In-depth exploration of the syntactic capabilities of autoencoding language models for downstream applications

Laura Pérez-Mayos

DOCTORAL THESIS UPF / 2022

THESIS SUPERVISORS
Dr. Leo Wanner
Dr. Miguel Ballesteros

Dept. of Information and Communications Technologies

Universitat Pompeu Fabra Barcelona

To fantasy writers, who taught us not only that dragons exist, but that they can be beaten.

Thanks

This dissertation marks the ending of a long path, and naturally there is many people to thank and only myself to blame. I would like to start by thanking my supervisors, Leo and Miguel, for their support and their confidence. You both provided invaluable guidance along the way, boosting my curiosity and helping me become a stronger researcher and writer.

I've been lucky to share this path with a group of amazing people at TALN. To Paula and Alba, because collaborating with you was extremely easy and exciting. To Carla, because it felt sooo good to have another feminist in the office! To Giorgia, Guille, Alex, Ahmed, Pablo and all the other students, because we navigated together a very difficult path, and managed to have fun most of the way. And a huge 'Thank you!!' to Roberto, because if it wasn't for you I would never have finished my PhD. Thanks for being an awesome friend and a wise advisor :D We did it! We published a paper with 'BERT' in the title!!

I'd also like to thank the researchers that inspired me and helped me in many ways. To Mireia Farrús, who was an awesome master's thesis supervisor and helped me getting my first publication. Thanks for encouraging me to pursue my PhD! To the seniors of the group, and specially to Luis, who taught us how to find the best conferences, and to Alp, who reminded me that it was supposed to be fun and interesting. To Laura Aina, with whom I started studying contextual word embeddings, and to the DL reading group, for the motivation. To Shuai and Yogarshi, who made my internship at Amazon really fun and interesting. And to those who share their knowledge online, making it accessible to many others, most specially to Sebastian Ruder, who made it so much easier to follow the advances of the field, and to Jay Alammar, who illustrated the inner working of the Transformer wonderfully.

On a more personal note, I'd like to thank the Volley Circus family, because we managed not to win a single match, and it was awesome nevertheless! Special mention to my oldest and dearest friend Vero, who cook diner for me after every single one of my master's exams, and who gifted me the most practical pencil case ever. I will always save the last

croqueta for you! And to Víctor, with whom I started this academic adventure many years ago. Thanks for making life so fun and easy :D

I somehow managed to keep some non-PhD-related friends through years of being always too busy, and I feel very thankful to them. Anna, Iván, Isart, Aldo, Cris, Lidia: I did it!! To all my friends from PyLadiesBCN and PyBCN, let's celebrate!

And last, I'd like to warmly thank my family. Huge thanks to my parents, who encouraged me to follow my own path since I was I child, and to my brother Javi, who has been so many times the light to guide my steps. I would not be who I am or where I am without their unconditional support. To my sister-in-law Conxita, whose pursue of knowledge is absolutely inspiring. And finally to you, Àlex, for your patience, your love and your confidence. T'estimo molt!

Abstract

Pretrained Transformer-based language models have quickly replaced traditional approaches to model NLP tasks, pushing the state of the art to new levels, and will certainly continue to be very influential in the years to come. In this thesis, we offer an extensive empirical comparison of the morpho-syntactic capabilities of pretrained Transformer-based autoencoding models. We analyse the syntactic generalisation abilities of different widely-used pretrained models, comparing them along two dimensions: 1– language: monolingual (English and Spanish) and multilingual models; and 2– pretraining objectives: masked language modeling and next sentence prediction. We complement the analysis with a study of the impact of the pretraining data size on the syntactic generalisation abilities of the models and their performance on different downstream tasks. Finally, we investigate how the syntactic knowledge encoded in the models evolves along the fine-tuning process on different morpho-syntactic and semantics-related downstream tasks.

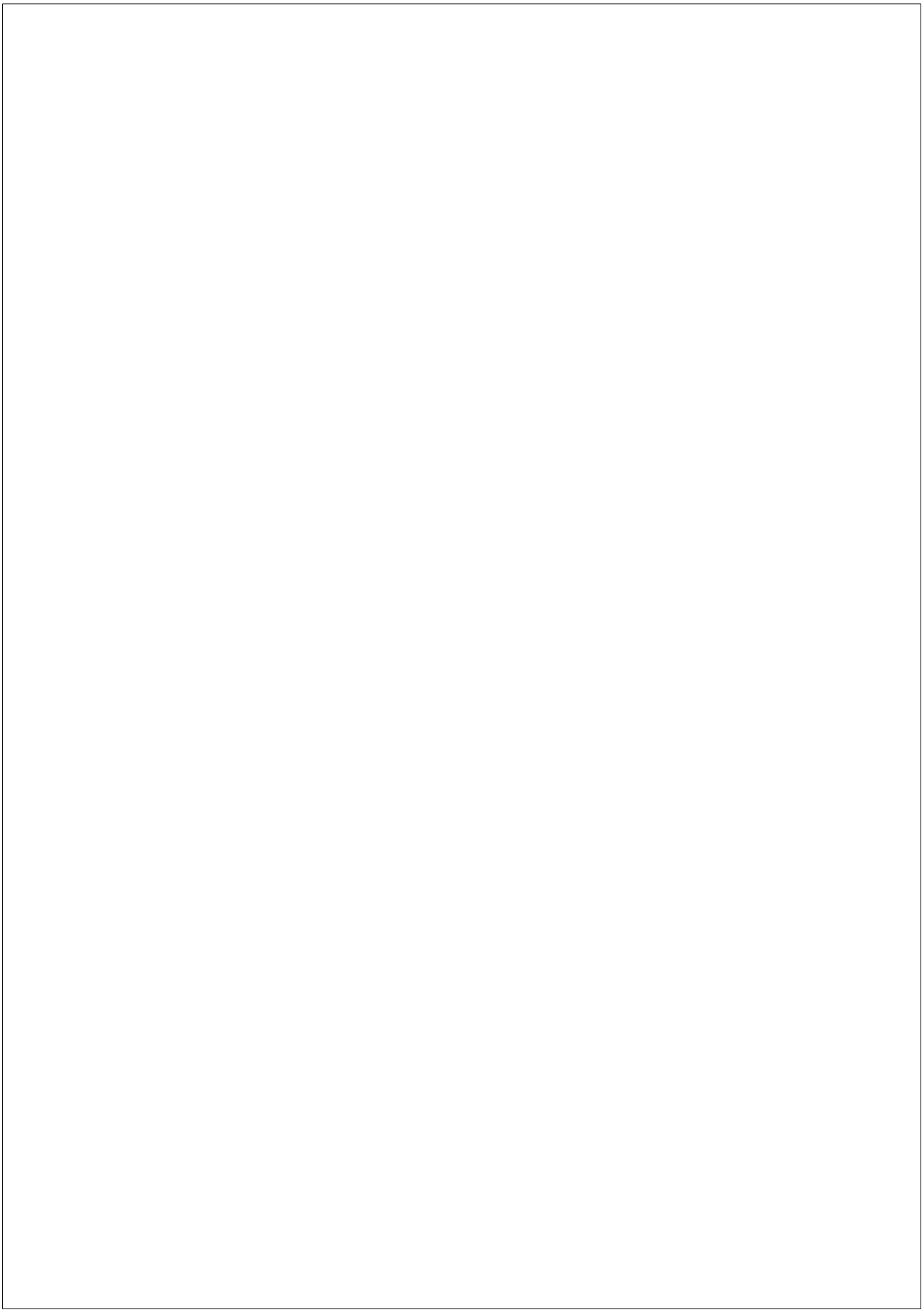
Resum

Els models de llenguatge preentrenats basats en Transformer han reemplaçat ràpidament els models tradicionals de Processat del Llenguatge Natural, fent avançar l'estat de l'art a nous nivells, i de ben segur continuaran sent molt influents durant els propers anys. En aquesta tesi presentem una extensa comparativa empírica de les capacitats morfosintàctiques de models de llenguatge preentrenats basats en Transformer de tipus *autoencoding*. Analitzem les capacitats de generalització sintàctica de diferents models que es fan servir habitualment, comparant-los en base a: 1– llenguatge: models monolingües (anglès i castellà) i multilingües; i 2– objectius d'entrenament: modelat del llenguatge amb màscares i predicció de la següent frase. Per complementar la comparativa, estudiem l'impacte del volum de les dades d'entrenament en les habilitats de generalització sintàctica dels models i el seu rendiment en diverses tasques. Finalment, investiguem com el coneixement sintàctic codificat als models evoluciona durant el seu entrenament en diverses tasques sintàctiques i semàntiques.

Contents

List of Figures

List of Tables



Abbreviations

Artificial Intelligence, Machine Learning

AI Artificial Intelligence

DL Deep Learning

$FFNN$ Feed-forward Neural Network

LSA Latent Semantic Analysis

$LSTM$ Long-Short Term Memory Network

ML Machine Learning

MLP Multi-Layer Perceptron

MSE Mean Squared Error

MTL Multi-task Learning

NLP Natural Language Processing

NLU Natural Language Understanding

NN Neural Network

$ON - LSTM$ Ordered Neurons LSTM

$ReLU$ Rectified Linear Unit

RNN Recurrent Neural Network

$RNNG$ Recurrent Neural Network Grammars

SVD Singular Value Decomposition

Benchmarks, Datasets

$CoLA$ Corpus of Linguistic Acceptability

$GLUE$ General Language Understanding Evaluation benchmark

$MLQA$ MultiLingual Question Answering

$MNLI$ Multi-Genre Natural Language Inference Corpus

$MRPC$ Microsoft Research Paraphrase Corpus

PTB Penn Treebank

$QNLI$ Question-answering Natural Language Inference

$QQPT$ Quora Question Pairs

$RACE$ Large-scale ReAding Comprehension Dataset From Examinations

RTE Recognising Textual Entailment

$SQuAD$ Stanford Question Answering Dataset

$SST-2$ Stanford Sentiment Treebank v2

$STS-B$ Semantic Textual Similarity Benchmark

$SWAG$ Situations With Adversarial Generations

UD Universal Dependencies

$XQuAD$ Cross-lingual Question Answering Dataset

$XTREME$ Cross-lingual TRansfer Evaluation of Multilingual Encoders

Language Modeling

$BERT$ Bidirectional Encoder Representations from Transformers

$ELMo$ Embeddings from Language Models

GPT Generative Pretrained Transformer

LM Language Model

$MLLM$ MultiLingual Language Model

PLM Pretrained Language Model

Metrics

LAS Labeled Attachment Score

MLE Maximum Likelihood Estimation

PMR Perfect Match Ratio

UAS Unlabeled Attachment Score

$UUAS$ Undirected Unlabeled Attachment Score

Miscellaneous

API Application Programming Interface

BPE Byte-pair Encoding

$CRFM$ Center for Research on Foundation Models

FGD Filler-Gap Dependencies

HAI Stanford Institute for Human-Centered Artificial Intelligence

LDD Long-distance dependencies

$MVRR$ Main verb/reduced relative clause

NP Noun Phrase

NPI Negative Polarity Item

PoS Part-of-speech

SD Semantic Dependencies

SG SyntaxGym

VP Verb Phrase

Tasks, Applications

MLM Masked Language Modeling

NER Named Entity Recognition

NSP Next Sentence Prediction

QA Question Answering

$SCWS$ Contextual Word Similarities

SRL Semantic Role Labeling

Chapter 1

INTRODUCTION

1.1 Motivation

Language is considered to be one of the key components of human intelligence, the source of creativity, cultural enrichment, and complex social structure. Central to human thought, language shapes how social and emotional relations are formed, how we identify ourselves socially and personally, and how we record knowledge and develop societal intelligence (Bommasani et al., 2021). It is estimated that there are more than 7,000 human languages in the world, and they are both incredibly diverse in the ways that they express and structure the information they convey, while also exhibiting surprising concordance in the richness of what makes a language (Comrie, 1989). On the quest towards general artificial intelligence, developing systems that can understand and generate human language has become a milestone that drives research in computational linguistics (Nilsson, 2009).

Natural Language Processing (NLP) is a field of research concerned with building computational tools for the automatic analysis, representation and generation of human language. NLP addresses a wide range of tasks, including, e.g., dependency parsing (Kübler et al., 2009; Nivre et al., 2007; Buchholz and Marsi, 2006), entity linking (Shen et al., 2021, 2014; Rao et al., 2013) or named entity recognition (Lample et al., 2016; Nadeau

and Sekine, 2007), and has multiple practical applications, from chatbots (Adamopoulou and Moussiades, 2020; Shum et al., 2018) and virtual assistants (Rawassizadeh et al., 2019; Siebra et al., 2018) to recommendation systems (Naumov et al., 2019; Isinkaye et al., 2015) and autocorrectors (Hládek et al., 2020). As we will see in detail in Section 2.3, which reviews the recent history of NLP, many of the most recent advances in the field, which draw upon neural models, are reduced to a form of language modeling (Bengio et al., 2000; Mikolov et al., 2010; Graves, 2013; Vaswani et al., 2017; Devlin et al., 2019a), i.e., to systems that determine the probability of a given sequence of words occurring in a sentence. One of the reasons behind their success is grounded in the fact that to predict the next word (or a set of missing words), language models are forced to encode complex syntactic and semantic information (Goldberg, 2019; Jawahar et al., 2019). Consequently, language models are excellent information encoders (Devlin et al., 2019a), well suited to generate representations of words and sentences. These representations, generally known as embeddings, are a central piece of NLP, as they allow us to encode information into low-dimensional vector representation spaces and are easily integrable into machine learning algorithms.

