position	Accuracy	Precision	Recall	F1-score
camembert-base	no	**0.77**	0.77	0.78	0.77
camembert-large	no	0.76	0.77	0.77	0.77
flaubert-small-cased	yes	0.69	0.70	0.70	0.69
	no	0.73	0.73	0.73	0.72
flaubert-base-uncased	yes	0.74	0.75	0.74	0.72
	no	0.76	0.76	0.76	0.75
flaubert-base-cased	yes	0.76	0.76	0.77	0.76
	no	0.77	0.78	0.78	0.78
flaubert-large-cased	yes	0.73	0.74	0.74	0.72
	no	0.75	0.76	0.76	0.74
CNN (50 epochs)	no	0.71	0.69	0.65	0.65
Logistic Regression BoW	no	0.61	0.59	0.59	0.59

Table 5.25: Results for telicity classification in French.

Model	Verb position	Accuracy	Precision	Recall	F1-score
camembert-base	no	0.82	0.82	0.82	0.82
camembert-large	no	**0.87**	0.87	0.87	0.87
flaubert-small-cased	yes	0.79	0.79	0.79	0.79
	no	0.81	0.81	0.81	0.8
flaubert-base-uncased	yes	0.80	0.81	0.80	0.80
	no	0.84	0.84	0.84	0.84
flaubert-base-cased	yes	0.81	0.82	0.82	0.81
	no	0.83	0.83	0.83	0.83
flaubert-large-cased	yes	0.81	0.81	0.81	0.80
	no	0.87	0.87	0.87	0.87
CNN (50 epochs)	no	0.80	0.82	0.82	0.82
Logistic Regression BoW	no	0.68	0.68	0.67	0.67

Table 5.26: Results for duration classification in French.

5.5.6.2 Qualitative analysis

The French finetuned models performed better on the qualitative sets than their English counterparts, avoiding the English models' common mistakes such as classifying the atelic sentence *Je mange un poisson à midi le vendredi.* "I eat a fish for lunch on Fridays." as telic. Unlike the English models that classified incorrectly mostly telic sentences, the French models' fewer mistakes occurred in the classification of both labels. However, there were still some interesting mistakes in the models' performance that were not common for English. For example, the sentences *Je renverse toujours le lait quand je le verse dans ma tasse.* "I always spill milk when I pour it in my mug." (atelic) and *Jenny a travaillé comme médecin toute sa vie.* "Jenny worked as a doctor her whole life." (atelic) were incorrectly classified as telic, perhaps due to the verb of the sentence. The sentence *Les cours duraient une heure et avaient lieu deux fois par semaine sur une période de quatre semaines.* "The classes lasted one hour and took place twice a week over a four-week period." (telic) was challenging both for the English and the French models, regardless of the presence of a continuous verb tense in either of those languages, because of its length and the presence of multiple verbs and temporal descriptors in the sentence.

Comparing minimal pairs, we notice that, unlike in English, the sentence *J'ai bu du jus de fruit.* "I drank juice." (atelic) was frequently marked as telic by the models, and so did its pair *J'ai bu toute la bouteille.* ("I drank the whole bottle." – telic). And unlike the common mistake of marking both sentences as telic in English, the French models marked the sentences *Le garçon mange [une pomme/des pommes].* ("The boy is eating [an apple/apples]) both as atelic.

For the duration classification, similarly to the English models, we observe that stative sentences were the ones which were occasionally or always incorrectly classified by the models; sentences with statements such as *Le pain est composé de farine, d'eau et de levure.* ("Bread consists of flour, water and yeast.") or *J'aime le chocolat.* ("I love chocolate.") were labeled incorrectly.

5.6 Discussion

The process of finetuning transformer models has brought state-of-the-art results to the already very capable pretrained contextual word embeddings. However, the difficulty of interpretability of results, called colloquially the "black-box" effect, is an ongoing challenge for the NLP community. In addition, the process of finetuning is also not transparent and is considered somewhat unstable; for example Dodge et al. (2020) point out that initializing the finetuning process with a random seed can lead to substantially different results, even with the same hyperparameters.

Our finetuned models were quite successful in the classification tasks, outperforming our baselines to a statistically significant degree. The CNN classifier without any embedding information was also able to achieve relatively high accuracy. Therefore, classifying lexical aspect (especially duration) must have been easy for the deep contextualized word embedding models. However, we did observe how impactful the datasets were for the transfer learning process. Both datasets contained errors, and even though we were able to eliminate most of the problems, there might have been several cases of mislabeling left – not necessarily a problem, since introducing some noisy data is a known NLP strategy. The first dataset of Friedrich and Gateva contained longer and more complex sentences than the Caption dataset Alikhani and Stone (2019). This may explain why models had trouble with some long sentences in the qualitative test sets, having seen shorter utterances with a more uncomplicated structure.

Additionally, for the duration classification, the superior performance with finetuned models did raise a question; did the models learn to classify duration or to identify the different corpora? With our qualitative analysis in two languages, we can conclude that the models are indeed able to classify duration and were successful because of the little overlap between stative and durative verbs and contexts. However, the models struggled with sentences for which world knowledge is crucial, which is a known issue of contextualized word embeddings (Rogers et al., 2021).

Another interesting finding is that the large models sometimes outperformed base models, even though they are more unpredictable in finetuning, as documented in literature (Dodge et al., 2020). Perhaps for a complex task such as lexical aspect identification,

the additional processing and information available to the large models were beneficial to their classification accuracy. However, during our experiments, we also noticed that sometimes the finetuning process for large models was a failure and they failed to classify, thus the process had to be repeated.

Additionally, the models proved to be quite successful in the classification tasks even without finetuning, simply with the information included in the single-verb embedding. Thus, contextual embeddings prove to efficiently encode the verb's interaction with its context, which is relied on for the verb's lexical aspect (for example, a telic verb such as "eat" would be found more frequently with count nouns that establish the endpoint of the action). This contextual knowledge is already learned in the pretraining process, and the finetuning process supplements information for higher accuracy.

Surprisingly, the classification results were not greatly harmed by the segmentation of verbs and context into subwords by the models' tokenizers, for example, the ALBERT tokenizer separating nouns from their plural suffix. This could have been problematic since the presence of plural tense sometimes affects the telicity of a sentence (Krifka, 1998). However, the model might need to focus on more tokens and may not favor some parts of the context, if additional segmentation separates verb characteristics from the root. Therefore, the models with smaller vocabularies such as ALBERT might have slightly underperformed because of this.

Examining the models' self-attention mechanisms allowed us to gain some, but very limited, insight into how input sequences were treated by the models. BERT's self-attention mechanism on earlier layers demonstrated a certain sensitivity to syntactic structure and better "focus" on individual tokens in early layers. However, the other models did not show a specific focus on constituents in any layer or attention head. This could have led to their weaker performance in the quantitative test set, compared to BERT models, especially from RoBERTa and ALBERT which are optimized versions of BERT and had slightly lower performance than BERT models. XLNet models, despite the architecture's reported improved performance on longer dependencies in other NLP tasks, were not able to attend to context more efficiently than BERT or encode more pertinent information in their encodings.

The use of the verb position was very beneficial for the first round of experiments,

and less for the second. This could be due to two reasons: first, the Friedrich dataset was annotated with the verb position in mind, while the Captions dataset wasn't annotated. We used a dependency parsing tool to extract the verb from the Captions sentences, which could have led to parsing mistakes. However, in our opinion, the likelihood of widespread mistakes is quite low, given that the sentences were quite small and contained one verb. The influence of finetuning with verb position information became more evident in the qualitative sets, where BERT models made fewer annotation mistakes when finetuned with this additional information.

Our examination of varying verb tenses and positions of prepositional phrases revealed that models showed some preference for the past perfective and continuous tenses, compared to the past simple tense. Word order was not a strong indicator of success or confusion, but placing a prepositional phrase of time at the start of the sentence (as opposed to the middle or end) occasionally enhanced predictions. This is to be expected because the bidirectional transformer architectures should not be very sensitive to word order, and also, as the self-attention visualization demonstrated, the prepositional phrases were not heavily attended to. Sentences with conflicting contexts were rarely classified correctly. This leads us to conclude that the verb embedding and its information is more important to the model's classification effort than the other word emebeddings.

Finally, our results on the French datasets demonstrated that the syntactic and semantic choices a language makes in conveying aspect did influence the models' capacity to categorize aspect, even with our lower-performing models. Even with different model architectures, the disparities in classification mistakes and successes in the qualitative datasets of the two languages show that the morphosyntax of French may lead to different semantic representations by the model. This is supported by the fact that errors occurred in the classification of telicity in English sentences with the presence of distinct simple/continuous tenses (which are not connected to telicity but were beneficial to the model in some cases), while their French translations were correctly classified regardless of the verb form.

5.7 Appendix

5.7.1 English datasets

label	sentence
telic	I **ate** a fish for lunch.
telic	John **built** a house in a year.
telic	The cat **drank** all the milk.
telic	I **spilled** the milk.
telic	Yesterday I **ran** a mile in under 10 minutes.
telic	The inspector **checked** our tickets after the first stop.
telic	The classes **lasted** one hour and took place twice a week over a four-week period.
telic	I **hang** the picture on the wall.
telic	The vase **broke** in a million pieces.
telic	John **kicked** the door shut.
telic	I **opened** the juice bottle.
telic	She **opens** the door and the dog jumps in her lap.
telic	Kim has **written** a song.
telic	You **fell** for my trap again.
telic	The advancements in technology have **changed** the world.
telic	Louise **made** the biggest progress of everyone this year.
telic	The dog **destroyed** the couch.
telic	She **cut** one single rose from the bush.
telic	The soup **cooled** in an hour.
telic	Jean was **born** in 1993 in Lyon.
atelic	I **eat** a fish for lunch on Fridays.
atelic	John is **building** good houses with his construction company.
atelic	John **watched** TV.
atelic	I always **spill** milk when I pour it in my mug.
atelic	I'm **running** 10 miles every day for my training process.
atelic	The inspectors are always **checking** every document very carefully.
atelic	The damage may **last** for many years.
atelic	We **swim** in the lake in the afternoons.
atelic	In the summer months James **sleeps** in every morning.
atelic	Cork **floats** on water.
atelic	My grandfather still **lives** in his childhood home.
atelic	Nobody **laughs** at my corny jokes.
atelic	Jenny **worked** as a doctor her whole life.
atelic	I am **working** on a big project now.
atelic	Kim is **singing**.
atelic	Kim is **writing** a song.
atelic	Grandma is **making** pancakes for breakfast.
atelic	He is constantly **changing** his script.
atelic	We **live** in a democratic age.
atelic	The Earth **revolves** around the Sun.

Table 5.27: Annotated sentences for telicity.

label	sentence
stative	She didn't **agree** with us.
stative	I don't **believe** the news.
stative	Bread **consists** of flour, water and yeast.
stative	This box **contains** a cake.
stative	I **disagree** with you.
stative	I have **disliked** mushrooms for years.
stative	This shirt **fits** me well.
stative	Julie's always **hated** dogs.
stative	Do you **hear** music?
stative	This cookbook **includes** a recipe for bread.
stative	I've **known** Julie for ten years.
stative	I **like** reading detective stories.
stative	I **love** chocolate.
stative	I **prefer** chocolate ice cream.
stative	I didn't **realise** the problem.
stative	I didn't **recognise** my old friend.
stative	He didn't **remember** my name.
stative	Your idea **sounds** great.
stative	I **suppose** John will be late.
stative	The noise **surprised** me.
durative	She **plays** tennis every Friday.
durative	She's **playing** tennis right now.
durative	The snow **melts** every spring.
durative	The snow is **melting** right now.
durative	The boxer **hits** his opponent.
durative	The boxer is **hitting** his opponent.
durative	They **ate** their dinner in silence.
durative	I **walked** past the barn.
durative	We **learned** to make pasta.
durative	He **grew** potatoes in his farm.
durative	I **slept** all morning.
durative	We **talked** for hours on our trips.
durative	I will **write** you a letter tomorrow.
durative	She **runs** ten kilometers a day.
durative	He **read** a fairytale to his kids.
durative	The boy **kicked** the ball hard.
durative	We will **go** soon.
durative	He **screamed** for help.
durative	The dogs **bark** all night.
durative	She **closed** the door.

Table 5.28: Annotated sentences for duration.

label	sentence
telic	The girl **walked** a kilometer yesterday.
atelic	The girl **walked** yesterday.
telic	I will **receive** new stock on Friday.
atelic	I will **receive** new stock on Fridays.
telic	The boy is **eating** an apple.
atelic	The boy is **eating** apples.
telic	I **drank** the whole bottle.
atelic	I **drank** juice.
telic	I **read** the book in an hour.
atelic	I **read** the book for an hour.
telic	The Prime Minister **made** that declaration yesterday.
atelic	The Prime Minister **made** that declaration for months.
telic	The workers **painted** the house in an hour.
atelic	The workers **painted** the house for an hour.
telic	The hunters **chased** the deer away.
atelic	The hunters **chased** the deer.
telic	I **finished** reading the book at 5 pm.
atelic	I **stopped** reading the book at 5 pm.
telic	The pond is **freezing** over.
atelic	It's **freezing** outside.
telic	The hunter **reached** the mountain hut.
atelic	The hunter **occupied** the mountain hut.
telic	I **put** on my red dress.
atelic	I **wore** my red dress.
telic	The artist **draws** a painting.
atelic	The artist **studies** a painting.
telic	The policemen **entered** the church.
atelic	The policemen **watched** the church.
telic	They **caught** the boar.
atelic	They **hunted** the boar.
telic	She **fell** asleep at 8 pm.
atelic	She **slept** at 8 pm.
telic	She **noticed** him.
atelic	She **looked** at him.
telic	The people **died** from starvation.
atelic	The people **suffered** from starvation.
telic	They **built** the house.
atelic	They have been **building** the house.
telic	She **ate** that sandwich.
atelic	She has been **eating** that sandwich.

Table 5.29: Minimal pairs of telicity.

label	Sentence
telic	I ate a fish for lunch at noon.
telic	I had eaten a fish for lunch at noon.
telic	At noon, I ate a fish for lunch.
telic	At noon, I had eaten a fish for lunch.
telic	John built a house in a year.
telic	John had built a house in a year.
telic	In a year, John built a house.
telic	In a year, John had built a house.
telic	I ran a mile in under 10 minutes yesterday.
telic	I had run a mile in under 10 minutes yesterday.
telic	I ran a mile yesterday in under 10 minutes.
telic	I had run a mile yesterday in under 10 minutes.
telic	Yesterday I ran a mile in under 10 minutes.
telic	Yesterday I had run a mile in under 10 minutes.
telic	The inspector checked our tickets after the first stop.
telic	The inspector had checked our tickets after the first stop.
telic	After the first stop, the inspector checked our tickets.
telic	After the first stop, the inspector had checked our tickets.
telic	The classes lasted one hour and took place twice a week over a four-week period.
telic	The classes lasted one hour and had taken place twice a week over a four-week period.
telic	The classes took place twice a week over a four-week period and lasted one hour.
telic	The classes had taken place twice a week over a four-week period and lasted one hour.
telic	Over a four-week period, the classes lasted one hour and took place twice a week.
telic	Over a four-week period, the classes lasted one hour and had taken place twice a week.
telic	Louise made the biggest progress out of everyone this year.
telic	Louise had made the biggest progress out of everyone this year.
telic	Out of everyone this year, Louise made the biggest progress.
telic	Out of everyone this year, Louise had made the biggest progress.
telic	This year, Louise had made the biggest progress out of everyone.
telic	This year, Louise made the biggest progress out of everyone.
telic	The soup cooled in an hour.
telic	The soup had cooled in an hour.
telic	In an hour, the soup cooled.
telic	In an hour, the soup had cooled.
telic	John Wilkes Booth killed Lincoln on 1865.
telic	On 1865, John Wilkes Booth killed Lincoln.
telic	Lincoln was killed by John Wilkes Booth on 1865.
telic	On 1865, Lincoln was killed by John Wilkes Booth.
telic	John Wilkes Booth had killed Lincoln before the play ended.
telic	Before the play ended, John Wilkes Booth had killed Lincoln.
atelic	I eat a fish for lunch on Fridays.
atelic	I usually eat a fish for lunch of Fridays.
atelic	On Fridays, I eat a fish for lunch.

Table 5.30 – continued from previous page.

label	Sentence
atelic	On Fridays, I usually eat a fish for lunch.
atelic	John watched TV.
atelic	John watched TV all afternoon.
atelic	John watched TV every afternoon.
atelic	John watched TV after finishing his homework.
atelic	I'm running 10 miles every day for my training process.
atelic	Every day I'm running 10 miles for my training process.
atelic	We swim in the lake in the afternoons.
atelic	We swim in the lake each afternoon.
atelic	In the afternoons, we swim in the lake.
atelic	Each afternoon, we swim in the lake.
atelic	Kim is singing.
atelic	Kim is singing a song.
atelic	Kim is writing.
atelic	Kim is writing a song.
atelic	In the summer months James sleeps in every morning.
atelic	James sleeps in every morning in the summer months.
atelic	Grandma is making pancakes for breakfast.
atelic	Grandma is making pancakes whenever we visit her.
atelic	For breakfast, grandma is making pancakes.
atelic	Whenever we visit her, grandma is making pancakes.
atelic	I will receive new stock on Fridays.
atelic	I receive new stock on Fridays.
atelic	On Fridays, I will receive new stock,
atelic	On Fridays, I receive new stock.
atelic	I read the book for an hour.
atelic	I have been reading the book for an hour.
atelic	The Prime Minister made that declaration for months.
atelic	The Prime Minister has been making that declaration for months.
atelic	For months the Prime Minister made that declaration.
atelic	For months the Prime Minister has been making that declaration.
atelic	The workers painted the house for an hour.
atelic	The workers have been painting the house for an hour.
atelic	The workers painted the house since 8 am.
atelic	The workers have been painting the house since 8 am.
atelic	The workers had been painting the house for an hour.
atelic	The workers had been painting the house since 8 am.

Table 5.30: Additional sentences annotated for telicity, with variations of verb tense and word order.

5.7.2 French datasets

label	sentence
telic	J'ai mangé un poisson pour le déjeuner.
telic	Jean a construit une maison dans un an.
telic	Le chat a bu tout le lait.
telic	J'ai renversé le lait.
telic	Hier, j'ai couru un kilomètre en moins de 10 minutes.
telic	L'inspecteur a vérifié nos billets après le premier arrêt.
telic	Les cours duraient une heure et avaient lieu deux fois par semaine sur une période de quatre semaines.
telic	J'ai accroché le tableau au mur.
telic	Le vase s'est brisé en mille morceaux.
telic	John a fermé la porte d'un coup de pied.
telic	J'ai ouvert la bouteille de jus de fruit.
telic	Elle ouvre la porte et le chien saute sur ses genoux.
telic	Kim a écrit une chanson.
telic	Tu es encore tombé dans mon piège.
telic	Les progrès de la technologie ont changé le monde.
telic	Louise a fait le plus gros progrès de tous cette année.
telic	Le chien a détruit le canapé.
telic	Elle a coupé une seule rose du buisson.
telic	La soupe a refroidi en une heure.
telic	Jean est né en 1993 à Lyon.
atelic	Je mange un poisson à midi le vendredi.
atelic	Jean construit de belles maisons avec son entreprise de construction.
atelic	Jean regarde la télévision.
atelic	Je renverse toujours le lait quand je le verse dans ma tasse.
atelic	Je cours 16 km tous les jours pour m'entraîner.
atelic	Les inspecteurs vérifient toujours très soigneusement chaque document.
atelic	Les dégâts peuvent durer de nombreuses années.
atelic	Nous nageons dans le lac l'après-midi.
atelic	Pendant les mois d'été, James fait la grasse matinée tous les matins.
atelic	Le liège flotte sur l'eau.
atelic	Mon grand-père vit toujours dans la maison de son enfance.
atelic	Personne ne rit de mes blagues à l'eau de rose.
atelic	Jenny a travaillé comme médecin toute sa vie.
atelic	Je travaille sur un grand projet en ce moment.
atelic	Kim chante.
atelic	Kim écrit une chanson.
atelic	Notre grand-mère fait des crêpes pour le petit-déjeuner.
atelic	Il change constamment son scénario.
atelic	Nous vivons dans une ère démocratique.
atelic	La Terre tourne autour du Soleil.

Table 5.31: French annotated sentences for telicity.

label	sentence
stative	Elle n'était pas d'**accord** avec nous.
stative	Je ne **crois** pas les nouvelles.
stative	Le pain est **composé** de farine , d'eau et de levure.
stative	Cette boîte **contient** un gâteau.
stative	Je ne suis pas d'**accord** avec vous.
stative	Je n'**aime** pas les champignons depuis des années.
stative	Cette chemise me **va** bien.
stative	Julie a toujours **détesté** les chiens.
stative	Tu **entends** de la musique ?
stative	Ce livre de cuisine **contient** une recette de pain.
stative	Je **connais** Julie depuis dix ans.
stative	J'**aime** lire des romans policiers.
stative	J'**aime** le chocolat.
stative	Je **préfère** la glace au chocolat.
stative	Je ne me suis pas **rendu** compte du problème.
stative	Je n'ai pas **reconnu** mon vieil ami.
stative	Il ne s'est pas **souvenu** de mon nom.
stative	Ton idée est **géniale**.
stative	Je **suppose** que John sera en retard.
stative	Le bruit m'a **surpris**.
durative	Elle **joue** au tennis tous les vendredis.
durative	Elle **joue** au tennis en ce moment.
durative	La neige **fond** chaque printemps.
durative	La neige **fond** en ce moment.
durative	Le boxeur **frappe** son adversaire.
durative	Le boxeur va **frappe**r son adversaire.
durative	Ils ont **mangé** leur dîner en silence.
durative	Je suis **passé** devant la grange.
durative	Nous avons **appris** à faire des pâtes.
durative	Il **cultivait** des pommes de terre dans sa ferme.
durative	J'ai **dormi** toute la matinée.
durative	Nous avons **parlé** pendant des heures de nos voyages.
durative	Je t'**écrirai** une lettre demain.
durative	Elle **court** dix kilomètres par jour.
durative	Il a **lu** un conte de fées à ses enfants.
durative	Le garçon a **frappé** le ballon avec force.
durative	Nous **allons** bientôt partir.
durative	Il a **crié** à l'aide.
durative	Les chiens **aboient** toute la nuit.
durative	Elle a **fermé** la porte.

Table 5.32: French annotated sentences for duration.

label	sentence
telic	La fille a **marché** un kilomètre hier.
atelic	La fille a **marché** hier.
telic	Je **recevrai** de nouveaux stocks ce vendredi.
atelic	Je **recevrai** de nouveaux stocks les vendredis.
telic	Le garçon **mange** une pomme.
atelic	Le **garçon** mange des pommes.
telic	J'ai bu toute la **bouteille**.
atelic	J'**ai** bu du jus de fruit.
telic	J'ai lu le **livre** en une heure.
atelic	J'ai lu le **livre** pendant une heure.
telic	Le Premier ministre a **fait** cette déclaration hier.
atelic	Le Premier ministre a **fait** cette déclaration depuis des mois.
telic	Les ouvriers ont **peint** la maison en une heure.
atelic	Les ouvriers ont **peint** la maison pendant une heure.
telic	Les chasseurs ont **chassé** le cerf.
atelic	Les chasseurs **chassaient** le cerf.
telic	J'ai **fini** de lire le livre à atelic7 heures.
atelic	J'ai **arrêté** de lire le livre à atelic7 heures.
telic	L'étang est **gelé**.
atelic	Il **gèle** dehors.
telic	Le chasseur a **atteint** le refuge de montagne.
atelic	Le chasseur a **occupé** la cabane de montagne.
telic	J'ai **mis** ma robe rouge.
atelic	Je **portais** ma robe rouge.
telic	L'artiste **dessine** un tableau.
atelic	L'artiste **étudie** un tableau.
telic	Les policiers sont **entrés** dans l'église.
atelic	Les policiers ont **surveillé** l'église.
telic	Ils ont **attrapé** le sanglier.
atelic	Ils ont **chassé** le sanglier.
telic	Elle s'est **endormie** à 20 heures.
atelic	Elle s'est **endormie**.
telic	Elle l'a **remarqué**.
atelic	Elle l'a **regardé**.
telic	Les gens sont **morts** de faim.
atelic	Les gens ont **souffert** de la famine.
telic	Ils ont **construit** la maison.
atelic	Ils sont en **train** de construire la maison.
telic	Elle a **mangé** ce sandwich.
atelic	Elle était en **train** de manger ce sandwich.