The widespread adoption of neural network approaches over the last years has boosted the evolution of language models, and traditional approaches based on N-Gram models (Shannon, 1948) were replaced first by feed-forward neural networks (Bengio et al., 2000) and later by specialised architectures for sequential data –recurrent neural networks (RNNs; Mikolov et al., 2010) and Long-Short Term Memory networks (LSTMs; Graves, 2013). In 2017, Vaswani et al. (2017) presented the Transformer, a new architecture that would become the seed of the NLP revolution the following year. Indeed, in 2018, the publication of the first contextual embeddings (Peters et al., 2018b), which relied on a bidirectional LSTM architecture, was almost immediately shadowed by the development of BERT (Devlin et al., 2019a), the first pretrained Transformer-based language model. BERT quickly became the new state of the art in a wide range of tasks and benchmarks, clearly outperforming all previous approaches. At the same time, the need for methods relevant to explaining

the decisions and analyzing the inner workings of NLP models became more urgent, giving birth to the new emergent subfield of Explainable NLP, that became an independent track at the Association for Computational Linguistics's main conference in 2020; cf. eg. Søgaard (2021).
The publication of BERT inspired the development of many new pretrained Transformer-based language models (cf. Section 2.5), coining the research of the entire field of NLP, and will certainly continue to be very influential in the years to come. While traditional approaches are quickly replaced with these new, powerful models, two main questions emerge: what do these models learn, and how do they use this knowledge? To shed light on these and more questions, this thesis is devoted to the study of the syntactic capabilities of modern language models, and their impact on downstream applications.

1.2 Research objectives

Pretrained Transformer-based autoencoding language models –e.g. BERT (Devlin et al., 2019a) and RoBERTa (Liu et al., 2019b); Section 2.5.3– , are trained in an unsupervised manner from raw text using different tasks such as Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). Even though the models are never exposed to explicit linguistic structures, it has been shown that they learn major aspects of these structures (Manning et al., 2020; Rogers et al., 2020; Jawahar et al., 2019). This thesis aims at unveiling what kind of syntactic knowledge do pretrained Transformer-based autoencoding language models learn, and how do they use it. We focus on the morpho-syntactic layer of the linguistic structure, studying the syntactic capabilities of the models, from their development to their application to downstream tasks:

Syntactic generalisation abilities of pretrained models (Chapter 4). Multilingual Transformer-based language models, usually pretrained on more than 100 languages, have been shown to achieve outstanding results in a wide range of cross-lingual transfer tasks. However, it remains

unknown whether the optimisation for different languages conditions the capacity of the models to generalise over syntactic structures, and how languages with syntactic phenomena of different complexity are affected. Specifically, we address the following aspects:

- Do multilingual models generalise equally well across languages?
- How well do monolingual models generalise over syntactic phenomena compared to multilingual models?
- Does the presence of modifiers affect the generalisation capabilities of the models?
- Does the nature of the training procedures employed to train the models affect the generalisation capabilities of the models?

Impact of the pretraining data size on the syntactic abilities of the models (Chapter 5). While pretraining methods are very convenient, they are expensive in terms of time and resources. This calls for a study of the impact of pretraining data size on the knowledge of the models. Specifically, we investigate:

- Do models pretrained with more data encode more syntactic information?
- Do models pretrained with more data generalise better over syntactic phenomena?
- Are models pretrained with more data more robust to the presence of modifiers?
- Do models pretrained with more data offer a better performance on downstream tasks such as dependency parsing and paraphrase identification?
- Is there a correlation between the language modeling abilities of the models and their syntactic generalisation abilities?

Evolution of syntactic knowledge during fine-tuning (Chapter 6). Pretrained models are often fine-tuned on downstream tasks, and therefore it becomes increasingly important to understand how the encoded knowledge evolves along the fine-tuning process.

- Is the syntactic information initially encoded in the models forgotten, preserved, or reinforced along the fine-tuning process?
- Does this evolution depend on the task in which the models are fine-tuned?

1.3 Contributions

The main contribution of this dissertation is an extensive empirical comparison of the morpho-syntactic capabilities of pretrained Transformer-based autoencoding language models, from their generalisation abilities over syntactic phenomena to their performance on downstream tasks. We thoroughly explore the syntactic generalisation abilities of different widely used pretrained models, comparing them along two dimensions: 1– language: monolingual (English and Spanish) and multilingual models (pretrained with more than 100 languages); and 2– pretraining objectives: masked language modeling and next sentence prediction. First, we study the ability of the models to generalise over different syntactic phenomena. Then, we focus on the impact of pretraining data size on the syntactic knowledge of the models. Finally, we study the evolution of the encoded syntactic knowledge along the fine-tuning process in different tasks.

1.4 Thesis outline

In Chapter 2, we offer a review of background information that is relevant to understand the contents of this thesis. First, we contextualise the research field and review important concepts related to the training of neural networks. Then, we present the evolution of computational neural methods for NLP. Next, we describe the language modeling task and the

traditional approaches to solve it, along with their limitations, followed by the current neural-based approaches. Last, we introduce pretrained language models, which are the central object of the thesis, reviewing the different dimensions that play a key part in their undeniable success, from architectures and pretraining objectives to information representation.
In Chapter 3, we present the current state of the art on the different topics covered in this thesis, from the linguistic knowledge encoded in pretrained language models and how to assess it to the evolution of syntactic knowledge during the fine-tuning process.
In Chapter 4, we explore the syntactic generalisation abilities of monolingual and multilingual pretrained models, analysing whether the optimisation for different languages conditions the capacity of the models to generalise over syntactic structures, and how languages with syntactic phenomena of different complexity are affected. Furthermore, we present SyntaxGymES, a novel ensemble of targeted syntactic tests in Spanish.
In Chapter 5, we study the impact of pretraining data size on the knowledge of pretrained language models, analysing models trained on incremental sizes of raw text data by means of structural probes, a targeted syntactic evaluation and a comparison of the performance of the models on different downstream applications. We complement our study with an analysis of the cost-benefit trade-off of training such models.
In Chapter 6, we study how the knowledge encoded in pretrained language models evolves along fine-tuning. Specifically, we analyse the evolution of the syntactic information embedded in the models along the fine-tuning process in six different tasks, covering all levels of the linguistic structure, to show whether it is forgotten, reinforced or preserved along the process.
Finally, in Chapter 7 we present a summary of our findings and we draw conclusions on the syntactic abilities of modern language models. Moreover, we offer a brief overview of the main concerns raised by the development of ever larger pretrained models, and we present and discus interesting leads to follow in the future.

Chapter 2

BACKGROUND

This chapter offers a review of background information that is relevant for understanding the contents of subsequent chapters. We start with an overview of the Artificial Intelligence scientific field and its subfields (Section 2.1). Then, we review the neural history of NLP (Section 2.3), from N-Gram models to modern Transformer-based architectures. Next, we formally describe the task of language modeling (Section 2.4), reviewing both the traditional and neural approaches to solve it. Last, we deepen the discussion of the different types of pretrained Transformer-based language models (Section 2.5).

2.1 Contextualising the research field

This thesis is devoted to the study of Deep Learning language models, and thus we start by offering a clear picture of how Deep Learning is subsumed into the Neural Networks family of Machine Learning techniques, a subfield of the Artificial Intelligence field (Figure 2.1).
Artificial Intelligence (AI) is a scientific field that aims to develop machines able to match human intelligence, and includes goals like reasoning, knowledge representation, planning, learning, natural language processing, perception and robotics. Machine Learning (ML) is a subfield of AI, comprised by computer algorithms that learn to make predic-

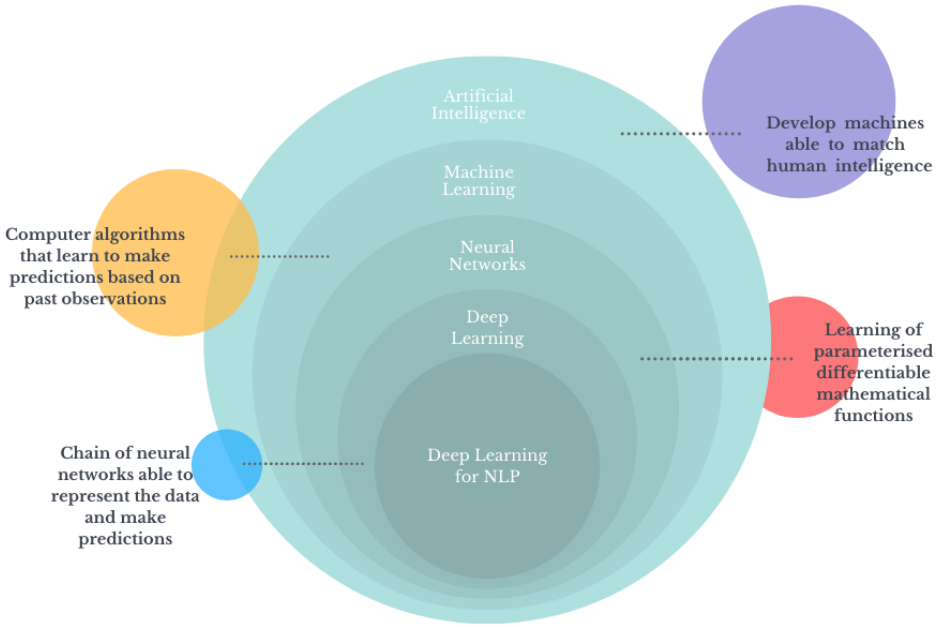


Figure 2.1: Deep Learning is subsumed into the Neural Networks family of Machine Learning techniques, a subfield of the Artificial Intelligence field.

tions based on past observations. Neural Networks (NNs) is a family of ML techniques inspired by the human brain that can be characterised as learning of parameterised differentiable mathematical functions. Neural Networks make up the backbone of Deep Learning (DL) algorithms, a subfamily of Neural Networks characterised by having at least three, but usually more, Neural Networks chained together, commonly referred to as layers. They aim not only to predict but also to correctly represent the data such that it is suitable for prediction: given a set of input-output examples, Deep Learning methods work by feeding the input data into the network, which successively transforms it while it traverses through each layer until a final transformation predicts the output (Figure 2.2). The transformations that take place in each layer are also learned in the process, and thus the learning of the correct representation of the data is performed automatically by the network.

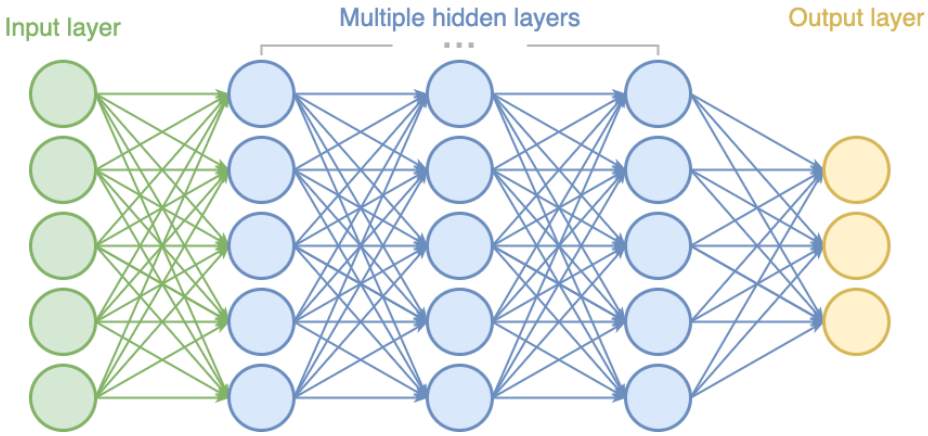


Figure 2.2: Deep neural network. Given a set of input-output examples, deep Learning methods work by feeding the input data into the network, which successively transforms it while it traverses through each layer until a final transformation predicts the output.

2.2 Training Neural Networks

In this section, we describe the training procedure of a simple fully-connected neural network to illustrate some concepts that will become useful later on, when we present the different approaches to building language models with neural networks. For a more complete review of training neural networks for NLP, see (Goldberg, 2017).

Neural Networks are differentiable parameterised functions. The network illustrated in fig. 2.3 has 5 neurons, arranged in 3 fully-connected layers, that is, each neuron from one layer is connected to each neuron from the following layer. The neurons at the input layer represent the input of the network, x_1 and x_2, and they do not perform any computation. The rest of the neurons are in fact mathematical functions: they take two numbers as input, e.g. x_1 and x_2, combine them with a weights vector $\mathbf{w}$ and apply a non-linear function on top, know as the *activation function*, to output a single number, its *activation*.[1] Two common activation functions are the

[1]Notice that, as the neurons are chained, if we do not add a non-linearity their combination would be also a linear function, not more powerful than a single neuron.

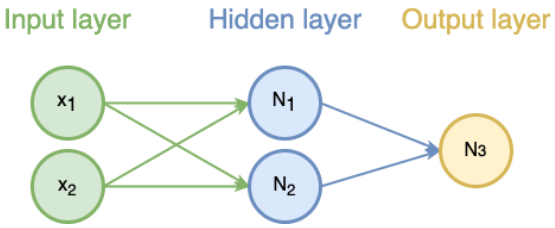


Figure 2.3: Fully-connected neural network.

Rectified Linear Unit (ReLU): $ReLU(x) = max(x, 0)$, and the Sigmoid, that outputs a number between 0 and 1 that we can treat as a probability, very useful for the output layer. Equation 2.1 shows the formula for the network depicted in fig. 2.3 when using ReLU activation functions for the hidden layer, and Sigmoid for the output layer, where superscripts of $\mathbf{w}$ denote the index of the neuron, and subscripts the index of the input.

$$f(x_1, x_2) = Sigmoid(w_1^3 ReLU(w_1^1 x_1 + w_2^1 x_2) + w_2^3 ReLU(w_1^2 x_1 + w_2^2 x_2)) \tag{2.1}$$

Thus, a neural network is a complex non-linear function parameterised by some weights. These weights are randomly initialised, and they are changed during training in such a manner that the network outputs match as closely as possible the expected outputs. Neural networks are trained with examples, using a *loss function*, that is, a function stating the loss of predicting $\hat{y}$ when the true output is y. There exists a wide range of loss functions that can be used in the context of neural networks, and while deepening into them is out of the scope of this work, we will use here the Mean Squared Error (MSE) as an example.[2] Having a training set with m examples, we could take each training example, pass it through the network to get the final prediction ($\hat{y}$), subtract it from the actual number we expected (y) and square it, as depicted in Equation 2.2. This phase is

[2]Further information on loss functions in the context of neural networks can be found in (LeCun and Huang, 2005; LeCun et al., 2006; Goodfellow et al., 2016).

the *forward pass*.

$$L_{MSE}(y, \hat{y}) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \tag{2.2}$$

We want the loss to be as small as possible, because it will mean that the predictions are really close to the expected values. Thus, the problem or training is equivalent to the problem of minimising the loss function. There exist different algorithms to minimise the loss, many of them relying not only on the information provided by the loss function but also on its gradient. To update the weights, we compute the gradient of the loss function w.r.t the weights and take small steps, whose size is controlled by a hyperparameter called *learning rate*, in the opposite direction of the gradient. This phase is called the *backward pass* or *backpropagation*. By adjusting the weights in this manner, the loss will gradually decrease until it converges to some (local) *minima*. A commonly used optimisation method is the *Stochastic Gradient Descent* (Bottou and GO, 1998; Bottou, 2012), a stochastic approximation of gradient descent optimization that replaces the actual gradient (calculated from the entire data set) by an estimate calculated from a randomly selected subset of the data.

To conclude this section, let us now review some practicalities of the training procedure of deep neural networks that will become useful in following chapters:

Training, testing and development sets. In practice, we often train several models, compare their quality, and select the best one. The usual approach is to use a three-way split of the data into train, validation (also called development), and test sets. While the training set is used for the training phase, all the experiments, tweaks, error analysis, and model selection should be performed based on the validation set. Then, a single run of the final model over the test set will give a good estimate of its generalisation over unseen examples.

Shuffling. During training, it is usual to go several times over the training data, feeding it to the network. Each such pass is called an *epoch*. The order in which the training examples are fed into the network is important, and therefore it is a common practice to perform random sampling without replacement, shuffling the training examples before each epoch.

Initialization. The non-convexity of the objective function means that the optimization procedure may get stuck in a local minima, and that starting from different initial points (e.g., different random values for the parameters) may result in different results. Thus, it is advised to run several restarts of the training, starting at different random initialisations, and choosing the best one based on a development set.

Random restarts. When training complex networks, different random initialisations are likely to end up with different final solutions, exhibiting different accuracies. Thus, it is advisable to run the training process several times, each with a different random initialization, and choose the best one based on the development set. Moreover, the average model accuracy across random seeds gives a hint as to the stability of the process.

Vanishing and exploding gradients. It is common for the error gradients to either vanish (become exceedingly close to 0) or explode (become exceedingly high) as they propagate back through the network. Dealing with exploding gradients can be solved by simply clipping the gradients if their norm exceeds a given threshold. However, dealing with vanishing gradients is still an open research question, and solutions include making networks shallower or using specialized architectures that are designed to assist in gradient flow (e.g., LSTMs).

2.3 The neural history of NLP

In this section, we review the evolution of computational neural methods for NLP, depicted in Figure 2.4. First, we present the early neu-

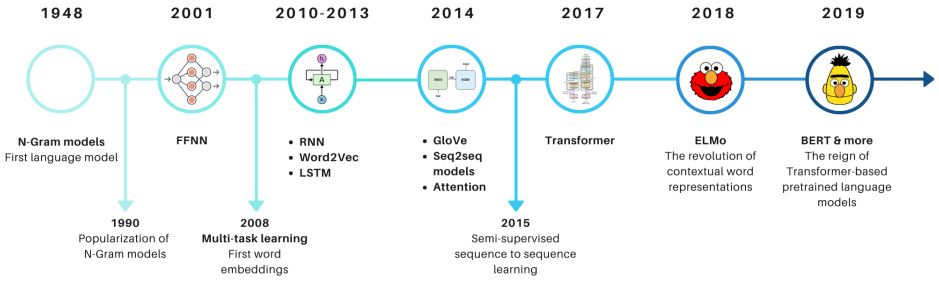


Figure 2.4: Evolution of NLP methods: from high-dimensional features to pretrained Transformer-based language models.

ral approaches to NLP (Section 2.3.1), from N-Gram models to RNN architectures and sequence-to-sequence models. Then, we describe the first approaches based on pretrained word representations (Section 2.3.2). Last, we analyse in depth the revolution of contextual word representations (Section 2.3.3).

2.3.1 Neural models for NLP

Many of the most important advances in NLP reduce to a form of language modeling, and thus the history of NLP is closely tied to the history of language models. Classic approaches to language modeling are based on N-Grams (Shannon, 1948), probabilistic models that assumed that the next word in a sequence depends only on the last N words (Markov assumption). However, they suffer from what is known as *the curse of dimensionality* (Section 2.4.1), and in the early 2000s the first nonlinear neural models were proposed to alleviate some of the related limitations. The first neural language model was proposed in (Bengio et al., 2000), and consisted of a single hidden layer feed-forward network used to predict the next word of a sequence (Section 2.4.2.1). The model uses a look-up table from which to extract word representations as vectors, and although feature vectors already existed by this time (Rumelhart et al., 1985; McClelland et al., 1986), Bengio et al.’s work greatly contributed to popularise the concept. Currently, these vector representations are known

as word embeddings, and constitute a major component of modern NLP systems.

More recently (2010-2013), specialised models for sequential data were developed, and feed-forward neural networks (FFNNs) were replaced by recurrent neural networks (RNNs; Mikolov et al., 2010) and Long-Short Term Memory networks (LSTMs; Graves, 2013), allowing to abandon the Markov assumption. RNNs take as input a sequence of items, and produce a fixed size vector that summarises that sequence. They allow us to condition on entire sentences while taking word order into account, and alleviate the statistical estimation problems derived from data sparsity. Rarely used as standalone components, these networks are usually employed as input-transformers trained to produce informative representations for feed-forward networks that will operate on top of them.

In 2014, Sutskever et al. (2014) proposed sequence-to-sequence models, a new framework for mapping one sequence to another relying on two neural components: an encoder NN that processes the input sentence symbol by symbol and compresses it into a vector representation, and a decoder NN that predicts the output symbol by symbol based on the encoder state, taking as input at every step the previously predicted symbol. Naturally, this framework was specially suited for machine translation tasks, and was further popularised by Google in (Wu et al., 2016). The framework is very flexible, and has been widely adopted for natural language generation tasks, with different models taking on the role of the encoder and the decoder. For example, it has been used to generate captions based on images (Vinyals et al., 2015) or to generate a description based on source code changes (Loyola et al., 2017).

The main problem with sequence-to-sequence models is that they require to compress the entire content of the source sequence into a fixed-size vector. In 2014, Bahdanau et al. (2015) presented Attention, a mechanism that allows the decoder to look back at the source sequence hidden states, which are then provided as a weighted average as additional input to the decoder. Thus, it is potentially useful for any task that requires making decisions based on certain parts of the input, and has a wide range of applications. This technique, which we will describe in Section 2.4.2.3,

grew to be the backbone of all current NLP approaches, as we will see all along this work.

2.3.2 Popularisation of pretrained word representations

The popularisation of the use of pretrained word representations is closely tied to the introduction of multi-task learning (MTL) in NLP, a subfield of machine learning in which multiple learning tasks are solved at the same time. MTL improves generalization by leveraging the domain-specific information contained in the training signals of related tasks.

The first applications of multi-task learning to NLP were presented in (Collobert and Weston, 2008; Collobert et al., 2011), were the word-embedding matrices are shared between two models trained on different tasks (including part-of-speech tagging, semantic role labeling and named entity recognition, among others), enabling the models to share general low-level information (Figure 2.5). This idea became the seed of pretraining word embeddings, and in recent years leveraging existing or *artificial* tasks to pretrain embeddings has become the backbone of many NLP methods.

In 2013, Mikolov et al. (2013a,b) introduced Word2Vec, a novel technique to efficiently learn high-quality word embeddings from huge corpora, transferable across NLP applications. Word2Vec is based on a simple but efficient feed-forward neural architecture trained with a language modeling objective. They proposed two different techniques: *continuous bag-of-words (CBOW)* predicts the centre word based based on the surrounding words, and *skip-gram* does the opposite. Importantly, training on very large corpora enables the embeddings to approximate certain morpho-syntactic, semantic and ontological properties between words, such as gender, verb tense, and country-capital relations, as shown in Figure 2.6, and many studies were devoted to investigate these relations (Handler, 2014; Arora et al., 2016; Mimno and Thompson, 2017; Antoniak and Mimno, 2018; Wendlandt et al., 2018; Jatnika et al., 2019; Miaschi and Dell'Orletta, 2020).

Using pretrained embeddings as initialisation was shown to improve per-

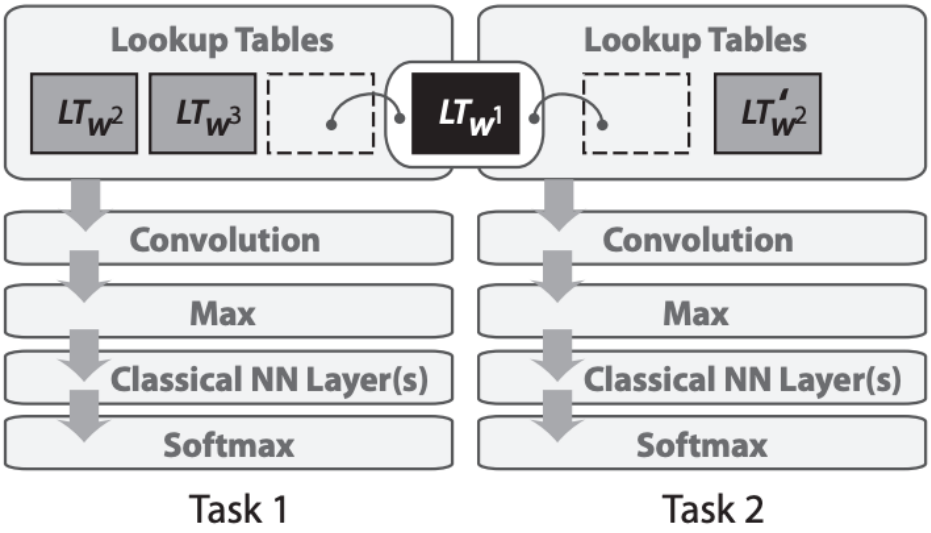


Figure 2.5: Example of multi-task learning with NN. The first layer (Convolution) extracts features for each word. The second layer (Max) extracts features from the sentence, and the following layers are FFNN layers. One lookup-table (in black) is shared, and the others are task specific. The principle is the same with more than two tasks. Source: Collobert and Weston (2008).

formance across a wide range of downstream tasks (Kim, 2014), and consequently the use of pretrained word embeddings was quickly popularised and has become an integral part of current NLP models. In 2014, a year after Word2Vec was published, Pennington et al. (2014) presented GloVe, a set of pretrained word embeddings leveraging statistical information by training only on the nonzero elements in a word-word co-occurrence matrix, rather than on the entire sparse matrix or on individual context windows in a large corpus. GloVe outperformed Word2Vec on word similarity[3] tasks and Named Entity Recognition (NER), proving that word embeddings can also be learned via matrix factorisation (Pennington et al., 2014; Levy and Goldberg, 2014). However, in 2015 Levy et al. (2015) revealed that much of the performance gains of word embeddings were

[3]With the exception of Contextual Word Similarities (SCWS; Huang et al., 2012), where Word2Vec outperformed GloVe.