Table 5.33: Minimal pairs for telicity in French.

label	sentence
telic	La fille a **marché** un kilomètre hier.
telic	Je **recevrai** de nouveaux stocks vendredi.
telic	Le garçon **mange** une pomme.
telic	J'ai bu toute la **bouteille**.
telic	J'ai lu le **livre** en une heure.
telic	Le Premier ministre a **fait** cette déclaration hier.
telic	Les ouvriers ont **peint** la maison en une heure.
telic	Les chasseurs ont **chassé** le cerf.
telic	J'ai **fini** de lire le livre à atelic7 heures.
telic	L'étang est **gelé**.
telic	Le chasseur a **atteint** le refuge de la montagne.
telic	J'ai **mis** ma robe rouge.
telic	L'artiste **dessine** un tableau.
telic	Les policiers sont **entrés** dans l'église.
telic	Ils ont **attrapé** le sanglier.
telic	Elle s'est **endormie** à 20 heures.
telic	Elle l'a **remarqué**.
telic	Les gens sont **morts** de faim.
telic	Ils ont **construit** la maison.
telic	Elle a **mangé** un sandwich.
atelic	La fille a **marché** hier.
atelic	Je **recevrai** de nouveaux stocks tous les vendredis.
atelic	Le garçon **mange** des pommes.
atelic	J'ai **bu** du jus de fruit.
atelic	J'ai lu le **livre** pendant une heure.
atelic	Le Premier ministre a **fait** cette déclaration pendant des mois.
atelic	Les ouvriers ont **peint** la maison pendant une heure.
atelic	Les chasseurs ont **poursuivi** le cerf.
atelic	J'ai **arrêté** de lire le livre à atelic7 heures.
atelic	Il **fait** froid dehors.
atelic	Le chasseur a **occupé** la cabane de montagne.
atelic	Je **portais** ma robe rouge.
atelic	L'artiste **étudie** un tableau.
atelic	Les policiers **surveillent** l'église.
atelic	Ils ont **chassé** le sanglier.
atelic	Elle a **dormi** à 2telic heures.
atelic	Elle l'a **regardé**.
atelic	Les gens ont **souffert** de la famine.
atelic	Ils étaient en **train** de construire la maison.
atelic	Elle était en **train** de manger un sandwich.

Table 5.34: Additional sentences for telicity.

Chapter 6

Classification of attributive adjective position in French

6.1 Introduction

Our topic of research explores the competencies of deep contextual word embeddings with word order when it is relevant to the grammaticality and meaning of a sequence. While previous work has shown that transformer models are insensitive to word order (Pham et al., 2021; Gupta et al., 2021), finetuned models have been successful in classifying permutated word order (Sinha et al., 2021b). The linguistic phenomenon related to word order that we studied in this chapter is adjective placement in French. Despite the traditional grammar rules that suggest postposition (Laurent and Delaunay, 2013, Paragraph 31), the placement of the attributive adjective in a noun phrase, with regard to its head noun, can vary significantly, based on syntactic and semantic processes. The position of the attributive adjective can be crucial to the meaning of the noun phrase. While linguistic intuition is sufficient for native speakers to make these decisions, our goal is to assess whether transformer models are capable of understanding the difference between the two possible positions of an adjective in a sequence.

We finetuned French transformer models to learn the preferred attributive adjective position in noun phrases, by providing the two possible positions and classifying for the preferred one, since the models do not have information on the proper syntactic structure of a noun phrase. We also tested with uninformed and traditional baselines and we also examined the effect of attention masks on classification (blocking attention to the noun phrase or to the rest of the context). We studied the pretrained word embed-

dings with masked predictions and with traditional visualization methods. Finally, we also had the opportunity to conduct an experiment with native French speakers, to compare the models' predictions to their choices in challenging cases of attributive adjective placement.

6.2 Linguistic background

Traditional prescriptivist grammar states that attributive adjectives in French should come after the noun in a noun phrase—however, linguistic analysis suggests that, in general, adjectives are mobile and can come before or after the noun (Abeillé and Godard, 1999). Linguistic studies were conducted based on frequency in large corpora (Benzitoun, 2014; Thuilier, 2013), in an effort to capture the preferred adjective position by native speakers rather than follow grammatical rules. Most attributive adjectives have a preference or tendency to appear in a specific position in a given context; for example, chromatic adjectives have almost exclusively been found in postposition. Nonetheless, polysemic adjectives may prefer different positions for their different meanings, for example, the adjective *cher* can be anteposed when it means "dear" (*ma chère maison* "my dear house") and postposed when it means "expensive" (*une maison chère* "an expensive house") (Thuilier, 2012). Benzitoun (2013) divided adjectives into three groups:

- Adjectives that only accept anteposition, for example, ordinal adjectives with the suffix *-ième* (e.g. *troisième* "third"), are almost always anteposed to the noun. The adjectives *satané* "damned", *triple* "triple" and *tiers* "third-party" are also only found in anteposition (Benzitoun, 2013).

- Adjectives that only accept postposition, for example, the adjectives *exotique* "exotic", *idéal* "ideal", *populaire* "popular", *moderne* "modern", *géant* "giant", *naturel* "natural" are always postposed according to Larsson (1994).

- Adjectives that accept either position, for example, *énorme* "huge", *immense* "immense", *superbe* "superb" alternate between the two possible positions (Larsson, 1994; Benzitoun, 2014).

However, classification based solely on frequency has proven to be inconsistent. For

example, Benzitoun (2014) reported that *puissant* "powerful" and *difficile* "difficult" are only found in postposition, but in later worked reported cases of anteposition for both nouns. Therefore, linguists have defined constraints, based on non-linguistic and linguistic features of adjectives and adjective classes, in order to group the behavior patterns of adjectives and adjective classes (Thuilier, 2012).

The preferred position of an attributive adjective depends on its **frequency**; for instance, according to corpus data, the adjective *prochain* "next" does not appear in postposition in plural form, although it does in the singular form (Benzitoun, 2014). According to calculations by Wilmet (1980, 1981) and by Forsgren (2016), the most frequent adjectives in written-word corpora almost always appear before the noun: *grand* "big", *petit* "small", *bon* "good", *jeune* "young", *beau* "pretty", *vieux* "old".

However, the high-frequency chromatic adjectives, such as *rouge* "red", are always postposed to nouns (except in multi-word expressions). In this case, the adjective **features and class** determine the position, despite frequency. Wilmet (1980, 1981) and Forsgren (2016) state that chromatic and nationality adjectives tend to prefer postposition, while ordinal adjectives prefer anteposition. Additionally, derivative adjectives also show a significant propensity for postposition of (Forsgren, 2016; Goes, 1999). Short adjectives have a tendency to antepose, but longer adjectives can only be postposed, according to Wilmet (1981) and Forsgren (1978), because of phonetic constraints (Abeillé and Godard, 1999).

The placement of an adjective with regard to its head word can also be influenced by **semantics**, as previously exhibited with the adjective *cher*. In multi-word expressions (also called fixed expressions in literature), adjectives appear in a specific position in the noun phrase. For some noun and adjective pairs, the specific combination can only be found in a fixed expression, hence the position of the adjective in the particular phrase is always determined, e.g. *chaise longue* "lounge chair" (Gross, 1996, Chapter 2). Additionally, Benzitoun (2014) introduces the idea of *word pairs*, noun-adjective pairs that can exist in a lexicalized fixed expression with a specific meaning, but that can also be combined in other phrases with different meanings, thus allowing for a different adjective position. For example, the fixed expression *arts premiers*, where *premier* is postposed, has a specific meaning ("arts of the non-Western world") compared to *premiers arts* "first

arts" where it used in its literal sense (and is not lexicalized).

The presence of other **dependents** in the noun phrase also influences the position of the attributive adjective. The general tendency in French is to place short elements before the longer ones (Thuilier, 2013; Forsgren, 2016). This tendency may be sometimes overruled by high-frequency adjectives, such as *magnifique* "magnificent". If there is a dependent to the adjective in the adjective phrase (another adjective, adverb, or phrase), then the adjective phrase will be postposed to the noun, in the noun phrase, e.g. the anteposed adjective *fier* "proud" will be postposed with the post-adjectival dependent *de son fils* "of his son" in the phrase *un homme fier de son fils* "a man proud of his son" in order to not be separated from the noun (Thuilier, 2013). When a noun has multiple adjective dependents, postposition is also preferred. For example, the anteposed adjective *grand* "large" will be postposed in the phrase *un appartement grand et calme* "a large and quiet apartment". Overall, the presence of additional dependents may drive the adjective phrase into postposition, raise the likelihood that it will be in postposition, or at the very least allow for more flexible placement of the adjective phrase concerning the noun.

6.3 Word order and Transformer models

By design, Transformer-based architectures learn in a parallel, non-sequential way; this has raised the question of whether this is directly reflected in the way language is learned. Even though most architectures also encode positions with relative positional embeddings—which are beneficial to their performance (Yang et al., 2019a)—it is questionable how important this information is during downstream tasks. For human languages, however, word order is controlled by the syntactic rules of a language (allowing for strict or free-er word order in sentences but with reasonable limitations) and is crucial for the grammaticality and acceptability of a sentence.

Transformer models trained with masked language modeling, such as BERT and RoBERTa, are able to learn absolute word positions, but they also learn structural word positions (i.e. phrase position in hierarchical tree structures) and make use of them (Wang et al., 2019b; Wang and Chen, 2020). Multiple experiments combine absolute and structural word positions to create better-informed and better-performing word embeddings (Wang et al., 2019a; He et al., 2020; Chang et al., 2021; Wang et al., 2020a).

A lot of experiments have been performed on Transformer-based architectures, by retraining the models with permutated word orders or testing the pretrained models' sensitivities on shuffled word order. Pham et al. (2021) conducted experiments on BERT-based models (BERT, RoBERTa, ALBERT) with the GLUE benchmark classification tasks, and showed that downstream tasks were not affected by shuffled word order, except for the grammatical correctness task. Hessel and Schofield (2021) train BERT and RoBERTa models with short-distance permutations (i.e. shuffled order that respects the self-attention distance) and notice no decline in performance in several GLUE tasks.

However, even though pretrained models do not always make use of word order information, perturbation can be catastrophic on performance. O'Connor and Andreas (2021) conducted experiments on the effect that context variation has on transformer models' usable information, and discovered that word shuffling has a negative effect, whether the shuffling was implemented on short or long distances among words. Gupta et al. (2021) conducted similar experiments with GLUE tasks and observed that model performance was lower on shuffled word orders (in methods that render a sequence ungrammatical and incomprehensible to humans) but close enough to support that models rely more on embedding information rather than sequential context.

Sinha et al. (2021b) confirm that pretrained language models are insensitive to word order, in tasks of natural language inference, but show that on some occasions classification is successful only with certain (random) permutations of the input. They also conducted experiments with a RoBERTa model which has been pretrained with a shuffled corpus model and finetuned with a non-shuffled dataset and noted its positive influence on learning word order. Finetuning improved performance on tasks of inference and grammaticality (even with models pretrained with scrambled word order) (Sinha et al., 2021a). Papadimitriou et al. (2022) observe that, in BERT and GPT-2, early-layer embeddings are mostly lexical in nature, but word order plays a significant role in creating the later-layer representations of words. They also highlight that positional information may seem redundant at first, as some positional information based on syntax is already included in word embeddings, but in cases where co-occurrence can be misleading, the positional information is utilized.

Abdou et al. (2022) study positional embeddings learned from shuffled text, and probe

language models for word order information, demonstrating that even shuffled models maintain information about grammatical word order. This is partially due to the perturbations happening before tokenization, thus preserving some meaningful word orders. They also criticize previous research on permutated word order that relies solely on GLUE benchmark results, since a range of language tasks actually necessitate knowledge of word order, frequently to a degree that cannot be taught by fine-tuning.

On the limited examples of studying word order in languages other than English, Li et al. (2021) examine the French Transformer-based architectures for their capacity to capture long-range object-verb agreement and word order. They observed that models performed worse with scrambled inputs, and proportionately worse with an increasing number of permutations. They take note of the models' preferences for more singular forms, which are more frequent than plural forms. However, they do support that the models capture important information on hierarchical grammatical structure with abstract representations.

6.4 Experiment 1: Classification of adjective position via finetuning

6.4.1 Methodology

There has been a great deal of analysis of the syntactic and semantic capacities of transformer models and their pretrained word embeddings. While the models are capable of capturing important linguistic information, they are not always sensitive to word order. Given the bibliographic research in Section 6.3 and our previous experiments, here we explore whether pretrained deep contextual word embeddings (with finetuning) are able to classify the position of the adjective in a sentence.

A pair of sentences are given as input; the first sentence always has the adjective anteposed to the noun, while the second sentence always has the adjective postposed. This provides the model with the two alternative positions the adjective may take in the noun phrase. Based on the adjective order in the original sentence, the two-sentence sequence is labeled as "anteposed" (0) if the adjective was originally anteposed and "postposed" (1) if the adjective was postposed. For an illustration, see Table 6.1. The special end-of-

On construit les éléments de plus **haut niveau**. ⇓	
Sentence	**Label**
On construit les éléments de plus haut niveau. </s>On construit les éléments de plus niveau haut.	0

Table 6.1: An example input of two sentences for the original sentence *On construit les éléments de plus haut niveau* "We build the higher level elements" with anteposition (0). We only shift the position of the adjective-noun pair in the noun phrase, without affecting any other elements of the phrase (e.g. the dependent adjective *plus*).

sequence token of model tokenizers is used to split the sentences. With this task, the model is required to distinguish between two sentences with the same tokens but different word orders. The other word order may be implausible for some adjectives and possible for others. The acceptability will be influenced by the specific context of the sentence.

The same finetuning experiment is also conducted with a single sentence input that is the original sentence, with no permutations, and the adjective position as the sentence label. In this setup, the model can only predict the correctness of classification but is unaware of the different possible positions of the adjective in the noun phrase.

In order to further investigate the effect of self-attention on different tokens in the input sequence, we also performed finetuning with blocked attention to particular tokens. To do so, we used the attention mask vector (see Section 2.5.4), as seen in our previous experiments. In addition to the *standard* setting where all tokens are attended to, there were additionally the *pair* setting (in which all tokens are masked except for the adjective and its head noun) and the *context* setting (in which the adjective and noun are masked and all the other tokens are visible). The objective of the former is to determine if the adjective-noun pair embeddings are capable of capturing their preferred positions. The goal of the latter is to see whether the context already provides enough information about preferred positions even when no explicit information about the pair is provided. In this case, the prediction is based on the preferred class of nouns and adjectives in the given context. We demonstrate the attention mask setting with an input sentence in Table 6.2.

Mask Type	Tokens								
No mask	on	construit	les	éléments	de	plus	**haut**	**niveau**	.
Standard	on	construit	les	éléments	de	plus	haut	niveau	.
Pair	-	-	-	-	-	-	haut	niveau	-
Context	on	construit	les	éléments	de	plus	-	-	.

Table 6.2: Use of attention masks for the sentence: *On construit les éléments de plus haut niveau.* In this sentence, the adjective-noun pair is *haut niveau* (the adjective is before the noun). The label for all three inputs is [0]. For the double-sentence input, the same process will be followed for the second sentence of the input *On construit les éléments de plus niveau haut.*

6.4.2 Datasets

We extracted sentences with adjective-noun pairs from two syntactically parsed corpora: the frWaC corpus (Baroni et al., 2009) and the French corpora of Universal Dependencies 2.9 (UD; Zeman et al., 2021)[1].

We extracted 120K sentences from frWaC sentences and added the entirety of the Universal Dependencies (UD) corpora, and from those, we kept the sentences with adjectives as modifiers to a noun. We used a 2/3 ratio of anteposition/postposition, which is roughly the ratio documented in the literature and the one that occurs in our corpora –as measured in one million frWaC sentences and the entire UD corpora. This ratio is beneficial since anteposed adjectives are fewer but more frequent than postposed adjectives. However, we excluded the adjectives and words that were incorrectly parsed as adjectives, such as numerals and some adjectives such as *autre* "other", *certain* "certain", *chacun* "each", *quelque* "some" that are also used as pronouns. We additionally had to exclude the adjectives and nouns that the transformer model tokenizers tokenized into subwords, in order to construct the attention mask, extract probabilities and perform the masked word experiment.

The sentences of the two datasets were combined and used in various ways. Throughout our finetuning experiments, we made different iterations of the datasets to train and test. In one setting, we trained the model only with frWaC, and used the UD sentences as

[1]The list of corpora can be found at https://universaldependencies.org/fr/

an additional test set. In another one, we added a subset of the UD sentences to the train set and tested on the rest of UD; we also finetuned the model just with the (significantly) smaller UD dataset. When applicable, we tested both with frWaC and UD sentences. The size of the datasets is presented in Table 6.3.

Dataset	Train	Validation	frWac test	UD test (entire)	UD test (part)
frWaC	76,164	7,672	7,740	27,373	5,151
frWaC+UD	91,615	7,672	7,740	-	5,151
UD	13,905	1,546	7,740	-	5,151

Table 6.3: Dataset sizes for word order classification.

6.4.3 Models and baselines

We used two monolingual French transformer-based models, available from the HuggingFace Python library (Wolf et al., 2020), CamemBERT (Martin et al., 2020) and FlauBERT (Le et al., 2020). Details on the models can be found in Section 2.5.8.

The most straightforward baseline that can be created is based on frequency in corpora. From the training sets, we extracted the most frequent position for each adjective (after lemmatization) and we assigned the label of ante-/postposition according to frequency. The other baselines were built with classical NLP classification methods and without pretrained embeddings, namely a Bag-of-Words logistic regression model, implemented with scikit-learn (Pedregosa et al., 2011), and a CNN-based classifier, more sensitive to word order, implemented with PyTorch (Paszke et al., 2019). The CNN was trained for 50 epochs, without pretrained word embeddings, embedding dimension of 300, filter size of $[3, 4, 5]$, 100 filters per dimension, a dropout rate of 0.5, a learning rate of 0.01 and the Adadelta optimizer.

6.4.4 Quantitative Results

The results for the two-sentence input experiment can be found in Table 6.5 (and for the one-input in Table 6.8). With a sufficiently large training set, the CNN classifier performs quite well. It should therefore come as no surprise that the finetuned transformer models performed even better and have a very low error rate, with nearly 100% accuracy

overall. Finetuning with the combination of the two datasets (frWaC and UD) was more successful than using only the frWaC or UD dataset, the results were consistently good regardless of the test set domain. However, performance deteriorated –but was still very high for some models– with the significantly smaller UD training set as expected; finetuning guidelines recommend a training set of at least 100K inputs (Clark et al., 2019). Concerning the baselines, even though the CNN classifier performed well, the logistic regression classifier was very weak, and sometimes failed to classify (by predicting only one label). The frequency-based baseline, which does not use any NLP learning methods, was able to correctly predict the position of the adjective with great success. Especially in the case of the one-sentence training, it outperformed the transformer models and the other baselines in many cases.

The probabilities of predicted labels (see Figure 6.1) show that the models are highly confident in their predictions, both correct and incorrect. However, there were some instances of lower probabilities in incorrect predictions, for example in Table 6.4. In this case, the marginally wrong prediction occurs with the mobile adjective *ancienne* "former/old", thus both adjective positions are acceptable.

The results of the experiments with attention masks are presented in Table 6.6 and 6.7 (and Table 6.9 and 6.10 for the one-input finetuning experiment). In these experiments, the models' attention mechanism had only access to certain tokens, in order to study which contextual word embeddings carry the most information about adjective position. When attention was only allowed to the adjective and noun pair, the Flaubert models were unable to classify, while the Camembert models were perturbed but still (mostly) successful with classification, especially on the frWaC sets. Meanwhile, masking the adjective and noun pair, only allowing attention to the rest of the sequence, was surprisingly successful for the finetuned models with the larger training sets. Specifically, camembert-base and the Flaubert models reached similar accuracies to those of the no-mask finetuned models, except with the small UD training set. For the one-input finetuning experiment, we observe that performance significantly improved for the masked context scenario only for CamemBERT models and only in the frWaC set, while the Flaubert models once more failed. The performance is noticeably worse for the UD domain and the adjective-noun masked scenario.

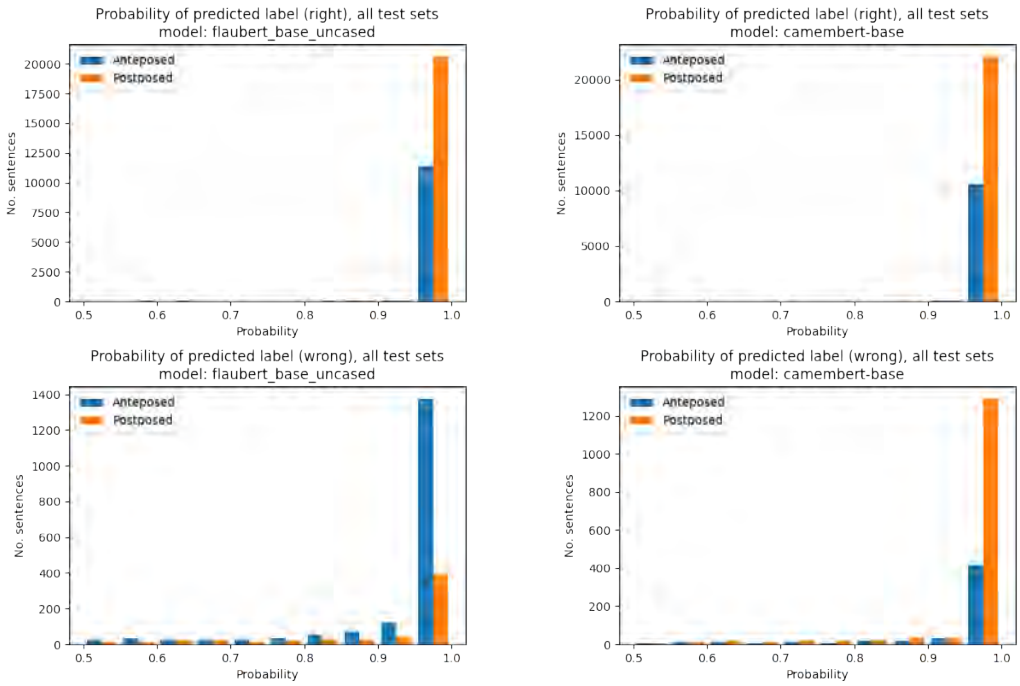


Figure 6.1: The probability of predicted labels, for correct and wrong predictions. The models were trained on the frWaC train set and the plots aggregate results from all test sets.

Input	une école a ouvert dans une **ancienne église** en 1950 . <sep> une école a ouvert dans une **église ancienne** en 1950 .
Label	Anteposition (0)
Probabilities	0: 0.4885, 1: 0.5115

Table 6.4: A (marginally) wrong prediction from the flaubert_base_uncased model. The last row displays the probabilities assigned to each label.

Model	frWaC train			frWaC+UD train		UD train	
	frWaC	**UD-full**	**UD-test**	**frWaC**	**UD-test**	**frWaC**	**UD-test**
camembert-base	**0.99**	**0.93**	0.93	**0.99**	**0.99**	0.93	0.95
camembert-large	**0.99**	0.91	0.93	**0.99**	**0.99**	**0.98**	*0.66*
flaubert-small-cased	**0.99**	0.90	0.9	**0.99**	**0.99**	0.62	*0.66*
flaubert-base-cased	**0.99**	0.90	0.87	**0.99**	0.97	0.96	**0.96**
flaubert-base-uncased	**0.99**	0.90	0.91	**0.99**	**0.99**	0.95	0.95
flaubert-large-cased	**0.99**	**0.93**	0.88	**0.99**	**0.99**	0.91	0.87
Position frequency	0.91	0.77	0.93	0.91	0.94	0.45	0.62
Logistic Regression	*0.45*	*0.68*	*0.66*	*0.45*	0.65	0.82	0.87
CNN	0.94	0.48	**0.94**	0.96	0.95	0.55	0.72

Table 6.5: Classification results for the finetuned models and baselines, with two sentence-input, for the different training and test sets. Values in *italics* indicate that the model completely failed to classify.

Model	frWaC train			frWaC+UD train		UD train	
	frWaC	**UD-full**	**UD-test**	**frWaC**	**UD-test**	**frWaC**	**UD-test**
camembert-base	**0.99**	**0.80**	**0.83**	**0.99**	**0.99**	0.78	0.83
camembert-large	0.98	0.76	0.76	*0.45*	*0.66*	**0.87**	**0.91**
flaubert-small-cased	*0.45*	*0.68*	*0.68*	*0.45*	*0.66*	*0.45*	*0.66*
flaubert-base-cased	*0.45*	*0.68*	*0.68*	*0.45*	*0.66*	*0.45*	*0.66*
flaubert-base-uncased	*0.45*	*0.68*	*0.68*	*0.45*	*0.66*	*0.45*	*0.66*
flaubert-large-cased	*0.45*	*0.68*	*0.68*	*0.45*	*0.66*	*0.45*	*0.66*

Table 6.6: Classification results of the finetuned models with two-sentence input and with attention mask: only adjective and noun visible, context is hidden. Values in *italics* indicate that the model failed completely to classify.

Model	frWaC train			frWaC+UD train		UD train	
	frWaC	**UD-full**	**UD-test**	**frWaC**	**UD-test**	**frWaC**	**UD-test**
camembert-base	**0.99**	0.45	**0.57**	**0.99**	0.98	0.45	0.66
camembert-large	*0.45*	*0.66*	*0.66*	*0.45*	*0.66*	0.45	0.63
flaubert-small-cased	0.99	**0.52**	0.52	0.99	0.98	0.47	0.64
flaubert-base-cased	**0.99**	0.47	0.47	**0.99**	**0.99**	**0.58**	**0.68**
flaubert-base-uncased	**0.99**	0.61	0.61	**0.99**	**0.99**	0.47	0.62
flaubert-large-cased	**0.99**	0.54	0.54	**0.99**	**0.99**	0.50	0.64

Table 6.7: Classification results of the finetuned models with two-sentence input and with attention mask: adjective and noun are hidden, context is visible. Values in *italics* indicate that the model failed completely to classify.

Model	frWaC train			frWaC+UD train		UD train	
	frWaC	**UD-full**	**UD-test**	**frWaC**	**UD-test**	**frWaC**	**UD-test**
camembert-base	0.89	0.80	0.80	0.89	0.87	**0.84**	**0.87**
camembert-large	0.89	0.80	0.80	0.89	0.87	**0.84**	**0.87**
flaubert-small-cased	0.88	0.81	0.81	0.88	0.87	**0.84**	0.85
flaubert-base-cased	0.89	0.81	0.81	0.89	0.87	0.82	**0.87**
flaubert-base-uncased	0.89	**0.82**	0.82	0.88	0.87	0.82	**0.87**
flaubert-large-cased	0.89	0.81	0.81	0.89	0.87	0.83	**0.87**
Position frequency	**0.91**	0.77	**0.93**	**0.91**	**0.94**	0.45	0.62
Logistic Regression	*0.45*	*0.68*	*0.66*	*0.45*	*0.65*	*0.45*	0.65
CNN	0.80	0.75	0.82	0.8	0.84	0.68	0.79

Table 6.8: Classification results for finetuning models and baselines, with only one sentence as input, with our different training and test sets. Values in *italics* indicate that the model failed completely to classify.

Model	frWaC train			frWaC+UD train		UD train	
	frWaC	**UD-full**	**UD-test**	**frWaC**	**UD-test**	**frWaC**	**UD-test**
camembert-base	**0.99**	**0.99**	0.99	0.80	0.80	0.69	0.83
camembert-large	0.97	0.98	0.98	**0.97**	**0.98**	*0.45*	*0.66*
flaubert-small-cased	*0.45*	*0.68*	*0.68*	0.76	0.79	0.57	0.72
flaubert-base-cased	*0.45*	*0.68*	*0.68*	0.80	0.80	**0.70**	**0.87**
flaubert-base-uncased	*0.45*	*0.68*	*0.68*	0.80	0.80	*0.45*	*0.68*
flaubert-large-cased	*0.45*	*0.68*	*0.68*	0.45	0.66	0.83	**0.87**

Table 6.9: Classification results of finetuning models with only one sentence as input and with attention mask: only adjective and noun visible, context is hidden. Values in *italics* indicate that the model failed completely to classify.

Model	frWaC train			frWaC+UD train		UD train	
	frWaC	**UD-full**	**UD-test**	**frWaC**	**UD-test**	**frWaC**	**UD-test**
camembert-base	0.79	0.77	0.77	0.79	0.89	0.67	0.82
camembert-large	*0.45*	*0.66*	*0.66*	*0.45*	*0.66*	*0.45*	*0.66*
flaubert-small-cased	0.76	0.75	0.75	0.76	0.82	0.59	0.74
flaubert-base-cased	0.80	0.69	0.69	**0.80**	**0.90**	**0.70**	**0.86**
flaubert-base-uncased	0.81	0.76	0.76	0.69	0.75	**0.70**	**0.86**
flaubert-large-cased	**0.82**	**0.79**	**0.79**	0.69	0.75	0.69	0.83

Table 6.10: Classification results of finetuning models with only one sentence as input and with attention mask: adjective and noun hidden, context visible. Values in *italics* indicate that the model failed completely to classify.

6.4.5 Qualitative analysis

Overall, it is challenging to deduce common factors in classification errors, since the models often produced very few errors that are inconsistent among models, and do not share an adjective or other elements. By concentrating on the frWaC training set and using the UD dataset as test set, the majority of the sentences that were incorrectly classified had an adjective that could have been in a different position with a different meaning than the original, i.e. the utterance remains grammatical and acceptable when the adjective-noun order is reversed, but the meaning changes. For example, the sentence *Une école a ouvert dans une* ***ancienne église*** *en 1950.* "A school opened in a former church." remains correct with *ancienne* postposed to the noun, but the meaning of the adjective changes from "former" to "old". The context provided by the sentence is not sufficient to decipher the actual meaning, and native French speakers agree that both sentences are grammatical. On the other hand, mistakes in the classification of sentences such as *Les* ***créations sensuelles****, modernes et orientales se font remarquer.* "The sensual, modern and oriental creations stand out." uncover the models' shallow perception of syntactic relations –these mistakes are, however, quite rare. Finally, we notice a few badly-parsed and badly-formed sentences in the dataset, which were not enough to warrant a redesign, but were confusing to the models. For example, in the sentence *Cette* ***campagne****,* ***dure*** *et sévère, contre un adversaire très mobile et mordant.* "This campaign, tough and stern, against a very agile and abrasive opponent.", the pair *campagne – dure* was identified. A correct way to create the anteposed variation would be *[...] dure campagne, [...]*, but our script generated the pair *[...] dure, campagne [...]* with the inclusion of a comma, to respect the original distance between the nouns and adjectives. These mistakes are not widespread but were easily identifiable in the few errors that the models produced.

This experiment demonstrated that the finetuned transformer models are quite effective in classifying the preferred position between two alternative positions for an adjective. However, it is important to explore whether this ability is a product of finetuning or if the pretrained models had already acquired enough information about the adjective's preferred location in relation to its context.

6.5 Experiment 2: Existing knowledge in pretrained embeddings

6.5.1 Classification with adjective embeddings

The layers of a transformer model specialize create different dynamic word embeddings, which capture and interact with a word's context in a different way than the previous layer. Therefore, the adjective embedding might contain the syntactic, contextual, and semantic information that determine its position with regard to the noun. We extracted the word embeddings of the adjective of each sentence, per layer, and we trained a simple logistic regression model –built in the same way as in Section 6.4.3. We opted to use the base and small models, in order to limit the results to a maximum of 12 layers as opposed to the 24 layers of the large models, an amount easier to study and visualize. We used the frWaC training set and tested on the frWaC test set and on the entire UD dataset.

The results of the classification for the two test sets can be seen in Figure 6.2. The classification results for the frWaC test set are quite low—close to failure of classification—except for the flaubert_base_uncased model, which unexpectedly reached 97% accuracy on the last layer. Results for the UD test set were more unpredictable, with a few layers of camembert-base reaching a very high accuracy, but the final layer having the lowest accuracy. On the other hand, the flaubert models had a progressively better performance, but they are not as good as their finetuned counterparts nor as the baselines. From the bibliography and our previous research, it has been indicated that the early-middle layers (3-6) of the base architectures tend to specialize in syntactic structures,

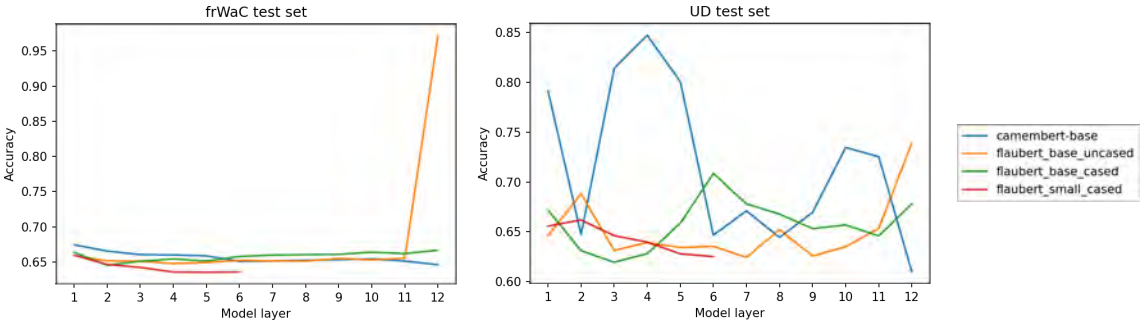


Figure 6.2: Logistic regression accuracy trained with layer-specific adjective embeddings, with the base and small models.

while late-middle layers (7-10) on semantic knowledge, and the last layers (11-12) aggregate the findings of previous layers. However, this remains a tendency and not a universal finding.

6.5.2 Adjective probabilities with Masked Language Models

One of the pretrained models' training objectives is Masked Language Modeling, i.e. the prediction of a masked token in a sequence (see Section 2.5.4). By using this objective, we retrieved the probability that the models assigned to the adjective in the sentence, specifically in the place it was originally located. Figure 6.3 presents these probabilities. It is noted that the models generally assigned higher probabilities to anteposed adjectives being in anteposition than to postposed adjectives in postposition. This could be due to multiple factors; first of all, there are stricter linguistic and feature constraints for anteposed adjectives (e.g. frequency, length, dependents). Specifically, on frequency, it is known from all previous experiments in this doctoral thesis that transformer models highly favor frequent tokens. The majority of the most frequent adjectives in French are anteposed, whereas postposition includes a much greater number of adjectives, hence the higher and lower probabilities. Additionally, we observed that CamemBERT models predict both anteposed and postposed adjectives with greater probabilities.

However, the models seemed to respect the original sentence's order of adjectives in predictions and probabilities. We shifted the [MASK] position from its original position to the opposite one and asked the models to assign the adjective's probability in the "wrong" position. The probability results can be seen in Figure 6.4. The probability of the adjectives, in the non-original position, was close to zero for at least 85% of the cases, even for anteposed adjectives which tend to be more mobile and quite frequent. Anteposed adjectives can also be found in postposition, when they are not attributive adjectives, yet the models were capable of understanding that the context of the sentence cannot have the presence of a predicative adjective in the postposed position.

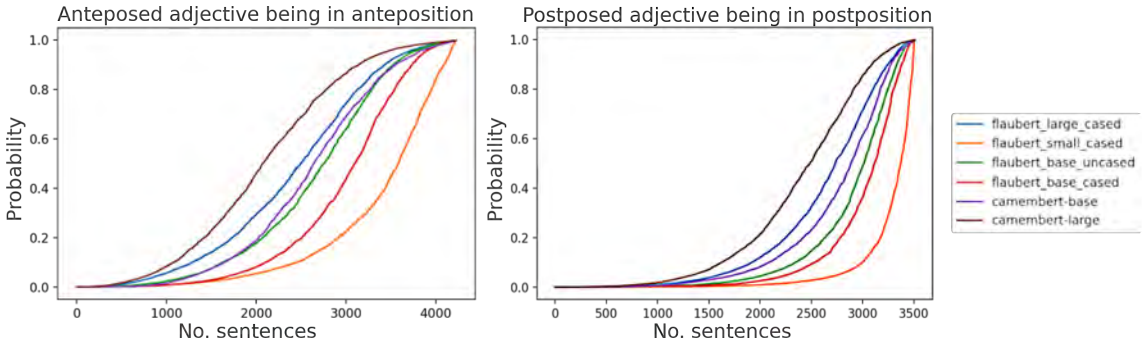


Figure 6.3: The assigned probability of each masked adjective instance, when placed in its original position, for each model.

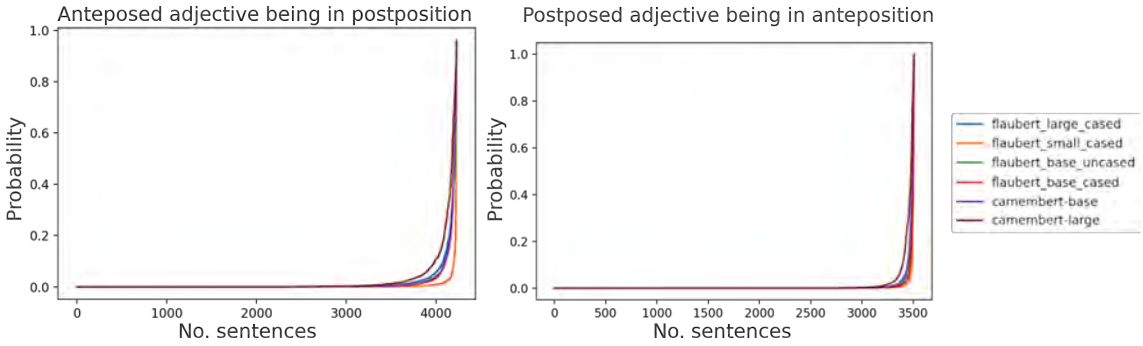


Figure 6.4: The assigned probability of each masked adjective instance, when placed in the opposite position of its original, for each model.

6.5.3 Visualizing adjective embeddings per layer

A traditional method of visualizing static word embeddings is to reduce their dimensionality and place them in a two-dimensional plot, in order to examine the algebraic relations between different vector representations of words. We attempted to create the same type of visualization with contextual word embeddings, in which case every instance of an adjective, in a specific sentence-context, for a specific layer of the transformer architecture, would correspond to one vector point. We extracted the layer-specific embeddings for some of the transformer models in our research and used them to visualize static embeddings by reducing their dimensions and plotting them on a 2-dimensional space. This allowed us to observe their nearest neighbors and examine if any clusters or patterns appear. We chose a few common adjectives from the literature (Benzitoun, 2013), either with a preferred position or mobile: *grand*, *petit* for always-anteposed, *naturel* for always-postposed, *ancien* for mobile. Each adjective's embeddings were used

and plotted separately for each layer.

We reduced the embeddings' dimensionality with t-distributed Stochastic Neighbor Embedding (t-SNE) from scikit-learn (Pedregosa et al., 2011) and plotted with matplotlib (Hunter, 2007). Some of the plots are presented in Figure 6.5. Intuitively, we assumed that the anteposed and postposed adjectives would have formed a cluster. However, we were unable to find any discernible clusters in any of the data. The closest to the formation of detectable clusters was in some early layers, for some adjectives, and not for all word forms (e.g. plurals, female).

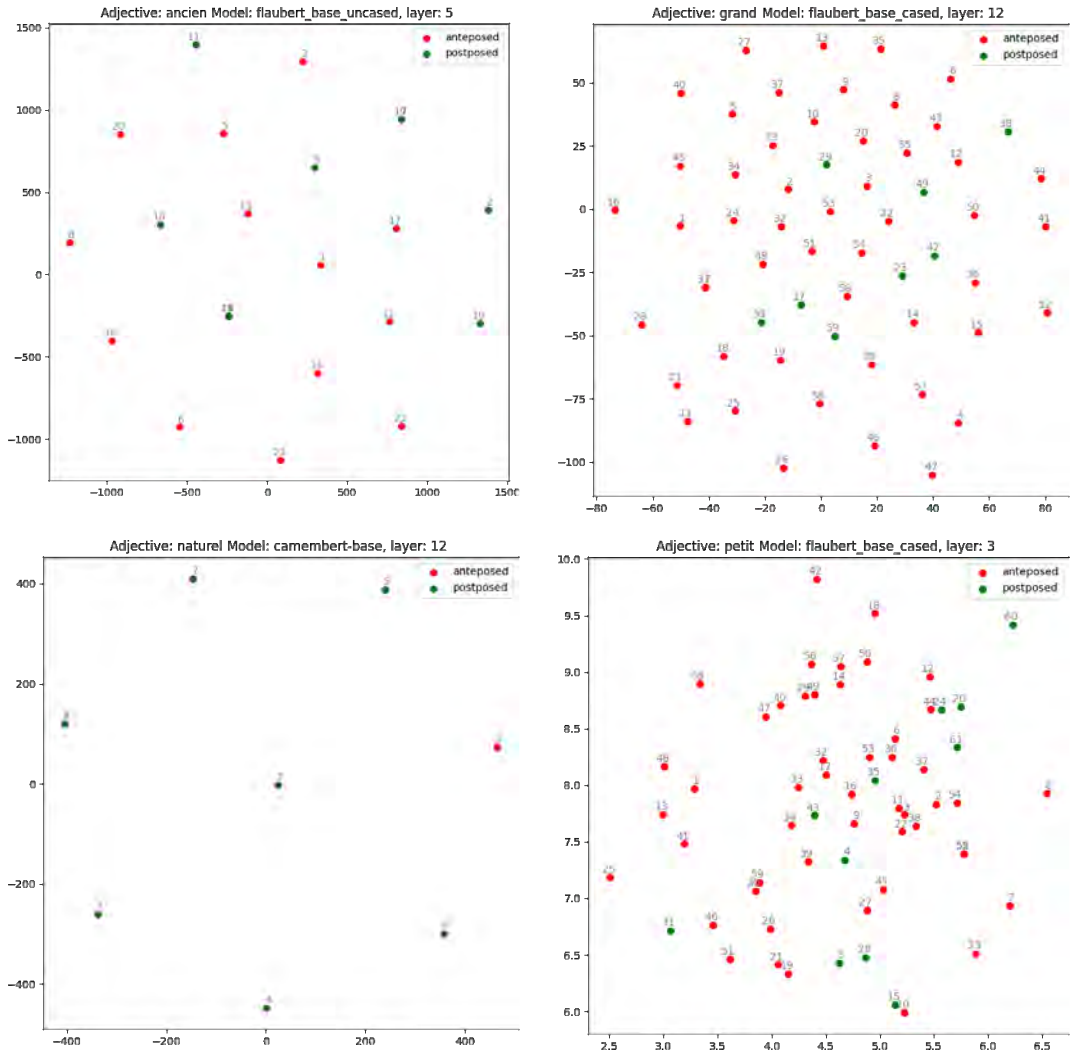


Figure 6.5: Embedding projections for base-form adjectives *ancien* 'old', *grand* "large", *naturel* "natural", *petit* "small" – from various layers and models. The numbers correspond to the sentence index.

6.6 Experiment 3: Human and Transformers judgments of adjective order

6.6.1 Methodology and Dataset

We also conducted an experiment on adjective word order with the participation of native speakers. In collaboration with Wissam Kerkri and Juliette Thuilier, we studied how native French speakers choose the preferred adjective position when presented with two variations of the sentence. The dataset of the sentences was composed of challenging cases of adjective position, caused by structural or semantic language particularities.

The experiment follows the same format as Experiment 1, in which the finetuned models and the native speakers were shown a sentence with a noun-adjective pairing and its variant in which the target adjective was in the opposite position. The models were trained on the combination of frWaC and UD datasets, which had the highest accuracy scores in Experiment 1. Both for the models and for the human participants, each pair of sentences from the two positions was given in the order of anteposition-postposition. We developed 89 prompt sentences that were either produced by a native French speaker or extracted and modified from frWaC (without any overlap with our existing training and test sets). Our goals during the creation of this dataset were to create sentences where an alternative position could either be acceptable or completely unacceptable and to limit the number of adjectives used throughout the dataset, in order to avoid great variation among participants. Based on the relationship between the adjective and the noun, or the context of the sentence, the sentences were divided into four categories. A few samples for each category are presented in Table 6.11, and the full dataset is presented in Appendix 6.8.

1. *Presence of adjective/noun dependent* (Table 6.13): The only categorical constraint that governs the position of the adjective in French is the presence of a dependent to the adjective, which forces the adjective to be postposed. However, if the dependent is to the noun, the position of the adjective is not restricted. We included sentences with the same adjectives and dependents either to the adjective or the noun.
2. *Fixed expressions* (Table 6.14): Adjectives in fixed expressions will always have a fixed position in this specific context and meaning. We contrasted the sentences with fixed

expressions with sentences whose adjectives occurred in those expressions, but not in restrictive structures.

3. *Structural persistence* (Table 6.15): Speakers are sensitive and tend to reuse repeating syntactic constructions (*syntactic priming*, Branigan et al. (1995)). The presence of a noun phrase with an adjective in a certain position may influence the processing of the next noun phrase, especially if it contains the same adjective. We want to test the extent of this effect on native speakers and our models.

4. *Blocked and mobile adjectives* (Table 6.16): In this category, we are including adjectives that are (almost) always found in postposition, and adjectives with free position depending on the meaning (*propre*, *ancien*). This category serves both as a control group, and could also provide unexpected results.

Presence of adjective/noun dependent			
Label	**Anteposition**	**Postposition**	**Translation**
anteposed	Cette **longue saison** de football a été intense.	Cette **saison longue** de football a été intense.	This long football season has been intense.
postposed	Cette **longue saison** de 4 mois a été intense.	Cette **saison longue** de 4 mois a été intense.	This 4 month long season has been intense.
Fixed Expressions			
anteposed	Il a passé une **dure semaine**.	Il a passé une **semaine dure**.	He had a tough week.
postposed	Depuis la mort de son hamster, il a une **dure vie**.	Depuis la mort de son hamster, il a une **vie dure**.	Since the death of his hamster, he has had a hard life.
Structural persistence			
anteposed	J'ai aimé le concept : bonne ambiance, **bonne musique**, les gens sont contents.	J'ai aimé le concept : bonne ambiance, **musique bonne**, les gens sont contents.	I liked the concept: good atmosphere, good music, people are happy.
postposed	Il lui a offert des volumineuses plantes à **volumineuses fleurs**.	Il lui a offert des volumineuses plantes à **fleurs volumineuses**.	He gave her bulky plants with voluminous flowers.
Blocked and mobile adjectives			
anteposed	Nous nous sommes rejoins autour d'un **chaleureux repas**.	Nous nous sommes rejoins autour d'un **repas chaleureux**.	We came together for a hearty meal.
postposed	Ce **chaleureux accueil** m'a fait chaud au cœur.	Cet **accueil chaleureux** m'a fait chaud au cœur.	This warm welcome warmed my heart.