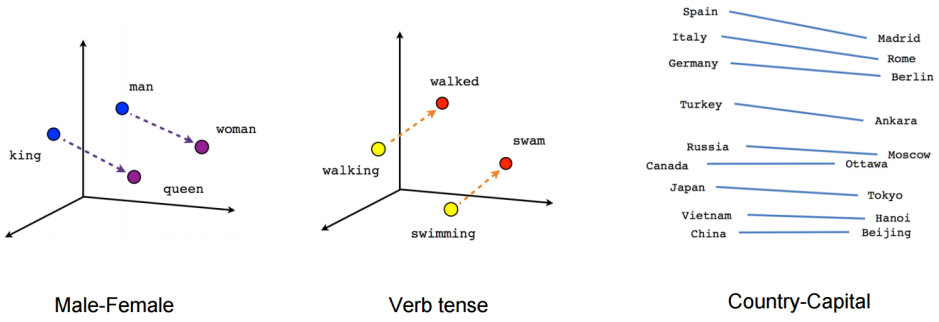


Figure 2.6: Word embeddings allow us to approximate certain morpho-syntactic, semantic and ontological properties between words, such as gender, verb tense, and country-capital relations.
Image source: https://towardsdatascience.com/evolution-of-word-representations-in-nlp-d4483fe23e93

due to certain system design choices and hyperparameter optimisations, rather than the embedding algorithms themselves, and that classic matrix factorisation methods like Singular Value Decomposition (SVD) and Latent Semantic Analysis (LSA) attained similar results to Word2Vec or GloVe.
Importantly, it was shown that the learnt relations are heavily biased. For instance, they exhibit female/male gender stereotypes to a disturbing extent, raising concerns because their widespread use tends to amplify these biases (Bolukbasi et al., 2016). Understanding what other biases they capture and finding ways to remove them will be key to developing fair NLP algorithms.

2.3.3 The revolution of contextual word representations

High quality representations should model not only complex word characteristics, such as its syntactic and semantic features, but also polysemy, that is, the capacity of a word or phrase to have multiple (often related) meanings. However, pretrained word representations (Mikolov et al., 2013b; Pennington et al., 2014) are context-agnostic and are only used to initialise the first layer in the models.

In 2018, Peters et al. (2018b) presented ELMo (Embeddings from Language Models), a new type of deep contextualised word representation that directly addresses both challenges and can be easily integrated into existing models. While Word2Vec and GloVe require supervised tasks to be trained, ELMo relies on language modeling to train a bidirectional LSTM, so that the model does not only have access to the next word, but also to the previous word. Given that language modeling is an unsupervised task and requires only unlabelled text, the training can scale to billions of tokens, new domains, and new languages, and is particularly beneficial for low-resource languages where labelled data is scarce. ELMo significantly improved the state of the art in every considered case across a range of challenging Natural Language Understanding (NLU) tasks. We explore this model further in Section 2.4.2.2.

While in ELMo, contextual embeddings are used as *frozen* features fed to a target model, pretrained language models can alternatively be fine-tuned on a target task data, as was proposed in (Ramachandran et al., 2017; Howard and Ruder, 2018). This second method became widely used in the following years with the introduction of the Transformer (Vaswani et al., 2017) and Transformer-based language models.

A few months after the publication of ELMo, a new pretrained language model was published by Google Brain in an event described as marking the beginning of a new era in NLP[4]: BERT (Devlin et al., 2019a), Bidirectional Encoder Representations from Transformers. The model, which we will describe in depth in Section 2.5.3.1, is built on several ideas and methods that had been developed by the NLP community in recent years, such as semi-supervised sequence learning (Dai and Le, 2015) and the Transformer. BERT quickly became the new state-of-the-art, clearly outperforming ELMo, and has inspired the development of many new models ever since (Section 2.5.3).

[4] https://twitter.com/lmthang/status/1050543868041555969

2.4 Language modeling

A language model is a statistical model that assigns probabilities to words and sentences. Formally, language modeling consists on assigning a probability $P(w_{1:n})$ to a given sequence of words $w_{1:n}$.
Traditional language models (Section 2.4.1) rely on the Markov assumption to approximate $P(w_{1:n})$, assuming that the next word in a sequence depends only on the last k words. However, these models do not scale well to larger ngrams, failing to capture long-range dependencies. Non-linear neural language models (Section 2.4.2) solve some of the limitations of traditional models. First, FFNNs allowed to condition on increasingly large context sizes (Section 2.4.2.1); later, RNNs and LSTMs allowed to represent arbitrarily sized sequential inputs in fixed-size vectors, while paying attention to the structure of the inputs (Section 2.4.2.2); and finally, the popularisation of the Transformer (Vaswani et al., 2017) allowed for language models with better parallelization and better suited to capture long-term dependencies (Section 2.4.2.3), with models such as BERT that allow conditioning on both the preceding and following words by relying on masked language modeling objectives (Section 2.5.3.1).

2.4.1 Traditional approaches and their limitations

Using the chain rule of probability, language modeling can be formulated as a sequence of word-prediction tasks whee each word is predicted conditioned on the preceding words:

$$P(w_{1:n}) = P(w_1)P(w_2|w_1)P(w_3|w_{1:2})...P(w_n|w_{1:n-1}) \tag{2.3}$$

While modeling the probability of a single word based on its left context seems easier than assigning a probability to an entire sequence, notice that the last term in the equation requires conditioning on almost the entire sequence. To approximate $P(w_{1:n})$, N-Gram models make use of the Markov assumption, and condition the probability of the next word in a sequence only on the last k words (usually 1, 2 or 3):

$$P(w_{i+1}|w_{1:i}) \approx P(w_{i+1}|w_{i-k:i}) \tag{2.4}$$

Relying on this assumption, traditional language models aim at accurately estimating $P(w_{i+1}|w_{i-k:i})$ given large amounts of text.
N-Gram models assume that the next word in a sequence depends only on the last k words: $P(w_{i+1} = m|w_{1:i}) \approx P(w_{i+1} = m|w_{i-k:i})$. For example, a Unigram model estimates $p(w_1, w_2, ..., w_n)$ as $p(w_1)p(w_2), ..., p(w_n)$, while a Trigram model estimates it as:
$p(w_1)p(w_2|w_1)p(w_3|w_2, w_1), ..., p(w_n|w_{n-1}, w_{n-2})$.
To estimate $p(w_n|w_{n-1}, w_{n-2}, ..., w_{n-N})$, we can use the Maximum Likelihood Estimation (MLE), simply counting the occurrence of word patterns in the corpus:

$$\hat{p}_{MLE}(w_{i+1} = m|w_{i-k:i}) = \frac{\#(w_{i-k:i+1})}{\#(w_{i-k:i})}) \tag{2.5}$$

However, if $w_{i-k:i+1}$ was never observed in the corpus, $\hat{p}_{MLE}(w_{i+1} = m|w_{i-k:i}) = 0$, which would result in a 0-probability assignment to the entire corpus because of the multiplicative nature of the sentence probability calculation, and an infinite perplexity. Given that zero-probability events are quite common,[5] two different techniques have been used to avoid them: *smoothing* and *back-off*.
Smoothing ensures an allocation of a small probability mass to every possible event. For example, *additive smoothing* (Lidstone, 1920), also called Laplace smoothing or $add - \alpha$ smoothing, assumes each event occurred at least α times in addition to its observations in the corpus:

$$\hat{p}_{add-\alpha}(w_{i+1} = m|w_{i-k:i}) = \frac{\#(w_{i-k:i+1} + \alpha)}{\#(w_{i-k:i}) + \alpha|V|}) \tag{2.6}$$

where $|V|$ is the vocabulary size and $0 < \alpha < 1$.
Back-off computes an estimate based on a $(k-1)$gram if the $k$gram was not observed. For example, the *Jelinek Mercer interpolated smoothing*:

$$\hat{p}_{int}(w_{i+1} = m|w_{i-k:i}) = \lambda_{w_{i-k:i}} \frac{\#(w_{i-k:i+1})}{\#(w_{i-k:i})}) + (1 - \lambda_{w_{i-k:i}})\hat{p}_{int}(w_{i+1} = m|w_{i-(k-1):i}) \tag{2.7}$$

[5] E.g. in a trigram language model with a vocabulary of 10,000 words there are $10,000^3 = 10^12$ possible triplets, so it is clear that many of them will not be observed in training corpora of, for example, 10^10 words.

Correctly setting the λ values has a big impact in the performance: $\lambda_{w_{i-k:i}}$ should depend on the content of the conditioning context $w_{i-k:i}$, differentiating rare contexts from frequent ones.
While language modeling approaches based on smoothed MLE estimates are easy to train, scale to large corpora, and work well in practice, they present important limitations. First, they suffer from the curse of dimensionality: a word sequence on which the model will be tested is likely to be different from all the word sequences seen during training. Therefore, they do not scale well to larger ngrams, both due to the sparsity of larger ngrams in the corpus and the memory required to work with larger conditioning contexts. Moreover, *back-off* techniques need to be designed by hand, making it hard to scale toward larger ngrams to capture long-range dependencies: e.g., to condition on the last 10 words the model needs to see a relevant 11-gram in the corpus, which is quite rare, and therefore the model backs off from the long history. Lastly, they do not generalise across contexts, and previous observations of similar events (*red bike*, *black bike*) do not condition the probability of observing a similar but not previously observed event (*green bike*).

2.4.2 Neural language models

Nonlinear neural language models solve some of the limitations of traditional language models presented in the previous section: they allow conditioning on increasingly large context sizes with only a linear increase in the number of parameters, they do not require manually designing back-off orders, and they generalise well across different contexts.

2.4.2.1 FFNNs for language modeling

To fight the curse of dimensionality, Bengio et al. (2000) proposed the first neural language model, a FFNN able to learn a distributed representation for words which allows each training sentence to inform the model about an exponential number of semantically neighboring sentences. The model learns simultaneously a distributed representation for each word along

with the probability function for word sequences, expressed in terms of these representations.
This kind of model takes a $k$gram of words $w_{1:k}$ as input, represents each word with an embedding vector $v(w) \in \mathbb{R}^{d_w}$, and creates the input vector $\mathbf{x}$ as concatenation of the k words: $\mathbf{x} = [v(w_1); v(w_2); ...; v(w_k)]$. The input $\mathbf{x}$ is then fed into a hidden layer, whose output is then provided to a softmax layer that outputs a probability distribution over the next word:

$$\begin{aligned} \hat{\mathbf{y}} = P(w_i|w_{1:k}) &= softmax(\mathbf{h}\mathbf{W^2} + \mathbf{b^2}) \\ \mathbf{h} &= g(\mathbf{x}\mathbf{W^1} + \mathbf{b^1}) \\ \mathbf{x} &= [v(w_1); v(w_2); ...; v(w_k)] \\ v(w) &= \mathbf{E}_{[w]} \end{aligned} \tag{2.8}$$

$w_i \in V \quad \mathbf{E} \in \mathbb{R}^{|V| \times d_w} \quad \mathbf{W^1} \in \mathbb{R}^{k \cdot d_W \times d_{hid}} \quad \mathbf{b^1} \in \mathbb{R}^{d_{hid}} \quad \mathbf{W^2} \in \mathbb{R}^{d_{hid} \times |V|} \quad \mathbf{b^2} \in \mathbb{R}^{|V|}$

V is a finite vocabulary of size $|V|$, and includes unique symbols to represent unknown words and to mark the beginning and ending of sentences.

To train this kind of model, the examples are $k$grams from the corpus, using $k-1$ words as features and the last word as the target label for classification. The hidden layers of the models are responsible for finding informative word combinations. As traditional language models, they are able to back-up to smaller $k$grams and to skip words if needed, in a context-dependent way. The model is trained with cross-entropy loss or approximations of it,[6] as it requires the use of a *softmax* operation that is costly if $|V|$ is large: the softmax at the output layer requires an expensive matrix-vector multiplication with the matrix $\mathbf{W^2} \in \mathbb{R}^{d_{hid} \times |V|}$, followed by $|V|$ exponentiations.
Compared to traditional language models, this model achieves better perplexities and can scale to much larger orders, because parameters are associated with individual words, and not with $k$grams. Also, it is able to generalise because unseen sequences can get high probability if they

[6] A comparison of techniques for dealing with large output vocabularies can be found in (Chen et al., 2016).

are composed of words that are similar to words composing already seen sentences (in terms of having a nearby representation). On the other side, prediction becomes more expensive, and using large vocabularies can become prohibitive due to the use of the *softmax* function.

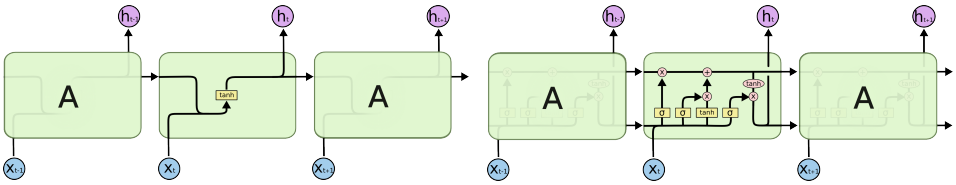
2.4.2.2 RNNs and LSTMs for language modeling

Recurrent Neural Networks (RNN) allow representing arbitrarily sized sequential inputs in fixed-size vectors, while paying attention to the structure of the inputs. Aligning the positions of input symbols to steps in computation time, they generate a sequence of hidden states h_t, as a function of the previous hidden state h_{t1} and the input for position t. In particular, RNNs with gated architectures such as the Long-Short Term Memory network (LSTM; Hochreiter and Schmidhuber, 1997) excel at capturing statistical regularities in sequential inputs, are capable of learning long-term dependencies and are more resilient to the vanishing and exploding gradient problem.

All RNNs are composed of a chain of NN modules. In standard RNNs, the modules have a very simple structure, such as a single $tanh$ layer (Figure 2.7a). In LSTMs, the module is composed of four interconnected NN layers instead of only one (Figure 2.7b). The cell state, depicted in the figure as a horizontal line at the top, is propagated through the entire chain with only some minor linear interactions. The LSTM can remove or add information to the cell state, a mechanism that is carefully regulated by structures called "gates", which optionally let information through.

There are three different gates in an LSTM cell: a forget gate, an input gate, and an output gate, each one composed of a sigmoid neural net layer determining how much of each component should be let through and of a pointwise multiplication operation. The forget gate decides which information needs attention and which can be ignored. The input gate determines what new information will be stored in the cell state. And finally, the output gate determines the value of the next hidden state.

The first language model based on RNNs and LSTMs were presented in (Mikolov et al., 2010) and (Graves, 2013)), respectively, but arguably the



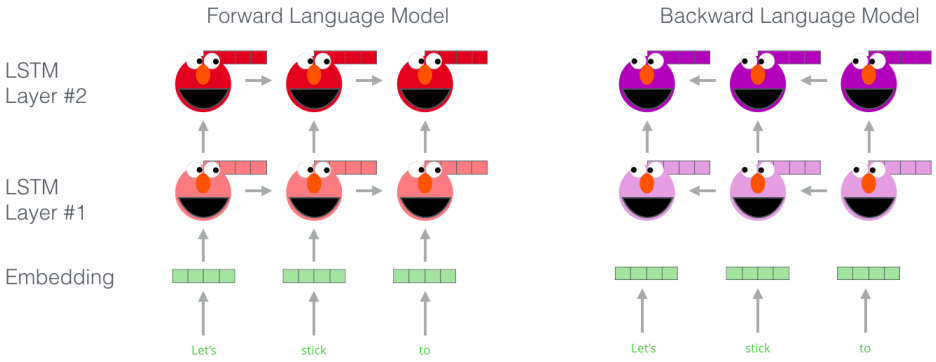
(a) RNN (b) LSTM

Figure 2.7: Modeling language with RNNs and LSTMs. Each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned NN layers. Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.
Images source: https://colah.github.io/posts/2015-08-Understanding-LSTMs

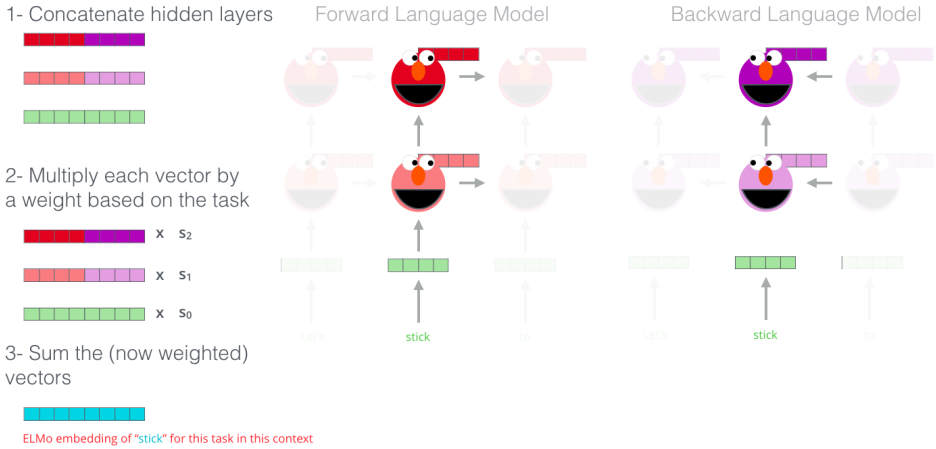
most well-known LSTM-based language model is ELMo (Peters et al., 2018b), illustrated in Figure 2.8. ELMo is mainly a bidirectional LSTM pretrained with a language modeling task, and therefore the training can be easily scaled and applied to new domains and languages. The model leverages the representations from a forward and a backward LSTM networks to generate contextualised word embeddings, concatenating the hidden layers and multiplying each resulting vector by a task-specific weight, generating the final contextualised embedding by summing the resulting vectors. Thus, when generating the embedding of each word, the model does not only have a sense of the next word, but also of the previous word. ELMo can be easily integrated into existing models by directly using the contextual embeddings as *frozen* features fed to a target model. It significantly improved the state of the art across a wide range of challenging NLU tasks.

2.4.2.3 The Transformer

Until the publication of the Transformer (Vaswani et al., 2017), the dominant language models were based on complex recurrent networks. While these models offered a better performance than their predecessors, they had parallelisation issues and were costly to train. Also, the main issue

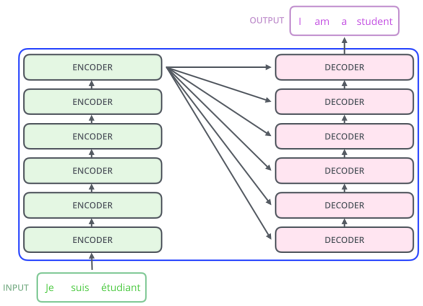


(a) Step 1. Feeding the input into the bidirectional LSTM network.

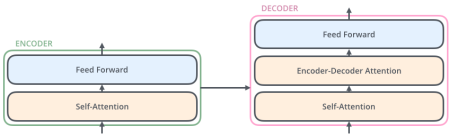


(b) Step 2. Generating the contextualised embedding by concatenating the hidden states and initial embedding, and then performing a weighted summation.

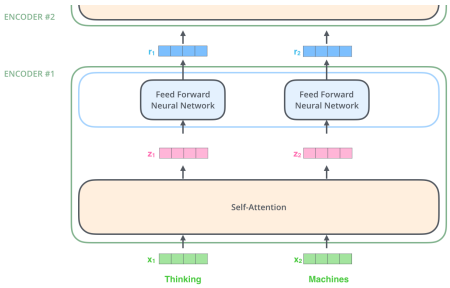
Figure 2.8: ELMo embeddings. Example: embedding of “stick” in “Let’s stick to”. Images source: https://jalammar.github.io/illustrated-bert



(a) Original architecture: stack of six encoders and a stack of six decoders.



(b) Encoder and decoder components.



(c) Words flow through the encoders.

Figure 2.9: Overview of the Transformer architecture.
Images source:
https://jalammar.github.io/illustrated-transformer

with RNNs and LSTMs remained: they do not capture well *long-term dependencies* because they tend to *forget* what was learnt if the sentences get too long. To alleviate these limitations, the Transformer relies exclusively on attention mechanisms, getting rid of recurrence entirely.

The Transformer is at the core of the vast majority of current state-of-the-art NLU systems. In what follows we will detail its architecture and training procedure through an example application to machine translation.[7] The Transformer is composed of a stack of encoders and a stack of decoders[8], as depicted in Figure 2.9a. All the encoders are composed of two sublayers. First, the encoder's input flows through a self-attention layer that helps the encoder look at other words in the input sentence as

[7] Source: https://jalammar.github.io/illustrated-transformer

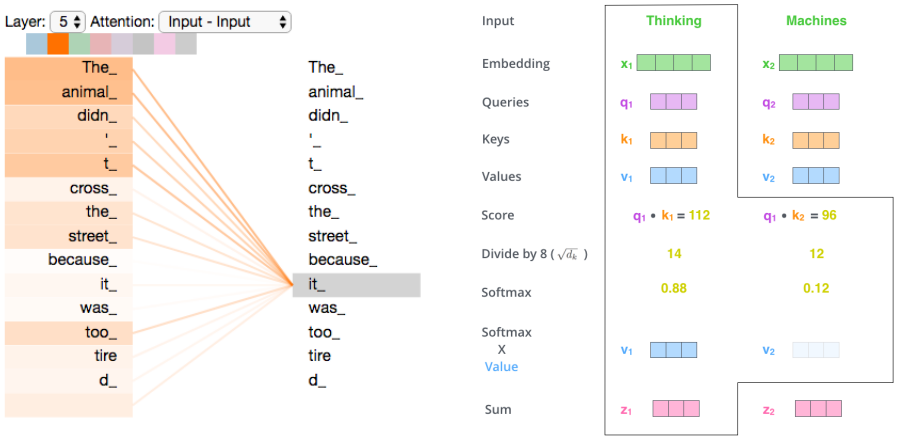
[8] The original model is composed by a stack of 6 encoders and a stack of six decoders.

it encodes a specific word. The outputs of this layer are then fed to an FFNN. Similarly, the decoder also has these two layers, but uses a self-attention layer between them to focus on relevant parts of the input sentence (see Figure 2.9b). First, each word is embedded into a vector of size 512, and each vector flows through each of the two layers of the encoder, so all the encoders receive a list of vectors each of the size 512: the word embeddings in the bottom encoder, and the output of the encoder that is directly below for the rest. As depicted in Figure 2.9c, the word vector in each position flows through its own path in the encoder, first through the self-attention layer that captures the dependencies between the flows, then into a FFNN that sends out the output upwards to the next encoder.
Self-attention allows the model to look at other positions in the input sequence for clues that can help lead to a better encoding for each word. For example, let us assume we want to translate the sentence *"The animal didn't cross the street because it was too tired"*. What does *"it"* refer to, the animal or the street? When the model is processing the word *"it"*, self-attention allows the model to associate *"it"* with *"animal"*, as shown in Figure 2.10a. The calculation of the self-attention is comprised of six steps, illustrated in Figure 2.10b:

1. Create three vectors from each of the encoder's input vectors by multiplying the embeddings by three matrices that are trained during the training process, generating a *Query vector* $\mathbf{q}$, a *Key vector* $\mathbf{k}$, and a *Value vector* $\mathbf{v}$, all of them of size 64.

2. Calculate a score that determines how much focus to place on other parts of the input sentence as we encode a word at a certain position. The score is calculated by taking the dot product of the Query vector with the Key vector of the respective word that we are scoring: for the word in position 1, the first score would be the dot product of $\mathbf{q1}$ and $\mathbf{k1}$, the second score would be the dot product of $\mathbf{q1}$ and $\mathbf{k2}$, and so on.

3. Divide the scores by 8 (the square root of the dimension of the key vectors – 64).



(a) When the model is processing the word *“it”*, self-attention allows the model to associate *“it”* with *“animal”*.

(b) Self-attention calculation for the example *Thinking Machines*.

Figure 2.10: Self-attention example (a) and calculation (b).
Images source:
https://jalammar.github.io/illustrated-transformer

4. Pass the result through a softmax operation that normalises the scores so they are all positive and add up to 1, determining how much each word will be expressed at this position.

5. Multiply each Value vector by the softmax score, to keep intact the values of the word(s) we want to focus on, and drown-out irrelevant words.

6. Sum up the weighted Value vectors to generate the output of the self-attention layer at this position (for the first word). The resulting vector is finally fed to the FFNN.

Notice that in the actual implementation this calculation is done in matrix form for faster processing.

Instead of a single self-attention head, the Transformer uses a mechanism called *multi-headed* attention, which expands the model’s ability to focus

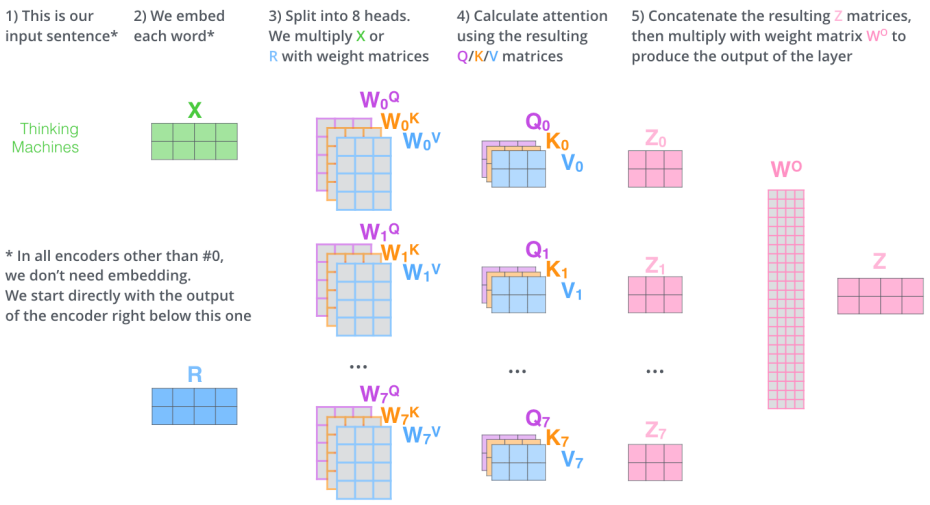


Figure 2.11: Multi-head self-attention calculation.
Image source:
https://jalammar.github.io/illustrated-transformer

on different positions in a sentence and gives the attention layer multiple *representation subspaces*. Thus, the calculation procedure takes place eight different times with different weight matrices, generating eight different $\mathbf{Z}$ matrices that are condensed into a single matrix by multiplying them by an additional weights matrix $\mathbf{W^O}$. The complete multi-headed self-attention calculation is shown in Figure 2.11.

Finally, to account for the order of the words in the input sequence, the Transformer adds a positional vector to each input embedding. These positional vectors follow a specific pattern learnt by the model. Adding them to the embeddings provides meaningful distances between the embedding vectors once they are projected into Query/Key/Value vectors and during dot-product attention. Also, each sub-layer in each encoder and each decoder has a residual connection around it, followed by a layer-normalisation step. The complete architecture is shown in Figure 2.12.