Table 6.11: Samples of sentences of the dataset created for the questionnaire.

6.6.2 Questionnaire distribution

In order to not tire the human participants, we divided the prompt sentences into 3 questionnaires, making sure that there is an equal proportion of the four categories in each. The finetuned models, however, were given the entirety of the dataset. Out of the two position variations, the participants were asked to pick the one that seemed "more natural" to them. At the beginning of each questionnaire, we asked participants to indicate their native language. To prepare participants for the experiment, we added two sentence pairs as a tutorial that native French speakers could not possibly misinterpret. This also allowed us to eliminate (if needed) potential participants who were not fluent in French. Despite our checks, we have to rely on the truthfulness of the participants on their language skills. The survey was created using LimeSurvey[2] and disseminated to French locals and French university students. 71 participants completed the questionnaire and were not outliers. Each version of the questionnaire had 22-25 participants, i.e. each sentence pair was evaluated by at least 22 speakers.

6.6.3 Quantitative and Qualitative results

As was the case in the previous experiments, the models demonstrated weaknesses in classifying the preferred adjective position, when there were neighboring adjectives that did not belong to the noun phrase (see category *Structural persistence*). They also had issues when there was ambiguity in the text, regarding the location of the adjective due to semantics (see category *Blocked and mobile adjectives*). These occurrences cannot always be categorized as errors if the result is a grammatical sentence. These sentences had limited context and intentionally included ambiguous adjectives. Even the native speakers occasionally made judgments that were the opposite of the annotation of the original sentence, whether on purpose (a different interpretation of context) or unintentionally (possibly an interface error, haste, or lack of attention).

We calculated the average selection across all speakers, and we used this as the standard against which to evaluate our models. In order to determine which of the models' performance was most similar to the speakers' behavior, we calculated the Pearson cor-

[2]https://www.limesurvey.org/

Model	Presence of dependent	Fixed expressions	Structural persistence	Blocked and mobile	TOTAL	
					Micro avg.	Macro avg.
camembert-base	0.2097	-0.1936	-0.0756	0.4703	0.3326	0.1629
camembert-large	**0.6731**	**0.6124**	0.5292	0.516	0.5801	**0.4673**
flaubert_small_cased	0.5131	-0.0323	0.1581	**0.7802**	**0.6014**	0.3711
flaubert_base_cased	0.5168	0.0913	0.378	0.7065	0.4330	0.3446
flaubert_base_uncased	0.4012	0.2222	**0.6325**	0.5898	0.5192	0.3298
flaubert_large_cased	0.4604	0.1750	**0.6325**	0.4663	0.3688	0.3554

Table 6.12: Correlation between the average choice of the speakers and each model's output. Micro-averaged is aggregating all sentences regardless of category while macro-average is category-sensitive.

relation between the speakers' selections and the models in Table 6.12. The model with the highest micro- and macro-averaged correlation was camembert-large, although the micro-averaged correlation of flaubert_small_cased model was marginally better. However, according to its creators, since it was developed for debugging purposes and the performance of this model may be unreliable[3]. We still opted to use it, since models with a smaller number of parameters have proven to be successful in many tasks. The camembert-base and flaubert_large_cased models showed the lowest correlations, and all models except for camembert-large did not show a strong positive correlation (>0.4) in the macro-averaged correlation.

We also conducted a comparison of the decisions made by the speakers and the predictions made by the models for each category. For the *Presence of adjective/noun dependent* category, even when the dependent phrase was attached to the noun, the speakers preferred longer adjectives in postposition. For instance, all of the speakers chose the postposed version of the sentence *Ils vivent une différente relation sans amour.* "They lived a different relationship without love." and so did most of the models. However, for shorter adjectives, the speakers chose anteposition when there was a noun dependent and postposition when there was an adjective dependent. The models did not behave consistently; some models tended to favor postposition (camembert-large) or anteposition (flaubert-large-cased), whereas the more successful ones made errors on the shorter adjectives.

In the *fixed expressions* category, the speakers naturally were able to differentiate

[3]Source: https://huggingface.co/flaubert/flaubert_base_cased

between the fixed and the free position of the same adjective in different contexts. The models, however, made a number of errors on widely used fixed expressions, e.g. *la grasse matinée* "the morning of sleeping in", but were not mistaken on expressions with a short adjective, e.g. *bénéfice net* "net benefit" (i.e. the short adjective was not anteposed, while its variations in non-fixed phrases are commonly anteposed).

In the category of *structural persistence*, the speakers were able to make their choices for the adjective position despite being primed by a preceding noun phrase with the opposite adjective position, e.g. they preferred the variation *Il lui a offert des volumineuses plantes à fleurs volumineuses.* "He offered them voluminous plants with voluminous flowers." for the noun phrase *fleurs volumineuses.* However, all the models predicted anteposition, and this could have been affected by the adjectives being in the same word form.

Finally, in the *blocked/mobile* adjectives category, the speakers did not make any unexplainable choices and always preferred postposition for the postposed adjectives (e.g. chromatic) and both positions for the mobile adjectives (despite the length). The only model which made mistakes on the postposed adjectives was flaubert-large-cased, while the other models made very few mistakes on mobile adjectives –decisions that are to some extent acceptable, since the meaning may be different but still grammatical.

6.7 Discussion

In this work, we studied the capabilities of transformer-based language models in taking word order into account, specifically the position of adjectives in a noun phrase in French. In our first and third experiments, we used finetuned models, based on the existing research claiming that large pretrained models are not sensitive to word order, but they can be taught to, via finetuning. Our findings demonstrate that the finetuned models were, in fact, able to distinguish between the original and the permutated word order in the classification task with very high accuracy. Yet, in the scope of the corpora we examined, the models were outperformed by a simple frequency-based baseline, and the CNN classifier was very successful as well. This could indicate that the task at hand is quite simple, but the sensitivity to word order they demonstrated here contradicts the multiple findings of Transformer-based architectures to shuffled input (sometimes

beyond the point of legibility). However, our results in the second experiment on pretrained models confirmed previous ones which argued that these models are agnostic to word position.

Finetuning with a larger and more varied dataset (two corpora, frWaC and UD) was beneficial, but it was successful with a smaller dataset too, unlike our baselines. Previous research on transformer models has confirmed that the size of the training data and the effectiveness of their frequency learning are key factors in their performance in NLP tasks.

Concerning the use of attention masks, allowing attention only to the noun phrase (adjective and noun) affected the models in different ways. The CamemBERT models were very capable of classifying word order by only attending to the adjective and noun, while for the Flaubert models, it was impossible. Meanwhile, only attending to the context without the adjective and noun was relatively harmless for all models. When the models' attention mechanism only has access to the context, and not to the adjective-noun pair itself, they were still somewhat capable of classifying adjective position, even without the attention mechanism having access to it. This observation is consistent with the linguistic observation that adjective position is also determined by context and not solely by the noun phrase. However, the fact that CamemBERT models were extremely successful in identifying position without the use of context, while Flaubert models failed completely, is caused by the models' different architectures and choices in the way the tokens are handled. In our more detailed experiments, we saw that CamemBERT models assign an overall higher probability to adjectives, regardless of their position, and that, at least for the UD dataset, the adjective embeddings were, in some layers, very informed on the preferred word position. This knowledge is correlated to the learned contextual word embeddings, rather than the word itself, as we observed a lack of semantic similarity in the visualization.

For most adjectives, predicting their position is a relatively easy decision based on frequency; to observe the models' underlying competencies in more complex cases, we carried out an error analysis and additional experiments and visualizations on the pretrained versions of the models. The differences between the two architectures were also reflected in our study of the pretrained word embeddings and the adjective probabili-

ties, where we noticed that CamemBERT's adjective embeddings were better informed. Speaking of adjective embeddings, the way that the embeddings are created seems to put more emphasis on the context than the word-specific information of the corresponding token. Examining the iterations of an adjective in different sentences did not demonstrate a pattern of behavior, akin to vector similarity in traditional embeddings. We noticed that classification with the adjective word embeddings was only successful for certain layers and certain models, but was unpredictable between our two datasets. We could not observe embedding clusters of the same adjective with regard to their position, either.

These findings suggest that the contextualized word embeddings include some information on a word's preferred word order, but only in certain layers. Finetuning a model helps to learn these variations in adjective position and very successfully select the correct one. Overall, models tend to favor frequent adjectives and contextual information, rather than the content of the adjective (its meaning and class). However, adjective position relies on frequency, hence the success of the uninformed frequency baseline and the success of our models. CamemBERT models were more successful than FlauBERT models over all experiments and captured more positional information in the finetuned adjective embeddings. However, all transformers models show weaknesses (to different degrees) in complex cases of adjective/noun-dependent phrases and fixed expressions.

Regarding the models' mistakes, the very few ones that were made (by the finetuned models) were justifiable to some extent and were either caused by low-frequency adjectives, bad parsing, or ambiguous meaning which is grammatical and acceptable in both adjective positions. Additionally, we noted a few sentences in our questionnaire dataset that might appear unnatural to native speakers, but they were either found in existing corpora or designed to be challenging by a French native speaker. However, comparing the models to human performance showed their true strengths and weaknesses; when they are successful, the models tend to follow a more rigid syntactic structure and favor postposition, as it is the most frequent adjective position over all adjectives. They showed severe problems in recognizing some fixed expressions and were more easily swayed than humans by being primed with the same adjective. In cases where both positions were possible, they usually preferred the more "traditional" postposition. These

findings may demonstrate that the models base their predictions more on frequency rather than the syntactic and semantic information of a particular adjective, and are impervious to factors that affect speakers' decisions such as length, the difficulty of processing with regard to cognitive load, and substantial or subtle semantic differences.

6.8 Appendix: Questionnaire datasets

Originally anteposed sentences		
Anteposition	**Postposition**	**Translation**
Ces fiers époux attendent avec impatience le jour J.	Ces époux fiers attendent avec impatience le jour J.	These proud spouses are eagerly awaiting the go time.
Cette fière équipe de travail se hâte de présenter son projet.	Cette équipe fière de travail se hâte de présenter son projet.	This proud work team is eager to present its project.
Cette longue saison de football a été intense.	Cette saison longue de football a été intense.	This long football season has been intense.
Elle connait ce fier artiste depuis des années.	Elle connait cet artiste fier depuis des années.	She has known this proud artist for years.
Il a écrit un long article de linguistique.	Il a écrit un article long de linguistique.	He wrote a long article on linguistics.
Ils ont emprunté un long chemin sans visibilité.	Ils ont emprunté un chemin long sans visibilité.	They took a long path without visibility.
J'ai lu un long roman comme je les aime.	J'ai lu un roman long comme je les aime.	I read a novel, long as I like them.
Les fiers ouvriers déjeunent actuellement.	Les ouvriers fiers déjeunent actuellement.	The proud workers are currently having lunch.
Ma tante est une fière cuisinière de renom.	Ma tante est une cuisinière fière de renom.	My aunt is a proud cook of renown.
Elle a participé à un long séminaire de quelques jours.	Elle a participé à un séminaire long de quelques jours.	She participated in a seminar lasting a few days.
Il a écrit un long article de 50 pages.	Il a écrit un article long de 50 pages.	He wrote a 50 page long article.
Ils ont emprunté un long chemin de plusieurs kilomètres.	Ils ont emprunté un chemin long de plusieurs kilomètres.	They took a path several kilometers long.
J'ai lu un long roman de plusieurs tomes.	J'ai lu un roman long de plusieurs tomes.	I read a novel several volumes long.

Table 6.13 continued in next page.

Originally postposed sentences		
Elle annote un différent segment de 32 caractères.	Elle annote un segment différent de 32 caractères.	She annotates a different segment of 32 characters.
Ils vivent une différente relation sans amour.	Ils vivent une relation différente sans amour.	They live a different relationship without love.
L'architecte a construit une différente maison dans le sud.	L'architecte a construit une maison différente dans le sud.	The architect built a different house in the south.
Tu as acheté un différent cahier pour dessiner.	Tu as acheté un cahier différent pour dessiner.	You bought a different notebook to draw.
Vous avez couru un différent marathon toujours populaire.	Vous avez couru un marathon différent toujours populaire.	You ran a different, ever-popular marathon.
Ces fiers époux de leurs préparatifs attendent avec impatience.	Ces époux fiers de leurs préparatifs attendent avec impatience.	These spouses proud of their preparations are waiting impatiently.
Cette fière équipe de son projet se hâte de le présenter.	Cette équipe fière de son projet se hâte de le présenter.	This team, proud of its project, is eager to present it.
Cette longue saison de 4 mois a été intense.	Cette saison longue de 4 mois a été intense.	This 4 month long season has been intense.
Elle annote un différent segment du précédent.	Elle annote un segment différent du précédent.	It annotates a different segment from the previous one.
Elle connait ce fier artiste de sa création.	Elle connait cet artiste fier de sa création.	She knows this artist who is proud of his creation.
Ils vivent une différente relation de la suivante.	Ils vivent une relation différente de la suivante.	They live a different relationship than the following one.
L'architecte a construit une différente maison de celle prévue.	L'architecte a construit une maison différente de celle prévue.	The architect built a different house than planned.
Les fiers ouvriers de leur avancement s'accordent une pause.	Les ouvriers fiers de leur avancement s'accordent une pause.	The workers, proud of their advancement, take a break.
Ma tante est une fière cuisinière de ses talents.	Ma tante est une cuisinière fière de ses talents.	My aunt is a cook proud of her talent.
Tu as acheté un différent cahier du sien.	Tu as acheté un cahier différent du sien.	You bought a notebook different from his.
Vous avez couru un différent marathon de celui de Toulouse.	Vous avez couru un marathon différent de celui de Toulouse.	You ran a different marathon than that of Toulouse.

Table 6.13: Sentences in the *Presence of adjective/noun dependent* category.

Originally anteposed sentences		
Anteposition	**Postposition**	**Translation**
Dimanche, ils ont pu faire la grasse matinée.	Dimanche, ils ont pu faire la matinée grasse.	On Sunday, they were able to sleep in.
Elle a écrit un vibrant hommage pour sa mère décédée.	Elle a écrit un hommage vibrant pour sa mère décédée.	She wrote a moving tribute for her late mother.
Elle aime la grasse matinée du lundi.	Elle aime la matinée grasse du lundi.	She loves sleeping in on Mondays.
Il a passé une dure semaine.	Il a passé une semaine dure.	He had a tough week.
Il admet son net avantage sur les autres.	Il admet son avantage net sur les autres.	He admits his clear advantage over others.
Il ne retient pas ses diverses leçons.	Il ne retient pas ses leçons diverses.	He does not retain his various lessons.
Ils ont rendu un vibrant hommage à ce digne soldat.	Ils ont rendu un hommage vibrant à ce digne soldat.	They paid a vibrant tribute to this worthy soldier.
J'avais des doubles objectifs précis.	J'avais des objectifs doubles précis.	I had specific dual objectives.
Nous effectuons diverses expériences.	Nous effectuons des expériences diverses.	We perform various experiments.
Elle a fait un net bénéfice ce mois-ci.	Elle a fait un bénéfice net ce mois-ci.	She made a net profit this month.
Originally postposed sentences		
Depuis la mort de son hamster, il a le dur cœur.	Depuis la mot de son hamster, il a le cœur dur.	Since the death of his hamster, he has had a hard heart.
Depuis la mort de son hamster, il a une dure vie.	Depuis la mort de son hamster, il a une vie dure.	Since the death of his hamster, he has had a hard life.
Dimanche, ils ont mangé des gras plats.	Dimanche, ils ont mangé des plats gras.	On Sunday, they ate fatty dishes.
Elle essaiera par elle-même pour en avoir le net cœur.	Elle essaiera par elle-même pour en avoir le cœur net.	She will try on her own to find out for sure.
Elle n'aime pas laver la grasse boîte.	Elle n'aime pas laver la boîte grasse.	She doesn't like to wash the greasy box.
Il est adepte de divers faits.	Il est adepte de faits divers.	He is adept at various facts.
Il n'a pas accepté sa défaite, il a le dur cœur.	Il n'a pas accepté sa défaite, il a le cœur dur.	He did not accept his defeat, he has a hard heart.
Ils ont acheté un vibrant fauteuil pour leur salon.	Ils ont acheté un fauteuil vibrant pour leur salon.	They bought a vibrant armchair for their living room.
J'ai mis les doubles bouchées pour arriver à temps.	J'ai mis les bouchées doubles pour arriver à temps.	I worked hard to get there on time.
Nous suivons les divers faits à la télévision.	Nous suivons les faits divers à la télévision.	We follow the news on television.
Vous avez mis les doubles bouchées pour terminer.	Vous avez mis les bouchées doubles pour terminer.	You worked hard to finish.

Table 6.14: Sentences in the *Fixed expressions* category.

Originally anteposed sentences		
Anteposition	**Postposition**	**Translation**
A nouvelle année, nouveaux dynamismes pour cette entreprise.	A nouvelle année, dynamismes nouveaux pour cette entreprise.	A new year, new dynamics for this company.
Fabuleux amis, fabuleux camarades : l'ennemi n'est pas à l'intérieur !	Fabuleux amis, camarades fabuleux : l'ennemi n'est pas à l'intérieur !	Fabulous friends, fabulous comrades: the enemy is not within!
J'ai aimé le concept : bonne ambiance, bonne musique, les gens sont contents.	J'ai aimé le concept : bonne ambiance, musique bonne, les gens sont contents.	I liked the concept: good atmosphere, good music, people are happy.
Ce document vise à expliquer le déficit véritable, la véritable dette dans son ensemble.	Ce document vise à expliquer le déficit véritable, la dette véritable dans son ensemble.	This document aims to explain the real deficit, the real debt as a whole.
Nous avons adopté pour des stratégies communes, actions communes et positions communes.	Nous avons adopté pour des stratégies communes, actions communes et communes positions.	We have adopted for common strategies, common actions and common positions.
Avec la merveilleuse sélection et de merveilleux essais, ils ont trouvé les résultats qu'ils cherchaient.	Avec la merveilleuse sélection et des essais merveilleux, ils ont trouvé les résultats qu'ils cherchaient.	With the wonderful selection and wonderful testing, they found the results they were looking for.
Originally postposed sentences		
Il lui a offert des volumineuses plantes à volumineuses fleurs.	Il lui a offert des volumineuses plantes à fleurs volumineuses.	He gave her bulky plants with voluminous flowers.
Je suis d'accord avec eux : à événement exceptionnel, exceptionnel dispositif.	Je suis d'accord avec eux : à événement exceptionnel, dispositif exceptionnel.	I agree with them: for an exceptional event, an exceptional device.
Cette année, ils préparent un diplôme professionnel en professionnel lycée.	Cette année, ils préparent un diplôme professionnel en lycée professionnel.	This year, they are preparing a professional diploma in vocational high school.
Concernant la protection des données personnelles, aucune personnelle information n'est collectée.	Concernant la protection des données personnelles, aucune information personnelle n'est collectée.	Regarding the protection of personal data, no personal information is collected.
Elle a procédé à l'étude de quelques instruments pitoyables et pitoyables illusions.	Elle a procédé à l'étude de quelques instruments pitoyables et illusions pitoyables.	She proceeded to study some pitiful instruments and pitiful illusions.
Ce bâtiment n'a pas changé depuis sa construction : lumineuses couleurs, lumineux lampadaires.	Ce bâtiment n'a pas changé depuis sa construction : lumineuses couleurs, lampadaires lumineux.	This building has not changed since its construction: bright colors, bright streetlights.

Table 6.15: Sentences in the *Structural persistence* category.

Originally anteposed sentences		
Anteposition	**Postposition**	**Translation**
Elle préfère son propre pantalon à celui de sa soeur.	Elle préfère son pantalon propre à celui de sa sœur.	She prefers her own pants to her sister's.
Nous nous sommes rejoins autour d'un chaleureux repas.	Nous nous sommes rejoins autour d'un repas chaleureux.	We came together for a hearty meal.
Tu m'as fait part de ta fabuleuse idée.	Tu m'as fait part de ton idée fabuleuse.	You told me about your fabulous idea.
Cet ancien fer n'est plus utilisé.	Ce fer ancien n'est plus utilisé.	This old iron is no longer used.
Originally postposed sentences		
C'était un fabuleux voyage que nous avons organisé.	C'était un voyage fabuleux que nous avons organisé.	It was a fabulous trip that we organized.
Ce chaleureux accueil m'a fait chaud au cœur.	Cet accueil chaleureux m'a fait chaud au cœur.	This warm welcome warmed my heart.
Ce légendaire récit me tourmente chaque jour.	Ce récit légendaire me tourmente chaque jour.	This legendary tale torments me every day.
Ce puéril discours lui a porté préjudice.	Ce discours puéril lui a porté préjudice.	This childish speech harmed him.
Cette fermière entreprise n'est plus aussi familiale que dans le temps.	Cette entreprise fermière n'est plus aussi familiale que dans le temps.	This farm business is no longer as family-run as it used to be.
Cette jaune chaise est très tendance.	Cette chaise jaune est très tendance.	This yellow chair is very trendy.
Cette puérile plaisanterie ne l'a pas fait rire.	Cette plaisanterie puérile ne l'a pas fait rire.	This childish joke did not make him laugh.
Elle m'a fourni la volumineuse archive.	Elle m'a fourni l'archive volumineuse.	She provided me with the voluminous archive.
Il m'a apporté une bleue gourde.	Il m'a apporté une gourde bleue.	He brought me a blue water bottle.
Il mange des roses bonbons.	Il mange des bonbons roses.	He eats pink candies.
Ils n'ont pas pu télécharger le volumineux fichier.	Ils n'ont pas pu télécharger le fichier volumineux.	They were unable to download the large file.
J'ai écrit sur une bleue feuille.	J'ai écrit sur une feuille bleue.	I wrote on a blue sheet.
La jaune trousse contient ses feutres.	La trousse jaune contient ses feutres.	The yellow pencil case contains her markers.
La pétrolière industrie ne m'attire pas du tout.	L'industrie pétrolière ne m'attire pas du tout.	The oil industry does not appeal to me at all.
Le ferroviaire transport est voué à s'étendre.	Le transport ferroviaire est voué à s'étendre.	Rail transport is destined to expand.

Table 6.16 continued in next page.

Anteposition	Postposition	Translation
Le ministériel arrêté a confirmé les mesures prises.	L'arrêté ministériel a confirmé les mesures prises.	The ministerial decree confirmed the measures taken.
Les filles ont opté pour une mauve couverture.	Les filles ont opté pour une couverture mauve.	The girls opted for a purple blanket.
Leur financière situation s'aggrave de jour en jour.	Leur situation financière s'aggrave de jour en jour.	Their financial situation is getting worse day by day.
Ma sœur porte des mauve lunettes.	Ma sœur porte des lunettes mauve.	My sister wears purple glasses.
Mon bureau est décoré d'un vert panier.	Mon bureau est décoré d'un panier vert.	My office is decorated with a green basket.
Sa rose poubelle lui plait énormément.	Sa poubelle rose lui plait énormément.	His pink trash can pleases him enormously.
Son doudou est une verte peluche.	Son doudou est une peluche verte.	His cuddly toy is a green plush.
Elle a acheté un vibrant jouet pour son fils.	Elle a acheté un jouet vibrant pour son fils.	She bought a vibrant toy for her son.

Table 6.16: Sentences in the *Blocked and mobile adjectives* category.

Chapter 7

Conclusion

In the Introduction, we posed the research questions that we aimed to answer with our experiments. Reflecting on the outcomes of the experiments, we have formed our opinion on the linguistic competencies of contextual word embeddings, and how the models' architectures and features deal with language.

To the question of whether **contextual word embeddings capture context sufficiently and effectively**, our answer leans positive. This is no news since they have been proven to yield better results than static word embeddings. It is more interesting to discuss *how* this contextual information is captured. By examining the different Transformer-based architectures that created these embeddings, we observe one common denominator; the pretraining process needs an immense number of data, even for the smaller pretrained models. The Transformer neural architecture is able to process this enormous volume of data in a dynamic way, extracting patterns based on the multi-headed self-attention mechanism. The masked language modeling objective is used by all the models examined in this doctoral work to create their respective pretrained embeddings. Seemingly, it is able to assess which of these patterns are the most fitting for the masked position in a sequence, but this success is a result of frequency, not any linguistic competence.

The models are linguistically agnostic, hence they treat each token like a piece of information and each masked token as a token void of any features other than its context. In preliminary experiments, we noticed that the task of prediction in the masked position yielded irregular results. At a first glance, they seem to usually predict a word fitting to the context at the masked position. When the masked token was not a verb, the models

showed affinity to predicting frequent tokens, such as adjectives of size, pronouns, and punctuation. However, in cases where the context was vague or complex, the model would predict pronouns or punctuation marks instead of adjectives or nouns. These experiments were, unfortunately, not fruitful, so they were not further pursued. To sum them up in a few words, the pretrained models' choices for a masked position are at best "the safest bet" and at worse a bland platitude, a nonsensical word, or the occasional offensive term.

Instead, we focused on only using the masked language prediction by injecting the originally masked word in said position and retrieving its probability (see Sections 4.4.3, 6.6). The models were not essentially unsuccessful at this task, but their behavior differs significantly from human choices. They assign high probabilities to frequent tokens, even if they are not the best fit for the given context. These tokens have been assigned a high likelihood because they have been frequently observed in many similar contexts. Likewise, tokens with a lower frequency but a better semantic fit to the masked position receive a lower probability, because of their overall infrequency.

The modeling of the embeddings is based exclusively on contextual information, meaning that different instances of the same word are treated and encoded differently. The models are not able to create classes or clusters of word embeddings, based on the content of the embeddings. In Section 6.5.3 we studied whether the properties of dynamic word embedding vectors corresponded to those of static word embedding vectors, by plotting different instances of the same word—plotting lemmas and word types. Intuitively, we expected clusters to appear, based on the similarity of contextual information in vectors of the same word, but it was not the case. In Section 4.5.2, we could not identify either discernible patterns of behavior of head words (verbs, nouns) choosing their constituents (masked words) based on the head word's class. We noticed some weak preferences, such as verbs selecting their subject and object based on animacy.

In the experiments of extracting and probing specific word embeddings, we also had the opportunity to study specific word embeddings further. In Section 5.5.4.4 we examined whether verb embeddings were able to capture lexical aspect information—based on the temporal features of the context, i.e. the verb's preferred company. This classification task was quite successful, especially for classifying duration. However, our

experiment in Section 6.5.1 on classifying adjective position based on the adjective embedding was not as successful. In Layman's terms, not all embeddings are created equal. Verb embeddings may contain more interesting information since they tend to put more constraints on the context than other tokens. Meanwhile, adjectives do not have a such transformative influence on their context so their pretrained embeddings encode enough information about the adjective's preferences.

This brings us to the next question, whether **contextual word embeddings show sensitivity to semantics**. Our three research topics explored linguistic phenomena that influence the acceptability of a sentence; the semantic preferences of a verb for its dependents, the temporal properties of a verb and their interaction with the context, and the preferred word order when it is semantically salient. In the experiments on selectional preferences, the pretrained models have shown that they can capture contextual information of individual word embeddings and that they have a preference for frequency and for semantic felicity. However, the finetuned models showed a degree of specialization on the given tasks, hence our choice to mainly use them for our experiments and observations in lexical aspect and word order. We used manually created datasets with small sentences (a slightly adversarial approach, since contextual word embeddings rely on context), a simple structure without complex syntactic phenomena, and carefully selected constituents to either complement or challenge the annotation category of the sentence.

We cannot argue that the models have no syntactic competencies; pretrained and finetuned models showed that they are able to distinguish the most important parts of a one-sentence input, i.e. the verb of the main clause, its auxiliary verb(s), subject, and object(s), and the verb of the subordinate clause. Their perception of syntax is not comparable to human syntactic abilities, and it should not be compared to human cognition, either. However, the models seemed to not rely on prepositional phrases, adverbs, and attributive adjectives to process a sentence and classify its properties. They have (mostly) successfully captured preferred literal contexts, and in cases of metaphors or antagonistic context, they prefer to "ignore" these discrepancies. Unfortunately, these elements are quintessential to human communication and convey important syntactic and semantic information, which led to incorrect predictions when ignored.

A serious motivation for our experiments was the correlation of the models to human behavior. We do not aim to support or reproduce the argument that the models can or should mimic human language production. Our goal was to observe human linguistic preferences, which did not necessarily match perfectly the assessments made by linguists, and compare them to the models' learned behavior. The term "stochastic parrot" coined by Bender and Koller (2020) comes to mind; while we agree that the models rely on repetition, we would like to append the adjective "stubborn" to this characterization. The architectures learn rudimentary syntactic patterns and insist on their use even when the context suggests otherwise.

Is there any hope to improve these contextual word embedding models? Focusing on the role of **finetuning in NLP tasks**, we have supported, throughout this doctoral work, that finetuning is beneficial to create embeddings with additional knowledge. Even though our experiments were testing limited phenomena and our results were not always impressive percentages that show quantitative success, we exhibited that transfer learning can sensitize models to linguistic phenomena, to a certain extent. The possible scope of use of these finetuned models may seem limited, as the community is focused on models for downstream tasks of a wider scope. Regardless, we hope that our findings on finetuning will be beneficial to the NLP community.

As for our technical criticism of finetuning, we observed that it is, in fact, not a very stable process. Random seeds in the finetuning process can cause the failure of classification, and we agree with the bibliography that larger models can produce unpredictable results. The base-size models were more accepting of finetuning, and a few epochs of finetuning were sufficient for our experiments. As for the recommended sizes of finetuning datasets, our experiments in word order showed that the models were "data-hungry"; larger and varied datasets performed better overall. However, our lexical aspect experiments were also successful with much smaller finetuning datasets. From the two rounds of experiments, we observed that finetuning with data of good quality is paramount to the success of finetuning. Combining datasets of different domains with the same annotation objective was also quite beneficial in all our experiments, since it introduced a variety in the data, making the finetuned models more robust to different test inputs.

An essential part of our research was to observe the **self-attention mechanism** and assess its abilities and its influence on the predictions of the models. Self-attention is the backbone of the Transformer. It is the reason the architecture was conceived and it is the reason for the neural network models' advanced capacities compared to their predecessors. We followed two methods of examining self-attention, either by using attention masks for predictions and finetuning or by visualizing layers and attention heads of the models. In our multiple experiments with attention masks, we observed different behaviors, sometimes anticipated, sometimes surprising. For the selectional preferences experiments, where the pretrained model had to predict the masked dependent word, focusing the models' attention only on the head word produced better correlations to human behavior than any other attention setting. As stated above, verb embeddings may contain a bigger breadth of information than other word embeddings, hence their bigger influence. Additionally, allowing the model to focus on the one token of the sequence that is the most important for the given task may have contributed to the prediction successes. However, in the case of classifying lexical aspect, we recreated the finetuning and classification task with an attention mask on the context and noticed a decline in accuracy. While this finding contradicted the importance of the verb embeddings for predictions, it may be caused by the finetuning process modifying the embeddings with additional information, thus distributing information in the entire sentence.

A long debate has been brewing, on whether attention should be taken as a solid metric of explainability and success in the interpretation of neural architecture output—and the popularity of the much more complex and less decipherable multi-headed self-attention has fanned the flames. We visualize pretrained and finetuned models, and we find tendencies, but not concrete answers. First of all, the only model that produced visually interesting findings was BERT, compared to RoBERTa and ALBERT which showed diffused attention weights without strong preferences. Some particular layers and heads produced consistently similar attention visualizations regardless of the input sentence, e.g. the third layer of bert-base-uncased producing monotonic attention from token t to token $t + 1$. We noticed that the strongest trends of attention occurred between verb tokens (head word or of the subordinate clause) and subjects, objects, and auxiliary verbs. To corroborate the bibliography, we also observed that certain layers of the architec-

tures show patterns of attention that are akin to syntactic relations. In conclusion, while multi-head self-attention is difficult to decipher and study, it can be an interesting observation tool for the models' inner workings and order of operations when they process language and produce output.

Finally, to the question of **word order importance**, if the models were truly agnostic to word order, they would treat inputs with shuffled tokens in the same way. This was the case neither in our word order experiments nor in the lexical aspect classification experiments in which word order was not a finetuning objective (see Section 5.5.4.2). The finetuned models were able to classify the preferred adjective word order very successfully with the classification datasets, in Section 6.4. While the task may have been quite simple, since a frequency-based metric also yielded good results, it would have been impossible for the models, if they were completely agnostic to word order. In addition, they treated grammatical and acceptable sentences of the same tokens in different ways, classifying their temporal qualities differently. While pretrained models may be insensitive to permutations, finetuned models can be made sensitive to permutations.

*

* *

Modeling and recreating the natural world and the human experience with computational methods has been a longtime aspiration for humans. The futuristic utopias (and dystopias) in science fiction literature seem more attainable than ever with the public release of artificial intelligence advancements, such as synthetic media, voice assistants, and toolkits for large language models (LLMs). Especially in the last few years, these tools have left the confines of academic and commercial research and have become accessible to use by a fraction of the human population (those who have access to education, technology, technological literacy, and freedom of speech). Meanwhile, they have been already exploited (and weaponized) for economic and political profit.

The scientific community has focused its efforts on studying and decrypting contextual word embedding models for the last few years, yet many questions are unanswered. At the same time, these models are vastly used for natural language processing tasks, sometimes blurring the lines between ethical and responsible use. With this doctoral

thesis, motivated by linguistic sensitivities, we hope to have contributed to shedding light on the "black box" of these models. They are potent, but they should and can be improved. These dynamic and powerful tools were capable of capturing some important linguistic information, from the pretraining and the finetuning stages. Hopefully, this may motivate future research to improve these models with the incorporation of targeted linguistic and semantic competencies. We want to face the future of natural language processing with optimism. They should not be dealt with as shallow copies of human speech, convincing but off-putting, force-fed more words than a lifetime of human activity, and confined to a superficial understanding of the world. Our large language models will only be as good as we allow them to be.

Chapter 8

Abstracts

8.1 Abstract in English

Transformer-based embeddings, also known as large language models, are being widely used in NLP applications, outperforming traditional methods and neural network approaches. However, quantitative success in NLP tasks does not guarantee a complete mastery of human language. Humans are capable of learning semantic concepts and expressing them with the appropriate syntactic patterns, while Transformer-based language models learn artifacts and idiosyncratic patterns of syntax, but no notions of semantics.

This doctoral thesis studies the linguistic abilities and limitations of Transformer-based contextual word embeddings, with experiments on complex syntactic-semantic phenomena. The main question is: even though contextual word embeddings can capture enough information to be competent in complex linguistic tasks, are their successes due to a true understanding of word relations and hierarchies or a repetition of language patterns? We selected linguistic features in English and French that are understood by native speakers with mature syntactic-semantic competencies but have been traditionally hard to define with linguistic rules.

Selectional preference is the tendency of a predicate to favor certain arguments within a certain linguistic context and reject others that result in conflicting or implausible meanings. This part of the study investigated whether BERT models in English contain information on the selectional preferences of words, by examining the probability it assigns to the dependent word given the presence of its head word in a sentence. These

probabilities were compared to human annotations. Results show that there is no strong positive or negative correlation between human judgments and model probabilities in any syntactic relation, but certain head words have a strong correlation, and masking all words but the head word yields the most positive correlations in most scenarios.

Lexical aspect is a verb feature that describes how an action, event, or state of a verb is situated in time regardless of verb tense. We explored, with two rounds of experiments, whether the models can identify and learn telicity and duration. We performed quantitative analyses with pretrained and finetuned models, and qualitative analyses to observe the models' behavior in challenging cases. Experiments were carried out in English and French. Results show that the models capture information on telicity and duration in their vectors, but are biased concerning verb tense and word order.

The final experiment examines the models' capacities for identifying and learning attributive adjective position in French. Even though these models are insensitive to permutated word order by design, we observed that the finetuned models could learn and select the correct position of the adjective. However, this is attributed to finetuning rather than knowledge learned during pretraining. Comparing the finetuned models to native speakers, we notice that the models favor context and global syntactic roles, and are weaker with complex structures and fixed expressions.

To summarize our findings, contextual word embeddings are very successful, but results are irregular. The models assign high probabilities to frequent tokens, but cannot create classes or clusters of word embeddings based on content. Verb embeddings can capture important syntactic-semantic information, but adjectives do not have a transformative influence. The models show sensitivity to syntax and learn rudimentary syntactic patterns. Semantically, the models rely on frequency and surface-level features, even when the context suggests otherwise.

8.2 Abstract in French

Les plongements lexicaux basés sur des Transformers, également connus comme grands modèles de langage, sont largement utilisés dans les applications TALN, surpassant les méthodes de statistique et de réseaux neuronaux. Cependant, le succès quantitatif dans les tâches de TALN ne garantit pas une maîtrise complète du langage humain.

Cette thèse étudie les capacités linguistiques et les limites des plongements lexicaux contextuels basés sur Transformers, avec des expériences sur des phénomènes syntactico-sémantiques complexes. La question principale est la suivante: même si les plongements lexicaux peuvent capturer suffisamment d'informations pour être compétents dans des tâches linguistiques complexes, leurs succès sont-ils dus à une véritable compréhension des relations et des hiérarchies entre les mots ou à une répétition de schémas de langue? Nous avons sélectionné des caractéristiques linguistiques en anglais et en français qui sont comprises par les locuteurs natifs ayant des compétences syntaxico-sémantiques matures, mais qui sont traditionnellement difficiles à définir avec des règles linguistiques.

La préférence sélective est la tendance d'un prédicat à favoriser certains arguments dans un certain contexte linguistique et à en rejeter d'autres qui aboutissent à des significations contradictoires ou peu plausibles. Cette partie de l'étude a examiné si les modèles BERT en anglais contiennent des informations sur les préférences sélectives, en examinant la probabilité qu'ils attribuent au mot dépendant compte tenu de la présence de son mot principal dans une phrase. Ces probabilités ont été comparées aux annotations humaines. Les résultats montrent qu'il n'y a pas de forte corrélation entre les jugements humains et les probabilités du modèle dans n'importe quelle relation syntaxique, mais certains mots de tête ont une forte corrélation, et le masquage de tous les mots sauf le mot de tête produit les corrélations les plus positives.

L'aspect lexical est une caractéristique du verbe qui décrit comment une action, un événement ou un état d'un verbe est situé dans le temps, indépendamment du temps du verbe. Nous avons exploré, avec deux séries d'expériences, si les modèles peuvent identifier et apprendre la télicité et la durée. Nous avons effectué des analyses quantitatives avec des modèles pré-entraînés et affinés, ainsi que des analyses qualitatives

pour observer le comportement des modèles dans des cas difficiles. Les expériences ont été menées en anglais et en français. Les résultats montrent que les modèles capturent l'information sur la télicité et la durée dans leurs vecteurs, mais qu'ils sont biaisés en ce qui concerne le temps du verbe et l'ordre des mots.

La dernière expérience examine les capacités des modèles à identifier et apprendre la position des adjectifs attributifs en français. Bien que ces modèles pré-entraînés soient insensibles à l'ordre des mots permutés, nous avons observé que les modèles affinés pouvaient apprendre et sélectionner la position correcte de l'adjectif. En comparant les modèles aux locuteurs natifs, on remarque que les modèles favorisent le contexte et les rôles syntaxiques globaux, et qu'ils sont plus faibles avec les structures complexes et les expressions fixes.

Pour résumer, les plongements lexicaux sont très efficaces, mais les résultats sont irréguliers. Les modèles attribuent des probabilités élevées aux tokens fréquents, mais ne peuvent pas créer de classes ou de groupes de mots selon le contenu. Les plongements de verbes peuvent capturer des informations syntactico-sémantiques importantes, mais les adjectifs n'ont pas d'influence. Les modèles sont sensibles à la syntaxe et apprennent des schémas syntaxiques rudimentaires. Sur le plan sémantique, les modèles s'appuient sur des caractéristiques de fréquence et de surface, même lorsque le contexte suggère le contraire.

8.3 Long abstract in French

Qu'est-ce que tu sais, BERT? Explorer les compétences linguistiques des plongements lexicaux contextuels basés sur Transformers

Le traitement du langage naturel (TALN) s'est traditionnellement concentré sur la définition et la conception de systèmes pour le traitement, la compréhension et la production du langage, avec la motivation que le succès de ces tâches se traduirait par des systèmes linguistiques compétents pour des applications en aval. Les applications du TALN comprennent des tâches de classification au niveau de la phrase ou du document (par exemple, la classification des sentiments), des tâches d'étiquetage de séquences au niveau du mot ou de la phrase (par exemple, l'analyse syntaxique, la reconnaissance des entités nommées), la classification des relations de portée, et des tâches de génération, qui impliquent la création d'un texte sur la base d'une entrée donnée (par exemple, la traduction automatique, la génération de dialogues, la production de discours).

Ces architectures algorithmiques spécifiques à une tâche peuvent être combinées avec d'autres modèles pour exécuter des tâches complexes et peuvent elles-mêmes être composées de différents modèles, par exemple des tokenizers et des marqueurs de partie du discours. Construits à l'origine par des linguistes à l'aide de règles écrites à la main, l'utilisation de méthodes statistiques avancées de régression logistique et de modèles de réseaux neuronaux est aujourd'hui devenue la norme dans la plupart des applications du TALN.

Ces dernières années, la mise en œuvre des architectures de réseaux neuronaux a connu des développements monumentaux et les capacités de traitement, de compréhension et de production du langage ont fait des bonds de géant. Ces développements ont permis la création d'architectures Transformer pour le TALN qui créent des plongements lexicaux contextuels. Ces architectures sont entraînées avec d'énormes ensembles de données sur plusieurs unités informatiques avec une puissance de traitement massive et produisent des représentations du langage sous la forme de modèles des plongements lexicaux contextuels. Même si ces modèles sont très performants, il est difficile de les examiner et de les comprendre avec les méthodes traditionnelles de TALN. De nombreuses

études et discussions ont été menées pour déterminer si ces calculs de sont interprétables.

L'objectif de cette thèse est d'étudier les capacités et limites linguistiques des plongements lexicaux contextuels basés sur les Transformers, avec des expériences sur des phénomènes syntactico-sémantiques complexes. L'hypothèse principale de cette thèse est la suivante: Les plongements lexicaux contextuels de mots peuvent-ils capturer suffisamment d'informations, pendant les phases de pré-entraînement et d'affinage, pour être compétents dans des tâches linguistiques complexes? Leurs succès sont-ils dus à une véritable compréhension des relations et des hiérarchies de tokens ou à une répétition superficielle de modèles dans l'ensemble d'apprentissage? Leurs échecs sont-ils graves, et s'agit-il de faiblesses systématiques ou d'événements aléatoires?

Nous avons sélectionné des caractéristiques et des phénomènes linguistiques qui sont facilement perçus par un locuteur natif ayant des compétences syntactico-sémantiques matures, mais qui sont traditionnellement difficiles à définir à l'aide de règles linguistiques. Plus précisément, nous nous sommes concentrés sur:

- Préférences de sélection, c'est-à-dire les arguments et les classes d'arguments qui complètent le mieux le sens du verbe, ce qui se traduit par des phrases grammaticales et sémantiquement acceptables.
- L'aspect lexical, c'est-à-dire un ensemble de caractéristiques qui déterminent les qualités temporelles d'un verbe indépendamment des caractéristiques grammaticales telles que le temps
- Ordre des mots des adjectifs épithètes en français, une tâche apparemment simple mais parfois complexe, en raison de la mobilité des adjectifs basée sur des facteurs linguistiques, non linguistiques et sémantiques.

Ci-dessous les questions spécifiques qui seront abordées par les approches proposées dans cette thèse :

- Les plongements lexicaux de mots contextuels capturent-ils le contexte de manière suffisante et efficace? Cette question est une observation sur les mots intégrés pré-entraînés. Au cours de la procédure de pré-entraînement, les plongements lexicaux

sont-ils capables de généraliser et de regrouper les contextes en classes, c.-à-d. groupes de similarité auxquels le modèle peut accéder pour faire de meilleures prédictions?

- Les plongements lexicaux contextuels de mots montrent-ils une sensibilité aux caractéristiques sémantiques et à la félicité sémantique? Certains phénomènes tels que l'aspect lexical sont des propriétés inhérentes et ne sont pas toujours exprimés morphologiquement ou à l'aide du contexte. Les modèles ont-ils encodé suffisamment d'informations afin d'identifier avec succès de tels phénomènes? Lorsqu'ils sont confrontés à une phrase infélicite, les modèles la rejettent-ils en raison de sa faible fréquence ou d'un certain motif de sémantique?

- L'affinage est-il nécessaire, bénéfique et stable pour les tâches difficiles? L'apprentissage par transfert est l'une des fonctionnalités les plus innovantes des ces modèles, qui permet aux plongements lexicaux déjà puissants de devenir encore plus spécialisés dans une tâche sans avoir besoin de grands ensembles de données. Cependant, la stabilité d'affinage a été critiquée.

- Quel est le rôle du mécanisme d'attention dans les prédictions, au regard de nos questions expérimentales? La haute performance des modèles a été attribuée à leur architecture de self-attention multi-tête. Les choix de nos modèles se reflètent-ils dans le fonctionnement interne des couches et des têtes du mécanisme d'attention?

- L'ordre des mots est-il vraiment sans importance pour les modèles basés sur les Transformers? La parallélisation de la procédure d'apprentissage dans les modèles Transformer signifie que ces modèles ne considèrent pas l'entrée de manière séquentielle. La recherche a montré une insensibilité à l'ordre des mots, mais les modèles y sont-ils insensibles lorsque l'ordre des mots est déterminé par la signification de la phrase?

Les méthodes de traitement du langage naturel nécessitent généralement la conversion du texte en vecteurs de valeurs numériques. Le codage peut être une procédure succincte de mise en correspondance des valeurs avec un vocabulaire, soit sous forme d'indices, soit sous forme de vecteurs optimisés pour le traitement.

Un modèle d'espace vectoriel ou un modèle des plongements lexicaux est un espace sémantique dans lequel les éléments lexicaux appelés tokens (mots ou termes à plusieurs mots) sont représentés sous forme de vecteurs. Les similitudes vectorielles peuvent être en corrélation avec les similitudes sémantiques, car les mots de la même classe, de la même fonction ou d'une signification similaire sont codés avec des vecteurs similaires sur la base de leurs occurrences similaires dans des contextes multiples. Cela a conduit à l'hypothèse commune que les modèles peuvent capturer des informations sémantiques importantes.

L'idée de modèles linguistiques représentant la sémantique découle de la linguistique structuraliste et de la philosophie du langage. Les premières tentatives de mesurer la similarité sémantique par le biais de représentations de caractéristiques ont utilisé des caractéristiques créées à la main (Osgood et al., 1957). Suite aux progrès de l'apprentissage automatique, des méthodes statistiques ont été introduites, permettant l'extraction de distributions à partir de corpora immenses de manière non supervisée (Mikolov et al., 2013a; Pennington et al., 2014).

Suite à l'approche initiale de Bengio et al. (2000) consistant à capturer les informations distributionnelles à l'aide d'un réseau neuronal, les plongements lexicaux modernes sont créés par des réseaux neuronaux. La création de plongements lexicaux à partir des architectures neuronales est possible grâce à la **couche de plongement** du modèle, qui cartographie la séquence d'entrée en une série de vecteurs. Ces vecteurs sont créés par le modèle dans le cadre d'une tâche d'apprentissage spécifique et contiennent donc toutes les informations apprises nécessaires à cette tâche. Ces réseaux neuronaux comprennent le plus souvent un mécanisme d'attention (Bahdanau et al., 2014; Luong et al., 2015). Vaswani et al. (2017) a introduit une nouvelle méthode d'attention appelée *self-attention*, qui est intégrée dans l'architecture d'un réseau neuronal appelé *Transformer*.