As shown in Figure 2.13a, to generate the output, the encoder starts by processing the input sequence. The output of the top encoder is then

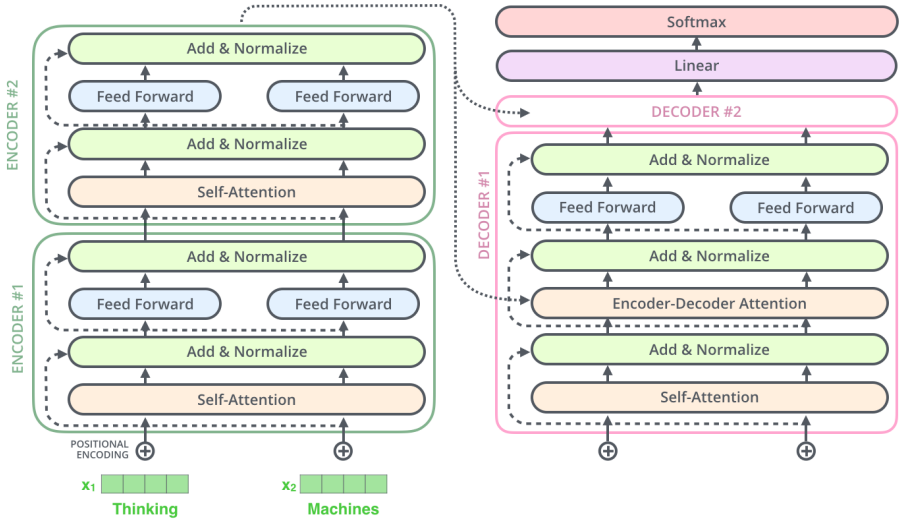
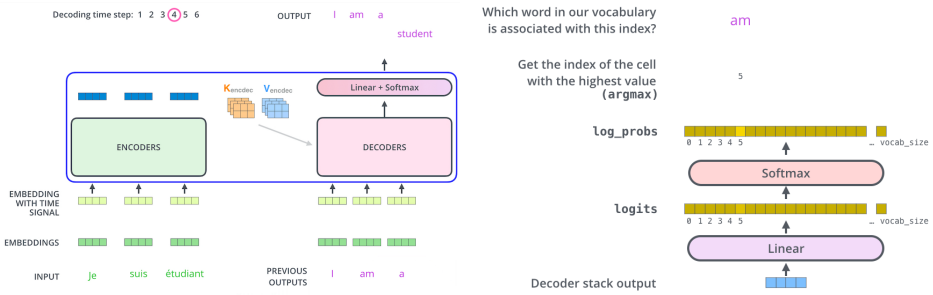


Figure 2.12: Detailed Transformer architecture, with positional embeddings and residual connections around encoders and decoders.
Image source: https://jalammar.github.io/illustrated-transformer

transformed into a set of attention vectors $\mathbf{k}$ and $\mathbf{v}$ that are used by each decoder in its *encoder-decoder attention* layer, helping the decoder focus on appropriate places in the input sequence. The process is repeated until a special symbol is reached indicating that the transformer decoder has completed its output. Importantly, the output of each step is fed to the bottom decoder in the next time step, adding a positional encoding to indicate the position of each word. Future positions are set to $-inf$ before the softmax step in the self-attention calculation, so that the self-attention layer is only allowed to attend to earlier positions in the output sequence. The *encoder-decoder attention* layer works just like multi-headed self-attention, except that it creates its Queries matrix from the layer below it, and takes the Keys and Values matrix from the output of the encoder stack. To generate the next word, the output of the decoder stack is fed to a linear layer of the size of the vocabulary, which is followed by a softmax layer to choose the final word (Figure 2.13b).



(a) (b)

Figure 2.13: Transformer output generation. a) after processing the input sequence, the output of the top encoder is transformed into a set of attention vectors $\mathbf{K}$ and $\mathbf{V}$ that are used by each decoder in its *encoder-decoder attention* layer, repeating the process and feeding the output of each step to the bottom decoder in the next time step, until a special symbol is reached; b) to generate the next word, the output of the decoder stack is fed to a linear layer of the size of the vocabulary, followed by a softmax layer to choose the final word.
Images source: https://jalammar.github.io/illustrated-transformer

The output of the softmax layer is in fact a probability distribution over the vocabulary, and thus to train the network we can use cross-entropy to compare this distribution with the perfect expected distribution, in which the following word will have a probability of 1, and all other words a probability of 0. The training procedure just described, selecting the word with the highest probability from that probability distribution and throwing away the rest, is called *greedy decoding*.[9]

2.4.3 Evaluation of language models

Two different approaches are used to evaluate and compare language models: extrinsic evaluation and intrinsic evaluation. Extrinsic evalua-

[9] Alternatively, we could use beam search, keeping at all times N partial solutions in memory. For example, we could hold on to the top 2 words, and then in the next step run the model 2 times, each time assuming that the previous position corresponds to one of the words, keeping the version that generates the lower error.

tion evaluates the models by applying them in an higher-level task, such as machine translation, and looking at their final loss or performance. It allows us to compare different models, as we can directly observe how they affect the task that we are interested in, but can be computationally expensive as it requires training the complete system.
On the other hand, intrinsic evaluation aims at finding a metric to evaluate the language model itself. A commonly used intrinsic evaluation metric is *perplexity*, an information theoretic metric that measures how well a probability model predicts a sample. Perplexity is defined as the inverse probability of the test set, normalised by the number of words in the test set (N):

$$\begin{aligned} PP(W) &= \frac{1}{P(w_1, w_2, ..., w_N)^{\frac{1}{N}}} \\ &= \sqrt[N]{\frac{1}{P(w_1, w_2, ..., w_N)}} \end{aligned} \tag{2.9}$$

Since we are taking the inverse probability, a model with lower perplexity is better because it assigns a higher probability to the unseen test corpus. Intuitively, we can interpret perplexity as the weighted branching factor: a perplexity of 100 means that whenever the model is trying to guess the next word it is as confused as if it had to pick a word between 100 words. While the perplexity is a good indicator of the quality of a language model, it is important to notice that improvements in perplexity do not necessarily imply improvements in higher-level tasks when embedding the model. Therefore, perplexity is a good metric for comparing different models in their ability to pick-up regularities in sequences of words, not for assessing progress in NLU tasks (Goldberg, 2017).

2.5 Pretrained Transformer-based Language Models

As claimed by its developers, the publication of BERT did indeed mark the beginning of a new era in NLP, inducing a paradigm shift in the way

NLP models were built. In a short period of time, pretrained word embeddings, such as Word2Vec and Glove, evolved to pretrained language models able to provide contextualised embeddings (e.g., ELMo), and shortly after the field shifted from initialising the input layer of a model with pretrained word embeddings to initialise an entire model architecture with pretrained weights, to be further fine-tuned in specific NLP tasks. The new approach to building NLP systems proved to be highly effective and quickly became the state of the art in many tasks. Since then, many new models have been developed and made publicly available, all of them with the Transformer at their core. In what follows, we review the different families of pretrained Transformer-based language models according to their architectures and pretraining objectives.

2.5.1 Sequence-to-sequence models

Sequence-to-sequence models are language models that rely on encoder-decoder architectures, that is, they are composed of an encoder and a decoder that are trained together. These models aim at transducing a sequence into another sequence. The encoder takes input sequences, e.g. sentences written in Spanish, and maps them to a high-dimensional representation. The decoder converts the high-dimensional representations into other sequences, e.g. sentences written in English.
Their most natural applications are language translation, summarisation and question answering, but they are often applied to other tasks by transforming them into sequence-to-sequence tasks. The original Transformer (Section 2.4.2.3; Vaswani et al. 2017), BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) are the most well-known examples.

2.5.2 Autoregressive models

Autoregressive models are language models that rely on the decoder of the original Transformer and aim at generating predictions by utilising its previous predictions. These models are pretrained on the original language modeling task: predicting the next token based on all the previous

ones. A mask is used on top of the full sentence to prevent the attention heads from seeing what is after the current position. Even though they can be used on many tasks, they are typically used for text generation. The most well-known example of autoregressive model is the Generative Pre-trained Transformer series: GPT (Radford et al., 2018), GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020), but there are many others, such as CTRL (Keskar et al., 2019), Transformer-XL (Dai et al., 2019), Reformer (Kitaev et al., 2020) and XLNet (Yang et al., 2019).

2.5.2.1 GPT-3

GPT-3 (Brown et al., 2020) is an autoregressive language model with 175 billion parameters (10x more than any previous non-sparse language model), trained on a mixture of datasets containing a total of 499 billion tokens: an improved version of the Common Crawl dataset (Raffel et al., 2020), WebText2 (Radford et al., 2019), two internet-based books corpora (Books1 and Books2) and English-language Wikipedia. GPT-3 is namely the most powerful language model in the market, and is currently being successfully applied to search, conversation, text completion, and many other advanced AI tasks.[10]

OpenAI conversion to a for-profit institution. GPT-3's builder, OpenAI, was initially founded as a non-profit initiative in 2015. In 2019, the company refused to release GPT-2 claiming that the model would help perpetuating fake news, and ended up releasing a version of the model that was 8% of the original size. That same year the company has been restructured to become for-profit, and in 2020, Microsoft announced that they had exclusive licensing of GPT-3 for Microsoft's products and services following a multi-billion dollar investment in OpenAI. The agreement permits OpenAI to offer a public-facing API (Application Programming Interface) such that users can send text to GPT-3 to receive the model's output, but only Microsoft has access to GPT-3's underlying model.

[10] See, for example, https://gpt3demo.com

2.5.3 Autoencoding models

Autoencoding models are language models that rely on the encoder of the original Transformer and aim at learning a representation (encoding) for the input data. These models are pretrained by corrupting the input tokens in some way and training the model to reconstruct the original sequence. They get access to the full input sequence, and usually build a bidirectional representation of the whole sequence.

Autoencoding models can be used in many applications, but they are most naturally used in sentence and token classification. The most well-known example of autoencoding model is BERT (Devlin et al., 2019a), from which many other models have been derived, such as ALBERT (Lan et al., 2020), RoBERTa (Liu et al., 2019b), XLM-R (Conneau et al., 2020), DistilBERT (Sanh et al., 2019) and FlauBERT (Le et al., 2020a). Other examples are XLM (Conneau and Lample, 2019), ELECTRA (Clark et al., 2020) and Longformer (Beltagy et al., 2020).

A combination of factors, ranging from their outstanding performance to their availability and facility of use, has made BERT and RoBERTa-based pretrained models extremely popular, with millions of downloads per month.[11] Indeed all the experiments comprised in this thesis will have one or more of these models at the core, and in what follows we will review them, along with other widely-used models derived from them.

2.5.3.1 BERT

BERT (Devlin et al., 2019a) stands for Bidirectional Encoder Representations from Transformers, and it is basically a trained Transformer Encoder stack. There are two versions of the model: the *base* version has 12 layers, 768 hidden units and 12 attention heads, while the *large* version has 24 layers, 1024 hidden units and 16 attention heads.

[11] E.g. BERT-base had 19,657,859 downloads in October 2021 from HuggingFace. Source: https://huggingface.co/bert-base-uncased

Pretraining procedure. BERT is trained on the BooksCorpus (800M words, Zhu et al. 2015) and English Wikipedia (2,500M words). In contrast to OpenAI GPT (Section 2.5.2.1), which uses a left-to-right Transformer, and ELMo (Section 2.4.2.2), which uses the concatenation of independently trained left-to-right and right-to-left LSTMs, BERT uses a bidirectional Transformer, as shown in Figure 2.14. To be able to condition BERT's representation on both left and right context, without allowing each word to indirectly see itself in a multi-layered context, BERT uses a MLM objective, traditionally known as Cloze task (Taylor, 1953), consisting on masking 15% of the input tokens at random, and then asking the model to predict the masked tokens (see Figure 2.15a). To make the prediction, the final hidden vectors corresponding to the masked tokens are fed into a softmax layer over the vocabulary, as in a standard language model. However, when applied to downstream tasks the model will not encounter *[MASK]* tokens, and to mitigate this mismatch between training and fine-tuning, the masking procedure is actually a bit more complex: if the i-th token is chosen, 80% of the time it is replaced by the *[MASK]*, 10% of the time by a random token, and 10% of the time it is left unchanged. Additionally, to make the model better at handling relationships between multiple sentences, as required for tasks such as Question Answering (QA), BERT is also pretrained with a NSP objective: given two sentences, the model must predict whether the second sentence is likely to follow the first sentence or not (see Figure 2.15b).