Self-attention est un type de mécanisme d'attention qui permet aux entrées du modèle d'interagir entre elles, contrairement à l'attention générale dans laquelle la sortie interagit avec chaque entrée. En termes simples, la fonction mathématique de self-attention est une correspondance entre une *requête* et un ensemble de paires *clé-valeur*, puis une sortie. Au cours de la procédure d'entraînement, le mécanisme de self-attention apprend la similarité entre une requête et une clé sous la forme d'un *poids d'attention*. Dans le

domaine du TALN, les clés et les valeurs correspondent à l'alignement de l'entrée et de la sortie attendue (par exemple, entre les tokens source et cible dans la traduction automatique) ou de l'entrée et de ses caractéristiques extraites (par exemple, dans la classification). Le résultat du mécanisme de self-attention est calculé comme une somme pondérée des valeurs, le poids de chaque valeur étant déterminé par la fonction de compatibilité de la requête avec sa clé associée.

Ce mécanisme d'attention est capable de générer les poids d'attention de chaque token en observant ses différents états cachés dans la séquence. Il capture des représentations multiples par rapport aux autres tokens, grâce à l'utilisation de plusieurs têtes de self-attention.

Les représentations linguistiques créées par les modèles Transformers sont dynamiques car elles sont capables de représenter un mot dans de multiples instances en fonction de son contexte, capturant ainsi des informations syntactico-sémantiques importantes et variées. En outre, les représentations peuvent être affinées pour une tâche et un ensemble de données donnés, devenant ainsi plus spécialisées avec des connaissances précises. Le succès obtenu dans diverses tâches complexes de traitement de texte a conduit au développement rapide de diverses architectures de Transformers et de modèles capables de produire des plongements lexicaux contextuels. En fonction des modèles étudiés dans ce travail documentaire, les modèles peuvent être auto-régressifs (GPT, XLNet) ou auto-encodeurs (BERT, RoBERTa, ALBERT, CamemBERT, FlauBERT).

Ces modèles ont rapidement surpassé les méthodes traditionnelles et les approches de réseaux neuronaux dans des tests standardisés appelés *benchmarks* ; ces tests mesurent la précision d'une tâche de TALN (par exemple, la traduction automatique) avec un ensemble de données et des mesures de précision prédéterminés et sont largement acceptés comme preuve de compétence dans les applications de TALN.

Cependant, la réussite à une tâche ne garantit pas une maîtrise complète de la langue. En revanche, sur la base des précisions élevées signalées dans de nombreuses tâches de TALN, ces modèles ont également été diffusés pour un usage public et commercial, sans que l'on comprenne vraiment leurs capacités, leurs limites et leurs dangers potentiels (Bender et al., 2021).

La première expérience que nous avons menée s'est concentrée sur le modèle BERT

en anglais. Nous avons exploré les *préférences sélectives* d'un mot, c'est-à-dire le type d'arguments et de significations avec lesquels un mot préfère être lié. La question de recherche est de savoir si les plongements lexicaux de BERT contiennent des informations sur les préférences sélectives des mots, en examinant la probabilité qu'ils attribuent au mot dépendant, compte tenu de la présence de son mot principal dans une phrase.

Pour nos expériences, nous avons utilisé un ensemble de données existant sur les préférences de sélection, qui a été annoté par des humains sur les préférences de paires de mots (Zhang et al., 2019b). Les paires de mots sont annotées avec un score de plausibilité moyen, une analogie avec la félicité du mot de tête choisissant un mot comme argument. Nous avons utilisé des paires de mots dépendant de la tête dans cinq relations syntaxiques différentes du corpus SP-10K de préférence sélective, telles que trouvées dans des phrases réelles du corpus ukWaC (Ferraresi et al., 2008). Nous avons calculé la corrélation entre le score de plausibilité et les probabilités attribuées par le modèle pour le mot dépendant, telles qu'elles ont été récupérées par la version de modélisation du langage masqué de bert-base-uncased.

Le coefficient de corrélation entre les jugements humains et les probabilités de BERT n'a pas montré une forte corrélation positive ou négative. BERT attribue des probabilités élevées aux paires de mots et aux contextes fréquemment observés. En outre, les paires de SP-10K ont été créées à partir de mots fréquents, et BERT a tendance à favoriser la prédiction de mots fréquents (par exemple, les adjectifs de taille), parfois à tort. Cependant, nous n'avons pas exigé de BERT une performance semblable à celle des humains, mais nous avons plutôt exploré ses préférences apprises et cherché à savoir si elles coïncidaient avec l'intuition humaine dans une certaine mesure.

L'utilisation des masques d'attention nous a permis d'étudier comment la probabilité du mot cible peut changer, en fonction de la manière dont la séquence d'entrée est traitée par le mécanisme de self-attention. Notre objectif était de déterminer dans quelle mesure le mot de tête influençait la probabilité du mot dépendant et si le contexte était plus significatif que le mot de tête seul. Le fait que les valeurs de corrélation positive les plus fortes résultent presque toujours de la focalisation de l'attention exclusivement sur le mot de tête suggère que le mot de tête est reconnu comme un élément essentiel et significatif de la séquence lorsqu'il s'agit de choisir un mot masqué. Empêcher le mé-

canisme d'attention d'accéder au mot de tête a également montré l'importance du mot de tête pour la probabilité attribuée au mot dépendant.

En comparant les différentes relations syntaxiques du corpus SP-10K, les scores de corrélation les plus faibles proviennent de la catégorie de relation syntaxique **amod**, même si certains noms ont également de fortes préférences lexicales. Cela s'explique par le fait que BERT privilégie les adjectifs à haute fréquence, qui, dans certains cas, peuvent ne pas être très heureux pour les noms. Il est intéressant de noter que les catégories verbe et adjectif (en tant que modificateur du sujet ou de l'objet direct) ont montré, pour la plupart, des corrélations positives similaires aux relations syntaxiques à un saut du verbe et du nom. Comme Zhang et al. (2019b) l'ont également mentionné, ces relations à deux sauts sont également influencées par les préférences de sélection d'un mot. Dans ces cas, le mot de tête est la tête de la séquence et le sujet ou l'objet direct sont ses arguments, de sorte que ses préférences de sélection pourraient avoir eu un impact sur la sélection d'un modificateur dans une plus large mesure que le contexte.

Notre deuxième série d'expériences a porté sur l'aspect lexical. L'aspect lexical est la propriété d'un verbe décrivant les qualités temporelles de l'action, de l'événement ou de l'état du verbe. Contrairement à l'aspect grammatical et au temps du verbe, il s'agit d'une propriété sémantique innée du verbe, qui ne peut changer qu'en présence de significations et de contextes différents. Notre objectif était de découvrir si les plongements lexicaux contextuels peuvent apprendre et encoder des informations sur l'aspect lexical. Nous nous sommes concentrés sur les propriétés de *télicité* (l'existence ou non d'un point final) et de *durée* (la présence d'une action ou d'un état).

Nous avons mené des expériences avec différents ensembles de données Friedrich and Gateva (2017); Alikhani and Stone (2019). Nous avons entraîné les modèles anglais sur une tâche de classification de séquence binaire de télicité ou de durée (*telique-atélique* et *statif-duratif*), en fournissant ou pas la position du verbe. En outre, nous avons réalisé des expériences avec des masques d'attention, des méthodes de visualisation de l'attention et la connaissance des plongements lexicaux pré-entraînés. Nous avons également mené certaines expériences en français, avec des traductions de nos ensembles de données et des modèles français.

Nos modèles affinés sont très performants dans les tâches de classification. Cepen-

dant, nous avons observé l'impact des ensembles de données sur l'affinage. Le premier ensemble de données Friedrich and Gateva contenait des phrases plus longues et plus complexes que l'ensemble de données de Alikhani and Stone (2019). Cela peut expliquer pourquoi les modèles ont eu des difficultés avec certaines phrases longues dans les ensembles de tests qualitatifs, après avoir vu des énoncés plus courts avec une structure moins compliquée.

Un autre résultat intéressant est que les modèles large ont parfois surpassé les modèles base.Pour une tâche complexe telle que l'identification de l'aspect lexical, le traitement et les informations supplémentaires dont disposent les grands modèles ont été bénéfiques pour la précision de leur classification. Cependant, au cours de nos expériences, nous avons également remarqué que le processus d'affinage des grands modèles était parfois un échec et qu'ils ne parvenaient pas à classifier, de sorte que le processus devait être répété.

En outre, les modèles ont bien performé dans les tâches de classification même sans affinage, simplement avec les informations incluses dans l'intégration d'un seul verbe. Ainsi, les plongements contextuels s'avèrent encoder efficacement l'interaction du verbe avec son contexte, sur lequel on s'appuie pour l'aspect lexical du verbe (par exemple, un verbe télique tel que "manger" se trouverait plus fréquemment avec des noms de compte qui établissent le point final de l'action).

De manière surprenante, les résultats de la classification n'ont pas été très affectés par la segmentation des verbes et du contexte en sous-mots par les tokenizers des modèles, par exemple, le tokenizer ALBERT séparant les noms de leur suffixe pluriel. Cela aurait pu être problématique puisque la présence d'un temps pluriel affecte parfois la télicité d'une phrase (Krifka, 1998). Cependant, le modèle pourrait devoir se concentrer sur un plus grand nombre de tokens et pourrait ne pas favoriser certaines parties du contexte si une segmentation supplémentaire séparait les caractéristiques du verbe de la racine. Par conséquent, les modèles dotés de vocabulaires plus restreints, tels qu'ALBERT, pourraient avoir légèrement sous-performé pour cette raison.

L'examen des mécanismes de self-attention des modèles nous a permis d'obtenir un aperçu, mais très limité, de la manière dont les séquences d'entrée étaient traitées par les modèles. Le mécanisme de self-attention de BERT sur les premières couches a démon-

tré une certaine sensibilité à la structure syntaxique et une meilleure "focalisation" sur les tokens individuels dans les premières couches. Cependant, les autres modèles n'ont pas montré de focalisation spécifique sur les constituants dans aucune couche ou tête d'attention. Cela pourrait avoir conduit à leur performance plus faible dans l'ensemble de tests quantitatifss, par rapport aux modèles BERT, en particulier de RoBERTa et ALBERT qui sont des versions optimisées de BERT et ont eu une performance légèrement inférieure à celle des modèles BERT. Les modèles XLNet, malgré l'amélioration des performances de l'architecture sur les dépendances plus longues dans d'autres tâches de TAL, n'ont pas été en mesure de tenir compte du contexte plus efficacement que BERT ou d'encoder des informations plus pertinentes dans leurs encodages.

Notre examen des différents temps de verbe et des positions des phrases prépositionnelles a révélé que les modèles montraient une certaine préférence pour le passé perfectif et le passé continu, par rapport au passé simple. L'ordre des mots n'était pas un indicateur important de réussite ou de confusion, mais le fait de placer un syntagme prépositionnel de temps au début de la phrase (plutôt qu'au milieu ou à la fin) a parfois amélioré les prédictions. Les phrases présentant des contextes conflictuels ont rarement été classées correctement. Cela nous amène à conclure que le plongement du verbe et ses informations sont plus importants pour l'effort de classification du modèle que les autres plongements lexicaux.

Enfin, nos résultats sur les ensembles de données français ont démontré que les choix syntaxiques et sémantiques d'une langue dans la transmission de l'aspect ont influencé la capacité des modèles à catégoriser l'aspect, même avec nos modèles les moins performants. Même avec des architectures de modèles différentes, les disparités entre les erreurs et les succès de classification dans les ensembles de données qualitatives des deux langues montrent que la morphosyntaxe du français peut conduire à des représentations sémantiques différentes par le modèle. Cela est confirmé par le fait que des erreurs se sont produites dans la classification de la télicité dans des phrases anglaises avec la présence de temps simples/continus distincts (qui ne sont pas liés à la télicité mais ont été bénéfiques pour le modèle dans certains cas), alors que leurs traductions françaises ont été correctement classées quelle que soit le temps du verbe.

Enfin, la troisième série d'expériences a exploré les compétences des plongements

lexicaux contextuels avec l'ordre des mots, c.-à-d. la position de l'adjectif attributif dans un syntagme nominal. Alors que des travaux antérieurs ont montré que les modèles Transformers sont insensibles à l'ordre des mots (Pham et al., 2021; Gupta et al., 2021), des modèles affinés ont réussi à classer l'ordre des mots permutés (Sinha et al., 2021b).

Malgré les règles de grammaire traditionnelles qui suggèrent la postposition, la position de l'adjectif attributif dans un syntagme nominal, par rapport à son nom de tête, peut varier de manière significative, en fonction des processus syntaxiques et sémantiques. La position de l'adjectif attributif peut être cruciale pour le sens du syntagme nominal. Alors que l'intuition linguistique suffit aux locuteurs natifs pour prendre ces décisions, notre objectif est d'évaluer si les modèles de Transformers sont capables de comprendre la différence entre les deux positions possibles d'un adjectif dans une séquence.

Nous avons affiné les modèles Transformers français pour qu'ils apprennent la position préférée de l'adjectif attributif dans les phrases nominales, en fournissant les deux positions possibles et en classant celle qui est préférée, puisque les modèles n'ont pas d'informations sur la structure syntaxique correcte d'une phrase nominative. Nous avons également testé avec des lignes de base non informées et traditionnelles et nous avons examiné l'effet des masques d'attention sur la classification (en bloquant l'attention sur le syntagme ou sur le reste du contexte). Nous avons étudié les plongements pré-entraînés avec des prédictions masquées et avec des méthodes de visualisation traditionnelles. Enfin, nous avons eu l'occasion de mener une expérience avec des locuteurs natifs français, afin de comparer les prédictions des modèles à leurs choix dans des cas difficiles de placement d'adjectifs attributifs.

Nos résultats démontrent que les modèles affinés étaient en fait capables de faire la distinction entre l'ordre des mots original et permuté dans la tâche de classification avec une très grande précision. Cependant, dans le cadre des corpus que nous avons examinés, les modèles ont été surpassés par une simple base de référence basée sur la fréquence, et le classificateur CNN s'est également avéré très performant. Cela pourrait indiquer que la tâche en question est assez simple, mais la sensibilité à l'ordre des mots qu'ils ont démontrée ici contredit les multiples résultats des architectures basées sur les Transformers pour les entrées mélangées (parfois au-delà de toute lisibilité). Cependant, les résultats de la deuxième expérience sur les modèles pré-entraînés ont confirmé les

résultats précédents, selon lesquels ces modèles ne dépendent pas de la position des mots.

La mise au point avec un ensemble de données plus large et plus varié a été bénéfique, mais elle a également réussi avec un ensemble de données plus petit, contrairement à nos lignes de base.

En ce qui concerne l'utilisation des masques d'attention, autoriser l'attention sur le syntagme nominal uniquement (adjectif et nom) a affecté les modèles de différentes manières. Les modèles CamemBERT étaient tout à fait capables de classer l'ordre des mots en ne prêtant attention qu'à l'adjectif et au nom, alors que c'était impossible pour les modèles FlauBERT. En revanche, le fait de ne tenir compte que du contexte sans l'adjectif et le nom était relativement inoffensif pour tous les modèles.

Lorsque le mécanisme d'attention des modèles n'a accès qu'au contexte, et non à la paire adjectif-nom elle-même, ils sont toujours capables de classer la position de l'adjectif, même si le mécanisme d'attention n'y a pas accès. Cette observation est cohérente avec l'observation linguistique selon laquelle la position de l'adjectif est également déterminée par le contexte et pas uniquement par le syntagme nominal. Cependant, le fait que les modèles CamemBERT aient extrêmement bien réussi à identifier la position sans utiliser le contexte, alors que les modèles FlauBERT ont complètement échoué, est dû aux différentes architectures des modèles et aux choix dans la façon dont les tokens sont traités. Dans nos expériences plus détaillées, nous avons constaté que les modèles CamemBERT attribuent une probabilité globalement plus élevée aux adjectifs, quelle que soit leur position, et que, au moins pour l'ensemble de données UD, les plongements d'adjectifs étaient, dans certaines couches, très bien informés sur la position préférée du mot. Cette connaissance est corrélée aux plongements lexicaux contextuels appris, plutôt qu'au mot lui-même, car nous avons observé un manque de similarité sémantique dans la visualisation.

Pour la plupart des adjectifs, la prédiction de leur position est une décision relativement facile basée sur la fréquence ; pour observer les compétences sous-jacentes des modèles dans des cas plus complexes, nous avons effectué une analyse d'erreur et des expériences et visualisations supplémentaires sur les versions pré-entraînées des modèles. Les différences entre les deux architectures se sont également reflétées dans notre étude des plongements lexicaux pré-entraînés et des probabilités d'adjectifs, où nous

avons remarqué que les plongements d'adjectifs de CamemBERT étaient mieux informés. En ce qui concerne les plongements d'adjectifs, la façon dont ils sont créés semble mettre davantage l'accent sur le contexte que sur l'information spécifique au mot du token correspondant. L'examen des itérations d'un adjectif dans différentes phrases n'a pas permis de mettre en évidence un modèle de comportement similaire à la similarité vectorielle dans les plongements lexicaux traditionnels. Nous avons remarqué que la classification avec les plongements lexicaux adjectifs n'était réussie que pour certaines couches et certains modèles, mais qu'elle était imprévisible entre nos deux ensembles de données. Nous n'avons pas non plus pu observer de grappes de plongement du même adjectif en ce qui concerne leur position.

Ces résultats suggèrent que les plongements lexicaux contextuels contiennent des informations sur l'ordre préféré des mots, mais seulement dans certaines couches. La mise au point d'un modèle permet d'apprendre ces variations dans la position de l'adjectif et de sélectionner avec succès l'adjectif correct. Dans l'ensemble, les modèles tendent à favoriser les adjectifs fréquents et les informations contextuelles, plutôt que le contenu de l'adjectif (son sens et sa classe). Cependant, la position de l'adjectif dépend de la fréquence, d'où le succès de la base de fréquence non informée et le succès de nos modèles. Les modèles CamemBERT ont mieux réussi que les modèles FlauBERT dans toutes les expériences et ont capturé plus d'informations positionnelles dans les plongements d'adjectifs ajustés. Cependant, tous les modèles de Transformers montrent des faiblesses (à différents degrés) dans les cas complexes de phrases dépendantes d'un adjectif ou d'un nom et d'expressions fixes.

En ce qui concerne les erreurs des modèles, les très rares erreurs commises (par les modèles affinés) étaient justifiables dans une certaine mesure et étaient dues soit à des adjectifs peu fréquents, soit à une mauvaise analyse syntaxique, soit à un sens ambigu qui est grammatical et acceptable dans les deux positions de l'adjectif. En outre, nous avons noté quelques phrases dans notre ensemble de données de questionnaire qui pourraient sembler peu naturelles pour des locuteurs natifs, mais elles ont été soit trouvées dans des corpus existants, soit conçues pour être difficiles à comprendre par un locuteur natif français. Cependant, la comparaison des modèles avec la performance humaine a montré leurs véritables forces et faiblesses ; lorsqu'ils réussissent, les modèles ont ten-

dance à suivre une structure syntaxique plus rigide et à favoriser la postposition, car c'est la position adjectivale la plus fréquente parmi tous les adjectifs. Ils ont montré de sérieuses difficultés à reconnaître certaines expressions figées et ont été plus facilement influencés que les humains lorsqu'ils ont été amorcés avec le même adjectif. Dans les cas où les deux positions étaient possibles, ils préféraient généralement la postposition la plus "traditionnelle". Ces résultats pourraient démontrer que les modèles fondent leurs prédictions davantage sur la fréquence que sur les informations syntaxiques et sémantiques d'un adjectif particulier, et qu'ils sont imperméables aux facteurs qui affectent les décisions des locuteurs, tels que la longueur, la difficulté du traitement en ce qui concerne la charge cognitive, et les différences sémantiques substantielles ou subtiles.

À la question de savoir si les plongements lexicaux contextuels capturent le contexte de manière suffisante et efficace, notre réponse penche vers l'affirmative. Ce n'est pas une nouveauté puisqu'il a été prouvé qu'ils donnaient de meilleurs résultats que les plongements lexicaux statiques. Il est plus intéressant d'examiner *comment* cette information contextuelle est capturée. En examinant les différentes architectures basées sur Transformer qui ont créé ces plongements, nous observons un dénominateur commun : le processus de pré-entraînement nécessite un très grand nombre de données, même pour les modèles pré-entraînés les plus petits. L'architecture neuronale Transformer est capable de traiter cet énorme volume de données de manière dynamique, en extrayant des modèles basés sur le mécanisme de self-attention à plusieurs têtes. L'objectif de modélisation du langage masqué est utilisé par tous les modèles examinés dans cette thèse pour créer leurs plongements pré-entraînés respectifs. En apparence, le modèle est capable d'évaluer lesquels de ces motifs sont les plus appropriés pour la position masquée dans une séquence, mais ce succès est le résultat de la fréquence et non d'une quelconque compétence linguistique.

Les modèles sont linguistiquement agnostiques, c'est-à-dire qu'ils traitent chaque token comme un élément d'information et chaque token masqué comme un token dépourvu de toute caractéristique autre que son contexte. Lors d'expériences préliminaires, nous avons remarqué que la tâche de prédiction en position masquée donnait des résultats irréguliers. À première vue, ils semblent généralement prédire un mot correspondant au contexte à la position masquée. Lorsque l'élément masqué n'était pas un verbe, les

modèles ont montré une affinité pour la prédiction d'éléments fréquents, tels que les adjectifs de taille, les pronoms et la ponctuation. Cependant, dans les cas où le contexte était vague ou complexe, le modèle prédisait des pronoms ou des signes de ponctuation au lieu d'adjectifs ou de noms. Ces expériences n'ont malheureusement pas été fructueuses et n'ont donc pas été poursuivies. Pour les résumer en quelques mots, les choix des modèles pré-entraînés pour une position masquée sont au mieux "le pari le plus sûr" et au pire une platitude fade, un mot absurde ou un terme offensant occasionnel.

Au lieu de cela, nous nous sommes concentrés sur l'utilisation de la prédiction de la langue masquée en injectant le mot initialement masqué dans cette position et en récupérant sa probabilité (voir les sections 4.4.3, 6.6). Les modèles n'ont pas essentiellement échoué dans cette tâche, mais leur comportement diffère considérablement des choix humains. Ils attribuent des probabilités élevées aux mots fréquents, même s'ils ne sont pas les mieux adaptés au contexte donné. Ces éléments se sont vus attribuer une probabilité élevée parce qu'ils ont été fréquemment observés dans de nombreux contextes similaires. De même, les mots moins fréquents mais qui correspondent mieux à la position masquée reçoivent une probabilité plus faible, en raison de leur rareté globale.

La modélisation des plongements repose exclusivement sur des informations contextuelles, ce qui signifie que les différentes occurrences d'un même mot sont traitées et encodées différemment. Les modèles ne sont pas en mesure de créer des classes ou des groupes de mots intégrés, sur la base du contenu des intégrations. Dans la section 6.5.3, nous avons étudié si les propriétés des vecteurs de plongement de mots dynamiques correspondaient à celles des vecteurs de plongement de mots statiques, en traçant différentes instances du même mot - en traçant des lemmes et des types de mots. Intuitivement, nous nous attendions à ce que des grappes apparaissent, sur la base de la similarité des informations contextuelles dans les vecteurs du même mot, mais ce n'était pas le cas. Dans la section 4.5.2, nous n'avons pas pu identifier non plus de modèles discernables de comportement des mots de tête (verbes, noms) choisissant leurs constituants (mots masqués) sur la base de la classe du mot de tête. Nous avons remarqué quelques préférences faibles, comme les verbes qui choisissent leur sujet et leur objet en fonction de l'animalité.

Dans les expériences d'extraction et d'exploration d'plongements lexicaux spéci-

fiques, nous avons également eu l'occasion d'étudier plus avant les plongements lexicaux spécifiques. Dans la section 5.5.4.4, nous avons examiné si les plongements de verbes étaient capables de capturer des informations sur l'aspect lexical, en se basant sur les caractéristiques temporelles du contexte, c'est-à-dire la compagnie préférée du verbe. Cette tâche de classification a été réussie, en particulier pour la classification de la durée. Cependant, notre expérience dans la section 6.5.1 sur la classification de la position de l'adjectif basée sur l'intégration de l'adjectif n'a pas été aussi fructueuse. En termes simples, tous les plongements ne se valent pas. Les plongements de verbes peuvent contenir des informations plus intéressantes, car ils ont tendance à imposer davantage de contraintes au contexte que les autres tokens. En revanche, les adjectifs n'ont pas une telle influence transformatrice sur leur contexte, de sorte que leurs plongements pré-entraînés encodent suffisamment d'informations sur les préférences de l'adjectif.

Cela nous amène à la question suivante, à savoir si **les plongements contextuels de mots montrent une sensibilité à la sémantique**. Nos trois sujets de recherche ont exploré les phénomènes linguistiques qui influencent l'acceptabilité d'une phrase. Dans les expériences sur les préférences de sélection, les modèles pré-entraînés ont montré qu'ils peuvent capturer les informations contextuelles des plongements lexicaux individuels et qu'ils ont une préférence pour la fréquence et pour la félicité sémantique.

Nous ne pouvons pas affirmer que les modèles n'ont pas de compétences syntaxiques; les modèles ont montré qu'ils sont capables de distinguer les parties les plus importantes d'une phrase, c'est-à-dire le verbe de la clause principale, son (ses) verbe(s) auxiliaire(s), son sujet et son (ses) objet(s), et le verbe de la clause subordonnée. Leur perception de la syntaxe n'est pas comparable aux capacités syntaxiques humaines et ne devrait pas non plus être comparée à la cognition humaine. Toutefois, les modèles semblent ne pas "comprendre" les phrases prépositionnelles, les adverbes et les adjectifs attributifs pour traiter une phrase et classer ses propriétés. Ils ont (la plupart du temps) réussi à capturer les contextes littéraux préférés et, dans les cas de métaphores ou de contextes antagonistes, ont préféré "ignorer" ces divergences. Malheureusement, ces éléments sont essentiels à la communication humaine et véhiculent d'importantes informations syntaxiques et sémantiques, qui ont conduit à des prédictions incorrectes lorsqu'elles ont été ignorées.

Une motivation importante pour nos expériences était la corrélation des modèles

avec le comportement humain. Nous ne cherchons pas à soutenir ou à reproduire l'argument selon lequel les modèles peuvent ou doivent imiter la production linguistique humaine. Notre objectif était d'observer les préférences linguistiques humaines, qui ne correspondaient pas nécessairement aux évaluations faites par les linguistes, et de les comparer au comportement appris des modèles. Le terme "perroquet stochastique" inventé par Bender and Koller (2020) nous vient à l'esprit ; bien que nous soyons d'accord sur le fait que les modèles reposent sur la répétition, nous aimerions ajouter l'adjectif "têtu" à cette caractérisation. Les architectures apprennent des schémas syntaxiques rudimentaires et insistent sur leur utilisation même lorsque le contexte suggère le contraire.

Existe-t-il un espoir d'améliorer ces modèles d'intégration contextuelle de mots? En nous concentrant sur le rôle de **finetuning dans les tâches de NLP**, nous avons soutenu tout au long de cette thèse que le finetuning est bénéfique pour créer des embeddings avec des connaissances supplémentaires. Même si nos expériences testaient des phénomènes limités et que nos résultats n'étaient pas toujours des pourcentages impressionnants montrant un succès quantitatif, nous avons démontré que l'apprentissage par transfert peut sensibiliser les modèles aux phénomènes linguistiques, dans une certaine mesure. Le champ d'utilisation possible de ces modèles affinés peut sembler limité, car la communauté se concentre sur des modèles pour des tâches en aval d'une portée plus large. Quoi qu'il en soit, nous espérons que nos conclusions sur l'affinage seront utiles à la communauté du TAL.

En ce qui concerne notre critique technique de l'affinage, nous avons observé qu'il ne s'agit pas, en fait, d'un processus très stable. Des graines aléatoires dans le processus de affinage peuvent perturber la classification, et les modèles de grande taille peuvent produire des résultats imprévisibles. Les modèles de base ont mieux accepté l'affinage, et quelques phases d'affinage ont été suffisantes pour nos expériences. Nos expériences sur l'ordre des mots ont montré que des ensembles de données plus grands et variés ont donné de meilleurs résultats dans l'ensemble. Cependant, nos expériences sur l'aspect lexical ont également réussi avec de petits ensembles de données. En outre, les données de bonne qualité sont importantes pour le succès de l'affinage. La combinaison d'ensembles de données de différents domaines avec le même objectif d'annotation a également été très bénéfique dans toutes nos expériences, car elle a introduit une var-

iété dans les données, rendant les modèles d'affinage plus robustes.

Une partie essentielle de notre recherche a été d'observer le **mécanisme de self-attention**. Dans nos multiples expériences avec les masques d'attention, nous avons observé différents comportements, parfois anticipés, parfois surprenants. Pour les expériences de préférences sélectives, concentrer l'attention des modèles uniquement sur le mot de tête a produit de meilleurs résultats. Les plongements de verbes peuvent contenir une plus grande quantité d'informations que les autres plongements lexicaux, d'où leur plus grande influence. Cependant, dans le cas de la classification de l'aspect lexical, nous avons recréé la tâche de mise au point et de classification avec un masque d'attention sur le contexte et nous avons constaté une baisse de la précision. Bien que cette constatation contredise l'importance des plongements du verbe pour les prédictions, elle peut être due au fait que le processus d'affinage modifie les plongements avec des informations supplémentaires, distribuant ainsi l'information dans la phrase entière.

Sur la question de savoir si l'attention est une mesure solide de l'explicabilité et de la réussite, nous avons trouvé des tendances mais pas de réponses concrètes. Tout d'abord, le seul modèle qui a produit des résultats visuellement intéressants est BERT, comparé à RoBERTa et ALBERT qui ont montré des poids d'attention diffus sans préférences marquées. Certaines couches et têtes particulières ont produit des visualisations d'attention constamment similaires indépendamment de la phrase d'entrée, par exemple la troisième couche de bert-base-uncased produisant une attention monotone du token t au token $t + 1$. Nous avons remarqué que les tendances les plus fortes de l'attention se produisaient entre les tokens de verbe (mot principal ou de la clause subordonnée) et les sujets, les objets et les verbes auxiliaires. Pour corroborer la bibliographie, nous avons également observé que certaines couches des architectures présentent des modèles d'attention qui s'apparentent à des relations syntaxiques. En conclusion, bien que la self-attention des têtes multiples soit difficile à déchiffrer et à étudier, elle peut constituer un outil d'observation intéressant du fonctionnement interne des modèles et de l'ordre des opérations lorsqu'ils traitent le langage et produisent des résultats.

Enfin, en ce qui concerne la question de l'importance de l'ordre des mots, si les modèles étaient réellement agnostiques à l'égard de l'ordre des mots, ils traiteraient les entrées avec des tokens mélangés de la même manière. Cela n'a été le cas ni dans nos

expériences sur l'ordre des mots, ni dans les expériences de classification des aspects lexicaux dans lesquelles l'ordre des mots n'était pas un objectif de l'affinage. Les modèles affinés ont été capables de classer l'ordre des mots de l'adjectif préféré avec succès dans les ensembles de données de classification, dans la Section 6.4. Bien que la tâche ait pu être assez simple, puisqu'une métrique basée sur la fréquence a également donné de bons résultats, elle aurait été impossible pour les modèles s'ils avaient été complètement agnostiques par rapport à l'ordre des mots. En outre, ils ont traité les phrases grammaticales et acceptables des mêmes tokens de manière différente, en classant leurs qualités temporelles différemment. Alors que les modèles pré-entraînés peuvent être insensibles aux permutations, les modèles finement réglés peuvent être rendus sensibles aux permutations.

Bibliography

Abdou, M., Kulmizev, A., Hill, F., Low, D. M., and Søgaard, A. (2019). Higher-order comparisons of sentence encoder representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5838–5845, Hong Kong, China. Association for Computational Linguistics.

Abdou, M., Ravishankar, V., Kulmizev, A., and Søgaard, A. (2022). Word order does matter and shuffled language models know it. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6907–6919.

Abeillé, A. and Godard, D. (1999). La position de l'adjectif épithète en français: le poids des mots. *Recherches linguistiques de Vincennes*, (28):9–32.

Abnar, S. and Zuidema, W. (2020). Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*.

Adaloglou, N. (2021). Why multi-head self attention works: Math, intuitions and 10+1 hidden insights. https://theaisummer.com/self-attention/#an-intuitive-illustration (Online; accessed 22.09.2022).

Al-Rfou', R., Perozzi, B., and Skiena, S. (2013). Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Alammar, J. (2018). The illustrated transformer. http://jalammar.github.io/illustrated-transformer/ (Online; accessed 22.09.2022).

Alikhani, M., Kober, T., Alhafni, B., Chen, Y., Inan, M., Nielsen, E., Raji, S., Steedman, M., and Stone, M. (2022). Zero-shot cross-linguistic learning of event semantics. *arXiv preprint arXiv:2207.02356*.

Alikhani, M. and Stone, M. (2019). "Caption" as a Coherence Relation: Evidence and

Implications. In *Proceedings of the Second Workshop on Shortcomings in Vision and Language*, pages 58–67.

Alishahi, A. and Stevenson, S. (2007). A cognitive model for the representation and acquisition of verb selectional preferences. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 41–48.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bard, E. G., Robertson, D., and Sorace, A. (1996). Magnitude estimation of linguistic acceptability. *Language*, pages 32–68.

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Bastings, J. and Filippova, K. (2020). The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155, Online. Association for Computational Linguistics.

Becker, C. (2020). Transfer learning for nlp i. https://slds-lmu.github.io/seminar_nlp_ss20/transfer-learning-for-nlp-i.html (Online; accessed 06.11.2022).

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Bender, E. M. and Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Bengio, Y., Ducharme, R., and Vincent, P. (2000). A neural probabilistic language model. *Advances in neural information processing systems*, 13.

Benzitoun, C. (2013). Adjectifs épithètes alternants en français parlé: premiers résultats. *TIPA. Travaux interdisciplinaires sur la parole et le langage*, (29).

Benzitoun, C. (2014). La place de l'adjectif épithète en français: ce que nous apprennent les corpus oraux. In *SHS Web of Conferences*, volume 8, pages 2333–2348. EDP Sciences.

Bergsma, S., Lin, D., and Goebel, R. (2008). Discriminative learning of selectional preference from unlabeled text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 59–68. Association for Computational Linguistics.

Bibal, A., Cardon, R., Alfter, D., Wilkens, R., Wang, X., François, T., and Watrin, P. (2022). Is attention explanation? an introduction to the debate. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3889–3900.

Bloem, P. (2019). Transformers from scratch. https://peterbloem.nl/blog/transformers (Online; accessed 25.03.2023).

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Brandl, S., Eberle, O., Pilot, J., and Søgaard, A. (2022). Do transformer models show similar attention patterns to task-specific human gaze? *arXiv preprint arXiv:2205.10226*.

Branigan, H. P., Pickering, M. J., Liversedge, S. P., Stewart, A. J., and Urbach, T. P. (1995). Syntactic priming: Investigating the mental representation of language. *Journal of Psycholinguistic Research*, 24(6):489–506.

Brown, P. F., Della Pietra, V. J., Desouza, P. V., Lai, J. C., and Mercer, R. L. (1992). Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–480.

Brunner, G., Liu, Y., Pascual, D., Richter, O., Ciaramita, M., and Wattenhofer, R. (2019). On identifiability in transformers. *arXiv preprint arXiv:1908.04211*.

Chambers, N., Cassidy, T., McDowell, B., and Bethard, S. (2014). Dense Event Ordering with a Multi-Pass Architecture. In *Transactions of the Association for Computational Linguistics*, volume 2, pages 273–284.

Chang, T., Xu, Y., Xu, W., and Tu, Z. (2021). Convolutions and self-attention: Re-interpreting relative positions in pre-trained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4322–4333, Online. Association for Computational Linguistics.

Chang, T. A. and Bergen, B. K. (2021). Word acquisition in neural language models. *arXiv preprint arXiv:2110.02406*.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, T. (2013). One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Cheng, J., Dong, L., and Lapata, M. (2016). Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

Chomsky, N. (1957). Syntactic structures. In *Syntactic Structures*. De Gruyter Mouton.

Chrupała, G. and Alishahi, A. (2019). Correlating neural and symbolic representations of language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2952–2962, Florence, Italy. Association for Computational Linguistics.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019). What Does BERT Look at? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Clark, S. and Weir, D. (2001). Class-based probability estimation using a semantic hierarchy. In *Proceedings of the second meeting of the North American Chapter of the*

Association for Computational Linguistics on Language technologies, pages 95–102. Association for Computational Linguistics.

Coenen, A., Reif, E., Yuan, A., Wattenberg, M., Viegas, F. B., Pearce, A., and Kim, B. (2019). Visualizing and measuring the geometry of BERT. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8594–8603. Curran Associates, Inc.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Comrie, B. (1989). *Language universals and linguistic typology: Syntax and morphology*. University of Chicago press.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. (2018). What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Conneau, A. and Lample, G. (2019). Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

Costa, F. and Branco, A. (2012). Aspectual Type and Temporal Relation Classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 266–275, Avignon, France. Association for Computational Linguistics.

Cristina, S. (2022a). The transformer attention mechanism. https://machinelearningmastery.com/the-transformer-attention-mechanism/ (Online; accessed 01.03.2023).

Cristina, S. (2022b). The transformer model. https://machinelearningmastery.com/the-transformer-model/ (Online; accessed 01.03.2023).

Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q., and Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., and Smith, N. (2020). Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Dowty, D. R. (1979). *Word Meaning and Montague Grammar: The Semantics of Verbs and Times in Generative Semantics and in Montague's Ptq*, volume 7. Springer.

Erk, K. (2007). A simple, similarity-based model for selectional preferences. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 216–223, Prague, Czech Republic. Association for Computational Linguistics.

Erk, K., Padó, S., and Padó, U. (2010). A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, 36(4):723–763.

Ettinger, A. (2020). What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Ettinger, A., Elgohary, A., and Resnik, P. (2016). Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st workshop on evaluating vector-space representations for nlp*, pages 134–139.

Falk, I. and Martin, F. (2016). Automatic identification of aspectual classes across verbal readings. In ** Sem 2016 THE FIFTH JOINT CONFERENCE ON LEXICAL AND COMPUTATIONAL SEMANTICS*.

Ferraresi, A., Zanchetta, E., Baroni, M., and Bernardini, S. (2008). Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54.

Ferretti, T. R., McRae, K., and Hatherell, A. (2001). Integrating verbs, situation schemas, and thematic role concepts. *Journal of Memory and Language*, 44(4):516–547.

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Forsgren, M. (1978). *La place de l'adjectif épithète en français contemporain: étude quantitative et sémantique*. PhD thesis, Acta Universitatis Upsaliensis.

Forsgren, M. (2016). La place de l'adjectif épithète. *Encyclopédie grammaticale du français (online)*. Accessed on Jan 04, 2022.

Friedrich, A. and Gateva, D. (2017). Classification of telicity using cross-linguistic annotation projection. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2559–2565.

Friedrich, A. and Palmer, A. (2014). Automatic prediction of aspectual class of verbs in context. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 517–523.

Friedrich, A., Palmer, A., and Pinkal, M. (2016). Situation entity types: automatic classification of clause-level aspect. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1757–1768.

Friedrich, A. and Pinkal, M. (2015). Automatic recognition of habituals: a three-way classification of clausal aspect. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2471–2481.

Fromkin, V., Rodman, R., and Hyams, N. (2013). *An introduction to language*. Cengage Learning.

Gage, P. (1994). A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Galassi, A., Lippi, M., and Torroni, P. (2020). Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(10):4291–4308.

Gallant, S. I. (1991). A practical approach for representing context and for performing word sense disambiguation using neural networks. *Neural Computation*, 3(3):293–309.

Garey, H. B. (1957). Verbal aspect in french. *Language*, 33(2):91–110.

Gildea, D. and Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288.

Giulianelli, M., Harding, J., Mohnert, F., Hupkes, D., and Zuidema, W. (2018). Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.

Goes, J. (1999). *L'adjectif: entre nom et verbe*, volume 777. De Boeck Supérieur.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

Goldberg, Y. (2019). Assessing BERT's Syntactic Abilities. *arXiv preprint arXiv:1901.05287*.

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Greenbaum, S. (1977). *Acceptability in language*. Mouton The Hague.

Greenberg, C., Demberg, V., and Sayeed, A. (2015). Verb polysemy and frequency effects in thematic fit modeling. In *Proceedings of the 6th Workshop on Cognitive Modeling and Computational Linguistics*, pages 48–57.

Gross, G. (1996). *Les expressions figées en français: noms composés et autres locutions*. Editions Ophrys.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42.

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., and Baroni, M. (2018). Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Gupta, A., Kvernadze, G., and Srikumar, V. (2021). Bert & family eat word salad: Experiments with text understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12946–12954.

Hagiwara, M. (2021). *Real-World Natural Language Processing: Practical Applications with Deep Learning*. Simon and Schuster.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

He, B. (2020). Transfer learning for nlp ii. https://slds-lmu.github.io/seminar_nlp_ss20/transfer-learning-for-nlp-ii.html (Online; accessed 06.11.2022).

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

He, P., Liu, X., Gao, J., and Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Hessel, J. and Schofield, A. (2021). How effective is bert without word ordering? implications for language understanding and data privacy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 204–211.

Hewitt, J. and Liang, P. (2019). Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python. Technical report, Zenodo.

Hoover, B., Strobelt, H., and Gehrmann, S. (2020). exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 187–196, Online. Association for Computational Linguistics.

Horev, R. (2018). Bert explained: State of the art language model for nlp. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270 (Online; accessed 10.11.2022).

Hosseini, M. J., Chambers, N., Reddy, S., Holt, X. R., Cohen, S. B., Johnson, M., and Steedman, M. (2018). Learning Typed Entailment Graphs with Global Soft Constraints. In *Transactions of the Association for Computational Linguistics*, volume 6, pages 703–717.

Hovav, M. R. and Levin, B. (2010). Reflections on manner/result complementarity. *Syntax, lexical semantics, and event structure*, pages 21–38.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Hupkes, D., Veldhoen, S., and Zuidema, W. (2018). Visualisation and 'Diagnostic Classifiers' Reveal how Recurrent and Recursive Neural Networks Process Hierarchical Structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Ide, N., Baker, C., Fellbaum, C., Fillmore, C., and Passonneau, R. (2008). MASC: the Manually Annotated Sub-Corpus of American English. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259.

Jacovi, A. and Goldberg, Y. (2020). Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Jain, S. and Wallace, B. C. (2019). Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556.

Jawahar, G., Sagot, B., and Seddah, D. (2019). What Does BERT Learn about the Structure of Language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657.

Jiang, Z., Xu, F. F., Araki, J., and Neubig, G. (2020). How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020). Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. (2016). Exploring the limits of language modeling. Technical report, Google Inc.

Kanerva, P., Kristoferson, J., and Holst, A. (2000). Random indexing of text samples for

latent semantic analysis. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 22(22).

Karim, R. (2019). Attn: Illustrated attention. https://towardsdatascience.com/attn-illustrated-attention-5ec4ad276ee3 (Online; accessed 22.09.2022).

Karim, R. (2022). Illustrated: Self-attention. https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a (Online; accessed 22.09.2022).

Katz, J. J. and Fodor, J. A. (1963). The structure of a semantic theory. *Language*, 39(2):170–210.

Kearns, K. (1991). *The Semantics of the English Progressive*. PhD thesis, MIT.

Kearns, K. (2000). *Semantics*. Palgrave Macmillan, Basingstoke.

Kim, N., Patel, R., Poliak, A., Xia, P., Wang, A., McCoy, T., Tenney, I., Ross, A., Linzen, T., Van Durme, B., et al. (2019). Probing what different nlp tasks teach machines about function word comprehension. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (* SEM 2019)*, pages 235–249.

Kim, S. J., Yu, L., and Ettinger, A. (2022). "no, they did not": Dialogue response dynamics in pre-trained language models. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 863–874, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Kobayashi, G., Kuribayashi, T., Yokoi, S., and Inui, K. (2020). Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Kober, T., Alikhani, M., Stone, M., and Steedman, M. (2020). Aspectuality Across Genre: A Distributional Semantics Approach. In *Proceedings of the 28th International Con-*

ference on Computational Linguistics, pages 4546–4562, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kober, T., Bijl de Vroe, S., and Steedman, M. (2019). Temporal and Aspectual Entailment. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 103–119, Gothenburg, Sweden. Association for Computational Linguistics.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *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. Association for Computational Linguistics.

Krifka, M. (1998). The origins of telicity. In *Events and grammar*, pages 197–235. Springer.

Kudo, T. and Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Laporte, É. (2004). Acceptability as the source of syntactic knowledge. *Applied Linguistics (ISSN 1003-5397)*, pages 9–22.

Larsson, B. (1994). *La place et le sens des adjectifs épithètes de valorisation positive: Une étude de 113 adjectifs d'emploi fréquent dans la langue du tourisme et dans d'autres types de prose non-littéraire*. Lund University Press.

Lasri, K., Lenci, A., and Poibeau, T. (2022). Does bert really agree? fine-grained analysis of lexical dependence on a syntactic task. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2309–2315.

Laurent, N. and Delaunay, B. (2013). *Bescherelle La grammaire pour tous: Ouvrage de référence sur la grammaire française*. Hatier.

Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L., and Schwab, D. (2020). Flaubert: Unsupervised language model pre-training for french. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2479–2490.

Le, M. and Fokkens, A. (2018). Neural models of selectional preferences for implicit semantic role labeling. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Lebret, R. and Collobert, R. (2014). Word embeddings through hellinger PCA. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–490, Gothenburg, Sweden. Association for Computational Linguistics.

Leivada, E. and Westergaard, M. (2020). Acceptable ungrammatical sentences, unacceptable grammatical sentences, and the role of the cognitive parser. *Frontiers in Psychology*, page 364.

Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Li, B., Wisniewski, G., and Crabbé, B. (2021). Are transformers a modern version of eliza? observations on french object verb agreement. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4599–4610.

Li, H. and Abe, N. (1998). Generalizing case frames using a thesaurus and the MDL principle. *Computational linguistics*, 24(2):217–244.

Li, Y., Xu, L., Tian, F., Jiang, L., Zhong, X., and Chen, E. (2015). Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Libovický, J. h. (2022). Why are residual connections needed in transformer architectures? Cross Validated. :https://stats.stackexchange.com/q/565203(version: 2022-02-21) (Online; accessed 01.03.2023).

Light, M. and Greiff, W. (2002). Statistical models for the induction and use of selectional preferences. *Cognitive Science*, 26(3):269–281.

Lin, D., Zhao, S., Qin, L., and Zhou, M. (2003). Identifying synonyms among distributionally similar words. In *IJCAI*, volume 3, pages 1492–1493.

Lindsay, G. W. (2020). Attention in psychology, neuroscience, and machine learning. *Frontiers in computational neuroscience*, 14:29.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loáiciga, S. and Grisot, C. (2016). Predicting and Using a Pragmatic Component of Lexical Aspect of Simple Past Verbal Tenses for Improving english-to-french Machine Translation. In *Linguistic Issues in Language Technology, Volume 13, 2016*. CSLI Publications.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Lynn, S. (2018). An introduction to word embeddings for text analysis. https://www.shanelynn.ie/get-busy-with-word-embeddings-introduction (Online; accessed 26.10.2022).

Mackenzie, J., Benham, R., Petri, M., Trippas, J. R., Culpepper, J. S., and Moffat, A. (2020). Cc-news-en: A large english news corpus. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3077–3084.