Input representation. BERT is able to represent both a single sentence and a pair of sentences. The first token of every sequence is a special classification token [CLS], whose final hidden state is used as the aggregate sequence representation for classification tasks. Sequence pairs are packed together into a single sequence, separated with a special token *[SEP]*. It is important to notice that encoding a concatenated text pair with self-attention effectively includes bidirectional cross attention between the two sentences. As depicted in Section 2.5.3.1, the input embeddings are the sum of three different learned embeddings: 1) the token embeddings, WordPiece embeddings (Wu et al., 2016) with a 30k

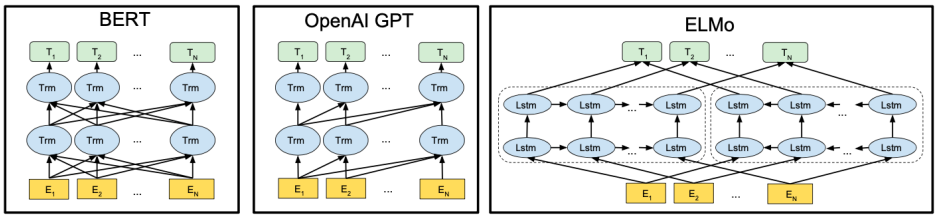
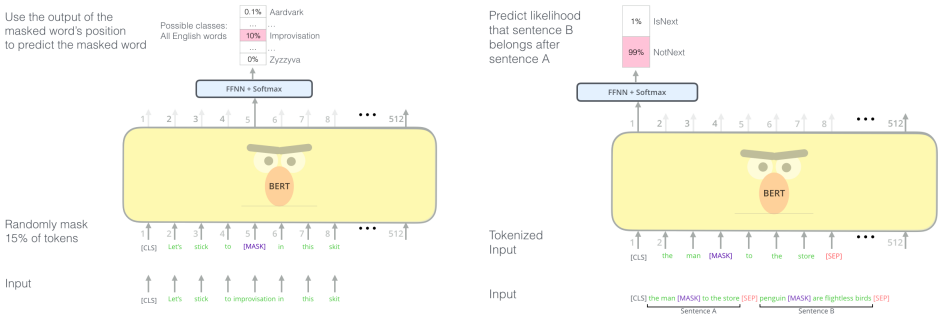


Figure 2.14: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to- left LSTMs to generate features for downstream tasks.
Source: Devlin et al. (2019a).



(a) Masked Language Model

(b) Next Sentence Prediction

Figure 2.15: BERT pretrainig procedure.
Source: https://jalammar.github.io/illustrated-bert

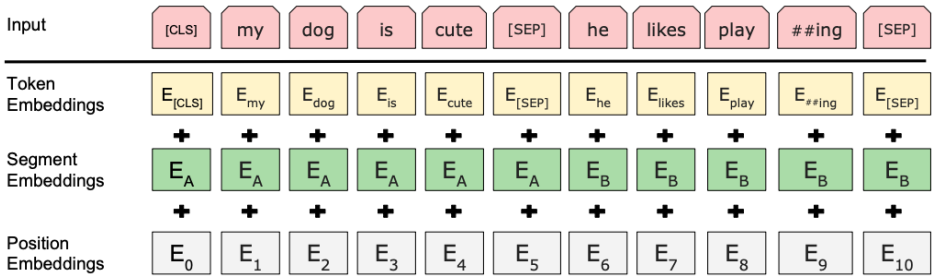


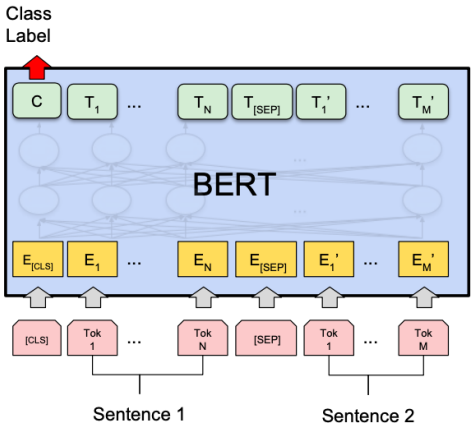
Figure 2.16: BERT input representation.
Image source: Devlin et al. (2019a).

token vocabulary; 2) the segmentation embeddings, indicating whether it belongs to sentence A or sentence B; and 3) the position embeddings, accounting for the token position in the input sequence.

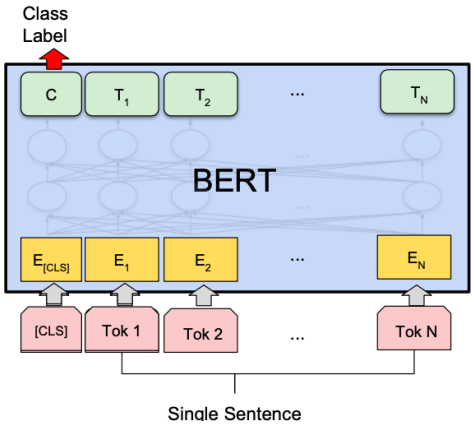
Output representation. BERT output includes two representations: 1) the hidden states, that is, the outputs from each layer; and 2) the pooler output. The *pooler* is a component that applies a linear transformation to the last hidden state of the [CLS] token, and it is trained while using the Next Sentence Prediction (NSP) strategy. The pooler transforms the output shape from [batch_size, seq_length, hidden_size] to [batch_size, hidden_size], and the resulting representation is considered a representation of the complete sentence.

Domain-specific pretraining. Optionally, in order to improve the performance of the model on new, specific domains, a second pretraining is used before fine-tuning for an specific task (Li et al., 2020; Gururangan et al., 2020; Kang et al., 2020; Gu et al., 2020).

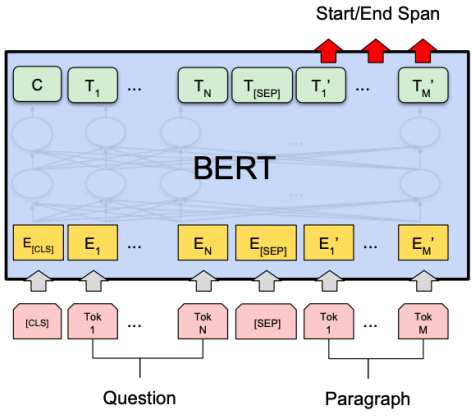
Fine-tuning. The Transformer self-attention mechanism allows BERT to easily model many downstream tasks involving single sentences and sentence pairs, as depicted in Section 2.5.3.1. For each task, the task-specific inputs and outputs are fed into BERT, and all of its parameters are then fine-tuned in an end-to-end manner.



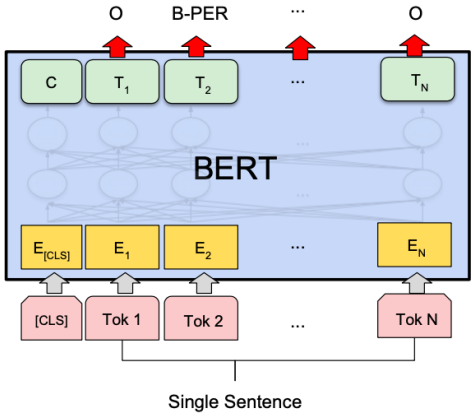
(a) Sentence pair classification taks: MNLI, QQPT, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single sentence classification tasks: SST-2, CoLA

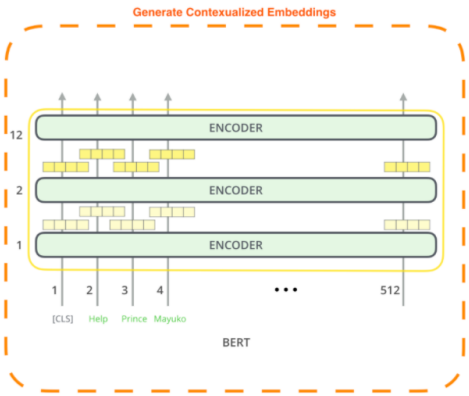


(c) Question answering tasks: SQuAD v1.1

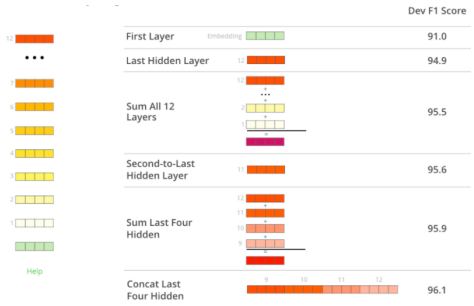


(d) Single sentence tagging tasks: CoNLL-2003 NER

Figure 2.17: Illustrations of fine-tuning BERT on different tasks.
Images source: Devlin et al. (2019a).



(a) The output of each encoder layer along each token’s path can be used as a feature representation of that token.



(b) Which embeddings combination is best is task-dependant. E.g. different embedding combinations for the word ‘Help’ in ‘Help prince Mayuko’, in CoNLL-2003 NER task.

Figure 2.18: BERT for feature extraction.
Images source: https://jalammar.github.io/illustrated-bert

BERT for feature extraction. Just like ELMo, BERT can also be used to extract contextualised word embeddings to be fed into existing models. Indeed, the output of each encoder layer along each token’s path can be used as a feature representation of that token, and therefore different layers or combination of layers can be used to represent each token. Which combination is best is a task-dependant decision, as shown in Section 2.5.3.1.

2.5.3.2 RoBERTa

Liu et al. (2019b) present a replication study of the pretraining procedure of BERT, measuring the impact of many key hyper parameters and training data size. They show that BERT was significantly undertrained, and present RoBERTa (Robustly optimised BERT approach), a new model that counts with several design and training improvements with respect to BERT. Specifically, they 1) train the model longer than BERT (4-5 times more training time), with bigger batches over more data (16 GB BERT

data + 144 GB additional data); 2) remove the next sentence prediction objective; 3) train on longer sequences; and 4) dynamically change the masking pattern applied to the training data.
RoBERTa achieves state-of-the-art results on GLUE (Wang et al., 2018), RACE (Lai et al., 2017) and SQuAD (Rajpurkar et al., 2018a), without multi-task fine-tuning for GLUE or additional data for SQuAD.

2.5.3.3 Multilingual models

The outstanding performance of pretrained Transformer-based language models for English has sparkled the development of models for other languages, such as FlauBERT (Le et al., 2020b) and CamemBERT (Martin et al., 2020) for French, BERTje (Delobelle et al., 2020) for Dutch, FinBERT (Rönnqvist et al., 2019) for Finnish, BERTeus (Agerri et al., 2020) for Basque, BETO (Cañete et al., 2020) for Spanish, AfriBERT (Ralethe, 2020) for Afrikaans, IndicBERT (Kakwani et al., 2020) for Indian, etc.
However, training language-specific models requires a large amount of data and computational resources that may not be available for all language and researchers, and this barrier limits recent advances in NLP to only a few high resource languages (Joshi et al., 2020b). In this scenario, multilingual language models (MLLMs) aim at bringing the benefit of pretrained language models to many low resource languages. A MLLM is a model pretrained using large amounts of unlabeled data from multiple languages at the same time, relying on a shared vocabulary. The intuition behind this idea is that low resource languages may benefit from high resource languages due to shared vocabulary, genetic relatedness (Nguyen and Chiang, 2017) or contact relatedness (Goyal et al., 2020). However, these models face the risk of running into what Conneau et al. (2020) refer to as “curse of multilinguality”: adding languages to the model increases the performance on low-resource languages up to a point, after which the overall performance on monolingual and cross-lingual benchmarks degrades.
In recent years, several MLLMs have been proposed, such as mBERT (Devlin et al., 2019a)), XLM (Conneau and Lample, 2019) and XLM-R

(Conneau et al., 2020). A comprehensive study of MLLMs, including a thorough analysis of their cross-lingual and zero-shot scenarios effectiveness, can be found in (Doddapaneni et al., 2021).

2.5.3.4 Model distillations

Knowledge distillation (Buciluǎ et al., 2006; Hinton et al., 2015) is a compression technique in which a small model (the *student*) is trained to reproduce the behaviour of a larger model (the *teacher*) or an ensemble of models. The student is trained with a distillation loss over the soft target probabilities of the teacher, and the final training objective is a linear combination of the distillation loss with the supervised training loss of the teacher model.
One such model is DistilBERT (Sanh et al., 2019), a BERT distillation that has the same general architecture as BERT. The number of layers is reduced by a factor of 2, initialised from the teacher by taking one layer out of two. The token-type embeddings and the pooler are removed, and most of the operations are highly optimised in modern linear algebra frameworks. The NSP objective is not used to train the student.
Many other distilled models have been recently developed for a wide range of models, such as multilingual BERT[12] and GPT-2[13], available through commonly used libraries such as HuggingFace Transformers (Wolf et al., 2020b).

[12] https://huggingface.co/distilbert-base-multilingual-cased
[13] https://huggingface.co/distilgpt2

Chapter 3

STATE OF THE ART

In this chapter, we offer a review of relevant works related to the contents of this dissertation. First, we focus on the syntactic capabilities of pretrained language models, reviewing current methods to assess their syntactic knowledge (Section 3.1). Then, we present works analysing the relation between the syntactic knowledge of language models and the size of the data used to pretrain them (Section 3.2). Finally, we present works analysing whether the syntactic knowledge encoded in the models is affected when fine-tuning them for downstream tasks (Section 3.3).

A note on generalisation. The vast majority of the works analysing the syntactic capabilities of pretrained Transformer-based autoencoding language models focus exclusively on the original BERT, as it was the first published model, widely available in different deep learning frameworks such as PyTorch (Paszke et al., 2019), Tensorflow (Abadi et al., 2016) or MXNet (Chen et al., 2015). While it has been commonly assumed that the conclusions extracted from analysing BERT generalise to other *similar* models such as RoBERTa or mBERT, it is not clear whether this is, in fact, the case. Moreover, little is known about the differences between different BERT models, pretrained with the same data but different random seed (e.g. the models available through different libraries).