Martin, L., Muller, B., Ortiz Suárez, P. J., Dupont, Y., Romary, L., de la Clergerie, É., Seddah, D., and Sagot, B. (2020). CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

McCarthy, D. and Carroll, J. (2003). Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Computational Linguistics*, 29(4):639–654.

McClure, W. (1994). *Syntactic Projections of the Semantics of Aspect*. PhD thesis, Cornell University.

McCormick, C. (2016). Word2vec tutorial-the skip-gram model. http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model (Online; accessed 26.10.2022).

McCormick, C. and Ryan, N. (2019). BERT Fine-Tuning Tutorial with PyTorch. Retrieved January 24, 2021.

McCoy, T., Pavlick, E., and Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

McRae, K., Spivey-Knowlton, M. J., and Tanenhaus, M. K. (1998). Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension. *Journal of Memory and Language*, 38(3):283–312.

Měchura, M. (2008). *Selectional Preferences, Corpora and Ontologies*. PhD thesis, Ph. D. thesis, Trinity College, University of Dublin.

Merchant, A., Rahimtoroghi, E., Pavlick, E., and Tenney, I. (2020). What happens to bert embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44.

Michel, P., Levy, O., and Neubig, G. (2019). Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

Mickus, T., Paperno, D., Constant, M., and van Deemter, K. (2020). What do you mean, BERT? Assessing BERT as a Distributional Semantics Model. *Proceedings of the Society for Computation in Linguistics*, 3(1):350–361.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013c). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Mnih, A. and Hinton, G. E. (2008). A scalable hierarchical distributed language model. *Advances in neural information processing systems*, 21.

Montalbetti, M. M. (1984). *After binding: On the interpretation of pronouns*. PhD thesis, Massachusetts Institute of Technology.

Morin, F. and Bengio, Y. (2005). Hierarchical probabilistic neural network language model. In *International workshop on artificial intelligence and statistics*, pages 246–252. PMLR.

Mosbach, M., Andriushchenko, M., and Klakow, D. (2020). On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines. *arXiv preprint arXiv:2006.04884*.

Nivre, J., Hall, J., and Nilsson, J. (2006). Maltparser: A data-driven parser-generator for dependency parsing. In *LREC*, volume 6, pages 2216–2219.

Ó Séaghdha, D. (2010). Latent variable models of selectional preference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 435–444. Association for Computational Linguistics.

Ó Séaghdha, D. and Korhonen, A. (2012). Modelling selectional preferences in a lexical hierarchy. In *Proceedings of the First Joint Conference on Lexical and Computational*

Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, pages 170–179. Association for Computational Linguistics.

Olsen, M. B. (1994). The semantics and pragmatics of lexical aspect features. *Studies in the Linguistic Science*, 24(1-2):361–375.

Olsen, M. B. (1997). *A Semantic and Pragmatic Model of Lexical and Grammatical Aspect*. Garland, New York.

Ortiz Suárez, P. J., Sagot, B., and Romary, L. (2019). Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In Bański, P., Barbaresi, A., Biber, H., Breiteneder, E., Clematide, S., Kupietz, M., Lüngen, H., and Iliadi, C., editors, *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*, Cardiff, United Kingdom. Leibniz-Institut für Deutsche Sprache.

Osgood, C. E., Suci, G. J., and Tannenbaum, P. H. (1957). *The measurement of meaning*. Number 47. University of Illinois press.

O'Connor, J. and Andreas, J. (2021). What context features can transformer language models use? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 851–864.

Padó, U. (2007). The integration of syntax and semantic plausibility in a wide-coverage model of human sentence processing. *Doctoral thesis*.

Pantel, P. and Lin, D. (2002). Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619.

Papadimitriou, I., Futrell, R., and Mahowald, K. (2022). When classifying grammatical role, BERT doesn't care about word order... except when it matters. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 636–643, Dublin, Ireland. Association for Computational Linguistics.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Peck, J., Lin, J., and Sun, C. (2013). Aspectual classification of mandarin chinese verbs: A perspective of scale structure. *Language and Linguistics*, 14(4):663.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peng, Q. (2018). *Towards aspectual classification of clauses in a large single-domain corpus*. School of Informatics, University of Edinburgh, Edingburgh, UK.

Pennington, J. (2014). Glove: Global vectors for word representation. https://nlp.stanford.edu/projects/glove/ (Online; accessed 26.10.2022).

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Pereira, F., Tishby, N., and Lee, L. (1993). Distributional clustering of English words. In *31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Columbus, Ohio, USA. Association for Computational Linguistics.

Peters, M. E., Ammar, W., Bhagavatula, C., and Power, R. (2017). Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), pages 1756–1765, Vancouver, Canada. Association for Computational Linguistics.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Peters, M. E., Ruder, S., and Smith, N. A. (2019). To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., and Miller, A. (2019). Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473.

Pham, T., Bui, T., Mai, L., and Nguyen, A. (2021). Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1145–1160.

Qureshi, M. A. and Greene, D. (2019). Eve: explainable vector based embedding technique using wikipedia. *Journal of Intelligent Information Systems*, 53(1):137–165.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training. Technical report, OpenAI.

Raganato, A. and Tiedemann, J. (2018). An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. The Association for Computational Linguistics.

Ras, G., Xie, N., van Gerven, M., and Doran, D. (2022). Explainable deep learning: A field guide for the uninitiated. *Journal of Artificial Intelligence Research*, 73:329–397.

Ravishankar, V., Kulmizev, A., Abdou, M., Søgaard, A., and Nivre, J. (2021). Attention can reflect syntactic structure (if you let it). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3031–3045.

Reisinger, J. and Mooney, R. (2010). Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117.

Resnik, P. (1993). Semantic classes and syntactic ambiguity. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Resnik, P. (1996). Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61:127–159.

Ritter, A., Mausam, and Etzioni, O. (2010). A latent dirichlet allocation method for selectional preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 424–434, Uppsala, Sweden. Association for Computational Linguistics.

Rodriguez, P., Gonfaus, J. M., Cucurull, G., XavierRoca, F., and Gonzalez, J. (2018). Attend and rectify: a gated attention mechanism for fine-grained recovery. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2021). A Primer in BERTology: What we know about how BERT works. In *Transactions of the Association for Computational Linguistics*, volume 8, pages 842–866. MIT Press.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Rooth, M., Riezler, S., Prescher, D., Carroll, G., and Beil, F. (1999). Inducing a semantically annotated lexicon via EM-based clustering. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 104–111. Association for Computational Linguistics.

Ruder, S. (2019). *Neural transfer learning for natural language processing*. PhD thesis, NUI Galway.

Saeed, M. (2022). A gentle introduction to positional encoding in transformer models, part 1. https://machinelearningmastery.com/a-gentle-introduction-to-positional-encoding-in-transformer-models-part-1/ (Online; accessed 01.03.2023).

Sahlgren, M. (2002). Random indexing of words in narrow context windows for vector-based semantic analysis. *Random Indexing of Words in Narrow Context Windows for Vector-Based Semantic Analysis*, pages 1000–1022.

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Schütze, C. (2016). *The empirical base of linguistics: Grammaticality judgments and linguistic methodology*. Language Science Press.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Serrano, S. and Smith, N. A. (2019). Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Ševčíková, M., Panevová, J., and Pognan, P. (2017). Inflectional and derivational paradigm of verbs in czech: the role of the category of aspect. In *First Workshop on Paradigmatic Word Formation Modeling*.

Shen, Y. and Huang, X.-J. (2016). Attention-based convolutional neural network for semantic relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2526–2536.

Shi, X., Padhi, I., and Knight, K. (2016). Does string-based neural mt learn source syntax? In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1526–1534.

Siegel, E. V. (1998). *Linguistic Indicators for Language Understanding: Using machine learning methods to combine corpus-based indicators for aspectual classification of clauses*. Columbia University. Ph.D. thesis.

Siegel, E. V. and McKeown, K. R. (2000). Learning Methods to Combine Linguistic Indicators:Improving Aspectual Classification and Revealing Linguistic Insights. In *Computational Linguistics*, volume 26, pages 595–627.

Šimandl, J., Rusínová, Z., and Petkevič, V. (2016). *Slovník afixů užívanỳch v češtině*. Univerzita Karlova, nakladatelství Karolinum.

Sinha, K., Jia, R., Hupkes, D., Pineau, J., Williams, A., and Kiela, D. (2021a). Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913.

Sinha, K., Parthasarathi, P., Pineau, J., and Williams, A. (2021b). UnNatural Language Inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7329–7346, Online. Association for Computational Linguistics.

Smith, C. S. (1997). *The Parameter of Aspect (2nd edition)*. Kluwer, Dordrecht.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. (2022). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Strobelt, H., Hoover, B., Satyanaryan, A., and Gehrmann, S. (2021). LMdiff: A visual diff tool to compare language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 96–105, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Subudhi, K. (2019). Bert attention visualization. https://krishansubudhi.github.io/deeplearning/2019/09/26/BertAttention.html (Online; accessed 15.01.2023).

Sukhbaatar, S., Weston, J., Fergus, R., et al. (2015). End-to-end memory networks. *Advances in neural information processing systems*, 28.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Van Durme, B., Bowman, S. R., Das, D., et al. (2019). What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

Thuilier, J. (2012). *Contraintes préférentielles et ordre des mots en français*. Phd thesis, Université Paris-Diderot - Paris VII.

Thuilier, J. (2013). Syntaxe du français parlé vs. écrit: le cas de la position de l'adjectif épithète par rapport au nom. *TIPA. Travaux interdisciplinaires sur la parole et le langage*, (29).

Van de Cruys, T. (2009). A non-negative tensor factorization model for selectional preference induction. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 83–90, Athens, Greece. Association for Computational Linguistics.

Van de Cruys, T. (2014). A neural network approach to selectional preference acquisition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 26–35, Doha, Qatar. Association for Computational Linguistics.

Vashishth, S., Upadhyay, S., Tomar, G. S., and Faruqui, M. (2020). Attention interpretability across NLP tasks. *OpenReview*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vendler, Z. (1967). Verbs and times. *Linguistics in Philosophy*, pages 97–121.

Vig, J. (2019). A Multiscale Visualization of Attention in the Transformer Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42.

Vig, J. and Belinkov, Y. (2019). Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*.

Vinokourov, A., Cristianini, N., and Shawe-Taylor, J. (2002). Inferring a semantic representation of text via cross-language correlation analysis. *Advances in neural information processing systems*, 15.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wang, B., Shang, L., Lioma, C., Jiang, X., Yang, H., Liu, Q., and Simonsen, J. G. (2020a). On position embeddings in bert. In *International Conference on Learning Representations*.

Wang, B., Zhao, D., Lioma, C., Li, Q., Zhang, P., and Simonsen, J. G. (2019a). Encoding word order in complex embeddings. In *International Conference on Learning Representations*.

Wang, X., Tu, Z., Wang, L., and Shi, S. (2019b). Self-attention with structural position representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1403–1409.

Wang, Y.-A. and Chen, Y.-N. (2020). What do position embeddings learn? an empirical study of pre-trained language model positional encoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6840–6849.

Wang, Z., K, K., Mayhew, S., and Roth, D. (2020b). Extending multilingual BERT to low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, Online. Association for Computational Linguistics.

Warstadt, A., Singh, A., and Bowman, S. R. (2018). Neural network acceptability judgments. *CoRR*, abs/1805.12471.

Weissweiler, L., Hofmann, V., Köksal, A., and Schütze, H. (2022). The better your syntax, the better your semantics? probing pretrained language models for the english comparative correlative. *arXiv preprint arXiv:2210.13181*.

Wiegreffe, S. and Pinter, Y. (2019). Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20.

Wilmet, M. (1980). Antéposition et postposition de l'épithète qualificative en français contemporain. *Travaux de linguistique*, 7:179–201.

Wilmet, M. (1981). La place de l'épithète qualificative en français contemporain. etude grammaticale et stylistique. *Revue de Linguistique Romane Lyon*, (177-178):17–73.

Wolf, T. (2019). Some additional experiments extending the tech report "Assessing BERT's syntactic abilities" by Yoav Goldberg. Technical report, Huggingface Inc.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online.

Wright, L. (2019). Meet ALBERT: a new 'Lite BERT' from Google & Toyota with State of the Art NLP performance and 18x fewer parameters. https://lessw.medium.com/meet-albert-a-new-lite-bert-from-google-toyota-with-state-of-the-art-nlp-performance-and-18x-df8f7b58fa28) (Online; accessed 24.01.2021).

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR.

Yang, B., Wang, L., Wong, D. F., Chao, L. S., and Tu, Z. (2019a). Assessing the ability of self-attention networks to learn word order. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3635–3644.

Yang, X. (2020). An overview of the attention mechanisms in computer vision. *Journal of Physics: Conference Series*, 1693(1):012173.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019b). XLNet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32:5753–5763.

Yang, Z. and Le, Q. (2019). Transformer-xl: Unleashing the potential of attention models. https://ai.googleblog.com/2019/01/transformer-xl-unleashing-potential-of.html (Online; accessed 20.11.2022).

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27.

Yu, L. and Ettinger, A. (2020). Assessing phrasal representation and composition in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4896–4907, Online. Association for Computational Linguistics.

Zapirain, B., Agirre, E., Marquez, L., and Surdeanu, M. (2013). Selectional preferences for semantic role classification. *Computational Linguistics*, 39(3):631–663.

Zeman, D., Nivre, J., Abrams, M., Ackermann, E., Aepli, N., Aghaei, H., Agić, Ž., Ahmadi, A., Ahrenberg, L., Ajede, C. K., Aleksandravičiūtė, G., Alfina, I., Antonsen, L., Aplonova, K., Aquino, A., Aragon, C., Aranzabe, M. J., Arıcan, B. N., Arnardóttir, H͡., Arutie, G., Arwidarasti, J. N., Asahara, M., Aslan, D. B., Ateyah, L., Atmaca, F., Attia, M., Atutxa, A., Augustinus, L., Badmaeva, E., Balasubramani, K., Ballesteros, M., Banerjee, E., Bank, S., Barbu Mititelu, V., Barkarson, S., Basile, R., Basmov, V., Batchelor, C., Bauer, J., Bedir, S. T., Bengoetxea, K., Berk, G., Berzak, Y., Bhat, I. A., Bhat, R. A., Biagetti, E., Bick, E., Bielinskienė, A., Bjarnadóttir, K., Blokland, R., Bobicev, V., Boizou, L., Borges Völker, E., Börstell, C., Bosco, C., Bouma, G., Bowman, S., Boyd, A., Braggaar, A., Brokaitė, K., Burchardt, A., Candito, M., Caron, B., Caron, G., Cassidy, L., Cavalcanti, T., Cebiroğlu Eryiğit, G., Cecchini, F. M., Celano, G. G. A., Čéplö, S., Cesur, N., Cetin, S., Çetinoğlu, Ö., Chalub, F., Chauhan, S., Chi, E., Chika, T., Cho, Y., Choi, J., Chun, J., Chung, J., Cignarella, A. T., Cinková, S., Collomb, A., Çöltekin, Ç., Connor, M., Courtin, M., Cristescu, M., Daniel, P., Davidson, E., de Marneffe, M.-C., de Paiva, V., Derin, M. O., de Souza, E., Diaz de Ilarraza, A., Dickerson, C., Dinakaramani, A., Di Nuovo, E., Dione, B., Dirix, P., Dobrovoljc, K., Dozat, T., Droganova, K., Dwivedi, P., Eckhoff, H., Eiche, S., Eli, M., Elkahky, A., Ephrem, B., Erina, O., Erjavec, T., Etienne, A., Evelyn, W., Facundes, S., Farkas, R., Ferdaousi, J., Fernanda, M., Fernandez Alcalde, H., Foster, J., Freitas, C., Fujita, K., Gajdošová, K., Galbraith, D., Garcia, M., Gärdenfors, M., Garza, S., Gerardi, F. F., Gerdes, K., Ginter, F., Godoy, G., Goenaga, I., Gojenola, K., Gökırmak, M., Goldberg, Y., Gómez Guinovart, X., González Saavedra, B., Griciūtė, B., Grioni, M., Grobol, L., Grūzītis, N., Guillaume, B., Guillot-Barbance, C., Güngör, T., Habash, N., Hafsteinsson, H., Hajič, J., Hajič jr., J., Hämäläinen, M., Hà Mỹ, L., Han, N.-R., Hanifmuti, M. Y., Hardwick, S., Harris, K., Haug, D., Heinecke, J., Hellwig, O., Hennig, F., Hladká, B., Hlaváčová, J., Hociung, F., Hohle, P., Huber, E., Hwang, J., Ikeda, T., Ingason, A. K., Ion, R., Irimia, E., Ishola, Ọ., Ito, K., Jannat, S., Jelínek, T., Jha, A., Johannsen, A., Jónsdóttir, H., Jørgensen, F., Juutinen, M., K, S., Kaşıkara, H., Kaasen, A., Kabaeva, N., Kahane, S., Kanayama, H., Kanerva,

J., Kara, N., Katz, B., Kayadelen, T., Kenney, J., Kettnerová, V., Kirchner, J., Klementieva, E., Klyachko, E., Köhn, A., Köksal, A., Kopacewicz, K., Korkiakangas, T., Köse, M., Kotsyba, N., Kovalevskaitė, J., Krek, S., Krishnamurthy, P., Kübler, S., Kuyrukçu, O., Kuzgun, A., Kwak, S., Laippala, V., Lam, L., Lambertino, L., Lando, T., Larasati, S. D., Lavrentiev, A., Lee, J., Lê Hồng, P., Lenci, A., Lertpradit, S., Leung, H., Levina, M., Li, C. Y., Li, J., Li, K., Li, Y., Lim, K., Lima Padovani, B., Lindén, K., Ljubešić, N., Loginova, O., Lusito, S., Luthfi, A., Luukko, M., Lyashevskaya, O., Lynn, T., Macketanz, V., Mahamdi, M., Maillard, J., Makazhanov, A., Mandl, M., Manning, C., Manurung, R., Marşan, B., Mărănduc, C., Mareček, D., Marheinecke, K., Martínez Alonso, H., Martín-Rodríguez, L., Martins, A., Mašek, J., Matsuda, H., Matsumoto, Y., Mazzei, A., McDonald, R., McGuinness, S., Mendonça, G., Merzhevich, T., Miekka, N., Mischenkova, K., Misirpashayeva, M., Missilä, A., Mititelu, C., Mitrofan, M., Miyao, Y., Mojiri Foroushani, A., Molnár, J., Moloodi, A., Montemagni, S., More, A., Moreno Romero, L., Moretti, G., Mori, K. S., Mori, S., Morioka, T., Moro, S., Mortensen, B., Moskalevskyi, B., Muischnek, K., Munro, R., Murawaki, Y., Müürisep, K., Nainwani, P., Nakhlé, M., Navarro Horñiacek, J. I., Nedoluzhko, A., Nešpore-Bērzkalne, G., Nevaci, M., Nguyễn Thị, L., Nguyễn Thị Minh, H., Nikaido, Y., Nikolaev, V., Nitisaroj, R., Nourian, A., Nurmi, H., Ojala, S., Ojha, A. K., Olúòkun, A., Omura, M., Onwuegbuzia, E., Osenova, P., Östling, R., Øvrelid, L., Özateş, Ş. B., Özçelik, M., Özgür, A., Öztürk Başaran, B., Park, H. H., Partanen, N., Pascual, E., Passarotti, M., Patejuk, A., Paulino-Passos, G., Peljak-Łapińska, A., Peng, S., Perez, C.-A., Perkova, N., Perrier, G., Petrov, S., Petrova, D., Phelan, J., Piitulainen, J., Pirinen, T. A., Pitler, E., Plank, B., Poibeau, T., Ponomareva, L., Popel, M., Pretkalniņa, L., Prévost, S., Prokopidis, P., Przepiórkowski, A., Puolakainen, T., Pyysalo, S., Qi, P., Rääbis, A., Rademaker, A., Rahoman, M., Rama, T., Ramasamy, L., Ramisch, C., Rashel, F., Rasooli, M. S., Ravishankar, V., Real, L., Rebeja, P., Reddy, S., Regnault, M., Rehm, G., Riabov, I., Rießler, M., Rimkutė, E., Rinaldi, L., Rituma, L., Rizqiyah, P., Rocha, L., Rögnvaldsson, E., Romanenko, M., Rosa, R., Roșca, V., Rovati, D., Rudina, O., Rueter, J., Rúnarsson, K., Sadde, S., Safari, P., Sagot, B., Sahala, A., Saleh, S., Salomoni, A., Samardžić, T., Samson, S., Sanguinetti, M., Sanıyar, E., Särg, D., Saulīte, B., Sawanakunanon, Y., Saxena, S., Scannell, K., Scarlata, S., Schneider, N.,

Schuster, S., Schwartz, L., Seddah, D., Seeker, W., Seraji, M., Shahzadi, S., Shen, M., Shimada, A., Shirasu, H., Shishkina, Y., Shohibussirri, M., Sichinava, D., Siewert, J., Sigurðsson, E. F., Silveira, A., Silveira, N., Simi, M., Simionescu, R., Simkó, K., Šimková, M., Simov, K., Skachedubova, M., Smith, A., Soares-Bastos, I., Sourov, S., Spadine, C., Sprugnoli, R., Steingrímsson, S., Stella, A., Straka, M., Strickland, E., Strnadová, J., Suhr, A., Sulestio, Y. L., Sulubacak, U., Suzuki, S., Szántó, Z., Taguchi, C., Taji, D., Takahashi, Y., Tamburini, F., Tan, M. A. C., Tanaka, T., Tanaya, D., Tella, S., Tellier, I., Testori, M., Thomas, G., Torga, L., Toska, M., Trosterud, T., Trukhina, A., Tsarfaty, R., Türk, U., Tyers, F., Uematsu, S., Untilov, R., Urešová, Z., Uria, L., Uszkoreit, H., Utka, A., Vajjala, S., van der Goot, R., Vanhove, M., van Niekerk, D., van Noord, G., Varga, V., Villemonte de la Clergerie, E., Vincze, V., Vlasova, N., Wakasa, A., Wallenberg, J. C., Wallin, L., Walsh, A., Wang, J. X., Washington, J. N., Wendt, M., Widmer, P., Wijono, S. H., Williams, S., Wirén, M., Wittern, C., Woldemariam, T., Wong, T.-s., Wróblewska, A., Yako, M., Yamashita, K., Yamazaki, N., Yan, C., Yasuoka, K., Yavrumyan, M. M., Yenice, A. B., Yıldız, O. T., Yu, Z., Yuliawati, A., Žabokrtský, Z., Zahra, S., Zeldes, A., Zhou, H., Zhu, H., Zhuravleva, A., and Ziane, R. (2021). Universal dependencies 2.9. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.

Zhang, H., Bai, J., Song, Y., Xu, K., Yu, C., Song, Y., Ng, W., and Yu, D. (2019a). Multiplex word embeddings for selectional preference acquisition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5247–5256, Hong Kong, China. Association for Computational Linguistics.

Zhang, H., Ding, H., and Song, Y. (2019b). SP-10K: A large-scale evaluation set for selectional preference acquisition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 722–731, Florence, Italy. Association for Computational Linguistics.

Zhang, T., Wu, F., Katiyar, A., Weinberger, K. Q., and Artzi, Y. (2020). Revisiting few-sample BERT fine-tuning. *arXiv preprint arXiv:2006.05987*.

Zhang, Z., Wu, Y., Zhao, H., Li, Z., Zhang, S., Zhou, X., and Zhou, X. (2019c). Semantics-aware BERT for language understanding. *arXiv preprint arXiv:1909.02209*.

Zhu, X., Li, T., and De Melo, G. (2018). Exploring semantic properties of sentence embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 632–637.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Čermák, F. and Rosen, A. (2012). The Case of InterCorp, a multilingual parallel corpus. In *International Journal of Corpus Linguistics*, volume 13, pages 411–427.