3.1 Assessing the syntactic capabilities of language models

BERT has become a default baseline in NLP, and consequently, numerous studies analyse its linguistic capabilities in general (Rogers et al., 2020; Henderson, 2020), and its syntactic capabilities in particular (Linzen and Baroni, 2020). While syntactic information is distributed across all layers (Durrani et al., 2020), it has been shown that BERT captures most phrase-level information in the lower layers, followed by surface features, syntactic features and semantic features in the intermediate and top layers (Jawahar et al., 2019; Tenney et al., 2019a; Hewitt and Manning, 2019). The syntactic structure captured by BERT adheres to that of the Universal Dependencies (Kulmizev et al., 2020); different syntactic and semantic relations are captured by self-attention patterns (Kovaleva et al., 2019; Limisiewicz et al., 2020; Ravishankar et al., 2021), and it has been shown that full dependency trees can be decoded from single attention heads (Ravishankar et al., 2021). BERT performs remarkably well on subject-verb agreement (Goldberg, 2019), and is able to do full parsing relying only on pretraining architectures and no decoding (Vilares et al., 2020), outperforming existing sequence labeling parsers on the Penn Treebank dataset (de Marneffe et al., 2006) and on the end-to-end Universal Dependencies Corpus for English (Silveira et al., 2014a). It can generally also distinguish correct from incorrect completions and robustly retrieves noun hypernyms, but shows insensitivity to the contextual impacts of negation (Ettinger, 2020). Additionally, Sachan et al. (2021a) showed that incorporating syntax information from dependency trees into pretrained models can improve task-specific transformer models.

A commonly used method to test models for the presence of a wide range of linguistic phenomena is **supervised probing** (Conneau et al., 2018; Liu et al., 2019a; Tenney et al., 2019b; Voita and Titov, 2020; Elazar et al., 2020; Lepori and McCoy, 2020), that is, training supervised models to predict properties from representations extracted from a model. Hewitt and Manning (2019)'s structural probe shows that entire syntax trees are embedded implicitly in BERT's vector geometry. Extending their work,

Chi et al. (2020) show that multilingual BERT recovers syntactic tree distances in languages other than English and learns representations of syntactic dependency labels. However, other works have criticised supervised probing methods, claiming that classifier probes can learn the linguistic task from training data (Hewitt and Liang, 2019), and can fail to determine whether the detected features are actually used (Voita and Titov, 2020; Pimentel et al., 2020a; Elazar et al., 2020). Drawing from neuroscience, Ivanova et al. (2021) argue that specific research goals play a paramount role when designing a probe and encourage future probing studies to be explicit in stating these goals.

Another commonly used method to test the knowledge of the models is the **targeted syntactic evaluation**, which incorporates methods from psycholinguistic experiments and focuses on highly specific measures of language modeling performance, allowing to distinguish models with human-like representations of syntactic structure (Linzen et al., 2016a; Lau et al., 2017; Gulordava et al., 2018; Marvin and Linzen, 2018; Futrell et al., 2019). To evaluate modern language models, Warstadt et al. (2020a) present a challenge set that isolates specific phenomena in syntax, morphology, and semantics, finding that state-of-the-art models struggle with some subtle semantic and syntactic phenomena, such as negative polarity items and extraction islands. Hu et al. (2020a) assembled a set of 34 English syntactic tests in order to assess the syntactic generalisation potential of a number of different neural LMs (LSTM, ON-LSTM, RNNG and GPT-2), finding substantial differences in syntactic generalisation performance by model architecture. The tests are accessible through the SyntaxGym toolkit (Gauthier et al., 2020a); cf. also Section 4.1.1.

A number of works also address the **cross-language assessment of models**. Hu et al. (2020b) introduces XTREME, a multi-task benchmark for evaluating the cross-lingual generalisation capabilities of multilingual representations across 40 languages and 9 tasks. They show that while XLM-R reduces the difference between the performance on the English test set and all other languages compared to mBERT for tasks such as XQuAD and MLQA, it does not have the same impact on structured prediction tasks such as PoS and NER. Mueller et al. (2020) introduces a

set of subject-verb agreement tests, showing that mBERT performs better than English BERT on Sentential Complements, Short VP Coordination, and Across a Prepositional Phrase, but worse on Within-an-Object Relative Clause, Across-an-Object Relative Clause and in Reflexive Anaphora Across a Relative Clause, and offers high syntactic accuracy on English, but noticeable deficiencies on other languages, most notably on those that do not use Latin script, as also noted by Hu et al. (2020b). Along the same lines, Rönnqvist et al. (2019) concludes that mBERT is not able to substitute a well-trained monolingual model in challenging tasks.

3.2 Relation between pretraining data size and linguistic knowledge

Several studies investigate the relation between pretraining data size and linguistic knowledge in language models. van Schijndel et al. (2019); Hu et al. (2020a); Micheli et al. (2020) find out that, given a relatively large data size (e.g., 10M words), models with less pretraining perform similarly to models with much more pretraining, concluding that model architecture plays a more important role than training data scale in yielding correct syntactic generalisations (Hu et al., 2020a). Complementary, Raffel et al. (2020) shows that performance can degrade when an unlabeled data set is small enough that it is repeated many times over the course of pretraining. In contrast, Zhang et al. (2021) argue that while relatively small datasets suffice to reliably encode most syntactic and semantic features, a much larger quantity of data is needed to master conventional NLU tasks. This discrepancy may be due to the difference in model architectures, pretraining techniques and the scaling and nature of the difference datasets.

3.3 Impact of fine-tuning on the knowledge of the models

The adaptation of pretrained language models to solve supervised tasks has become a base-line in NLP, and many recent works have focused on studying how linguistic information is encoded in the pretrained sentence representations (cf. Section 3.1). Among other information, it has been shown that entire syntax trees are implicitly embedded in the geometry of such models (Hewitt and Manning, 2019). However, even though pretrained models can be used frozen as feature extractors, they are often fine-tuned on downstream tasks, and therefore it becomes increasingly important to understand how the encoded knowledge evolves along the fine-tuning.

Few works have studied how fine-tuning affects the representations of BERT. Gauthier and Levy (2019) found a significant divergence between the final representations of models fine-tuned on different tasks when using the structural probe of Hewitt and Manning (2019), while Merchant et al. (2020) concluded that fine-tuning is conservative and does not lead to catastrophic forgetting of linguistic phenomena – which our experiments (Chapter 6) do not confirm.

Chapter 4

SYNTACTIC ABILITIES OF MONOLINGUAL AND MULTILINGUAL LANGUAGE MODELS

Transformer-based neural models such as BERT (Devlin et al., 2019a), RoBERTa (Liu et al., 2019b), DistilBERT (Sanh et al., 2019), XLNet (Yang et al., 2019), etc. are excellent learners. They have been shown to capture a range of different types of linguistic information, from morphological (Edmiston, 2020) over syntactic (Hewitt and Manning, 2019) to lexico-semantic (Joshi et al., 2020a). A particularly significant number of works study the degree to which these models capture and generalise over (i.e., learn to instantiate correctly in different contexts) syntactic phenomena, including, e.g., subject-verb agreement, long distance dependencies, garden path constructions, etc. (Linzen et al., 2016b; Marvin and Linzen, 2018; Futrell et al., 2019; Wilcox et al., 2019a). However, most of these works focus on monolingual models, and, if the coverage of syntactic phenomena is considered systematically and in detail, it is mainly for English, as, e.g., (Hu et al., 2020a). Here, we aim to extend the attention to multilingual models and to emphasize the importance to also consider

the syntactic phenomena of languages other than English when assessing the generalisation potential of a model, specially those that present syntactic phenomena that are not prominent or do not exist in English, such as determiner and adjective agreement within the noun phrase, subject pro-drop, or flexible word order.

Multilingual Transformer-based language models such as mBERT (multilingual BERT, Devlin et al., 2019a), XLM (Conneau and Lample, 2019) and XLM-R (Conneau et al., 2020), usually pretrained on more than 100 languages, proved to achieve outstanding performance on cross-lingual language understanding tasks, including on low-resource languages for which only little training data is available. However, it remains unknown whether the optimisation for multiple languages conditions the capacity of the models to generalise over syntactic structures, and how languages with syntactic phenomena of different complexity are affected. In this Chapter, we systematically assesses the syntactic generalisation potential of the monolingual and multilingual versions of BERT and RoBERTa on English and Spanish, comparing the syntactic abilities of monolingual and multilingual models on the same language (English), and of multilingual models on two different languages (English and Spanish). For English, we draw upon the SyntaxGym English targeted syntactic tests in (Hu et al., 2020a), available through the SyntaxGym toolkit (Gauthier et al., 2020a), which offers 34 English syntactic test suites designed to evaluate the syntactic generalisation capabilities of language models; for Spanish, we introduce SyntaxGymES, a novel ensemble of targeted syntactic tests in Spanish.

The remainder of the chapter is structured as follows. Section 4.1 describes the English test suites, and presents the novel Spanish SyntaxGym test suites. Section 4.2 details the models that we tested, outlines how we use them to evaluate the probability of a text sequence, and presents the results of the tests suits evaluation. Section 4.3 offers a detailed analysis of the syntactic generalisation abilities of the monolingual and multilingual versions of BERT and RoBERTa, and Section 4.4 summarises our findings.

4.1 Syntactic test suites

For English, we draw upon the syntactic tests assembled by Hu et al. (2020a), accessible through the SyntaxGym toolkit (Gauthier et al., 2020b), designed to assess the syntactic coverage of language models. It contains 34 suites, grouped into 6 different so-called *circuits*, a classification based on what is required from the models to process the targeted constructions. For Spanish, we created SyntaxGymEs[1], adapting 11 of the existing suites for English and building 15 new ones, including a whole new circuit. In what follows, we first introduce the original English SyntaxGym and then present in detail the novel SyntaxGymEs.

4.1.1 SyntaxGym for English

The tests in the SyntaxGym designed by Hu et al. (2020a) (henceforth also referred to as "English SyntaxGym") are based on the notion of *surprisal*. A sequence of words is given to a language model, which assigns a probability to each of the following candidate words. Given the syntactic properties of the considered language, some candidate words are less surprising than others, and so should be predicted by a language model. For instance, after the sequence *The cat*, the inflected word *sleeps* should be less surprising than *sleep*.

Each test consists of a list of ITEMS that vary in a controlled way according to a set of CONDITIONS determined by the experimental design. The other main component is a series of PREDICTIONS comparing surprisal values in specific regions of the items across conditions. If the relevant syntactic generalisation has been learned by the model, the predictions should hold. For example, Table 4.1 shows a single item from the Agreement test suite, composed of eight regions, for which the following conditions should hold:

$$\begin{aligned} match\_plural.matrix\_v &< mismatch\_plural.matrix\_v \\ match\_sing.matrix\_v &< mismatch\_sing.matrix\_v \end{aligned} \tag{4.1}$$

[1] SyntaxGymES has been developed in collaboration with Alba Táboas García.

Condition	intro	np_subject	that	the	embed_np	embed_vp	matrix_v	cont.
match_sing	The	author	that	the	senators	hurt	is	good
mismatch_sing	The	author	that	the	senators	hurt	are	good
match_plural	The	authors	that	the	senator	hurt	are	good
mismatch_plural	The	authors	that	the	senator	hurt	is	good

Table 4.1: Single test from SyntaxGym Agreement test suite. Conditions in Equation 4.1 must hold.

Moreover, some tests have versions with MODIFIERS, in which additional clauses or phrases have been embedded inside each item. These modifiers increase the linear distance between two co-varying items, making the task harder. Sometimes they also include a distractor word in the middle of a syntactic dependency, which can lead the models to misinterpret the dependency.

Notation . To exemplify the test suites, we follow the usual notations in linguistic literature. An asterisk ‘*’ preceding an example signals that the sentence is ungrammatical, it violates some principle or constraint. A question mark ‘?’ is used to indicate a marginal sentence, i.e., a sentence that is grammatical but very uncommon or that requires a non-straightforward interpretation. The exclamation mark ‘!’ indicates a highly difficult sentence to process for the human mind.

The test suites are arranged in terms of the following *circuits*:

Agreement. Morphosyntactic phenomena that occur when the features of an item constrain another item to adopt a specific form. This is a marginal phenomenon in English, so the original circuit only includes 3 test suites on *Subject-verb number agreement*, all of them with modifiers (Marvin and Linzen, 2018). Example:

(1) The author that the senators hurt is good.

(2) * The author that the senators hurt are good.

(3) The authors that the senators hurt are good.

(4) * The authors that the senators hurt is good.

Center embedding. Subordinate clauses that sit in the middle of their superordinate clause, creating nested dependencies. This circuit contains 2 test suites: *Center embedding* and *Center embedding with modifier*, from Wilcox et al. (2019a). Example:

(5) The painting that the artist painted deteriorated.

(6) * The painting that the artist deteriorated painted.

Garden path effects. Effects that emerge when an incorrect but locally likely parse needs to be abandoned in favor of the correct one, once a specific word appears in the sentence. Two such effects are considered in this circuit: *Main verb/reduced relative clause (MVRR)* and *NP/Z garden paths*, with respectively 2 and 4 suites, all from Futrell et al. (2018). Example:

(7) ! As the ship crossed the waters remained blue and calm.

(8) ! As the ship crossed the sea the waters remained blue and calm.

(9) As the ship crossed, the waters remained blue and calm.

(10) As the ship crossed the sea, the waters remained blue and calm.

Gross syntactic expectation. Expectation for a large syntactic structure usually induced by subordinating adverbs or conjunctions. 4 test suites on *Subordination* (from Futrell et al. (2018), 3 of them with modifiers) constitute the circuit.

(11) * As the doctor studied the book.

(12) The doctor studied the book.

(13) As the doctor studied the book, the nurse walked into the room.

(14) ? The doctor studied the book, the nurse walked into the room.

Licensing. A construction’s need for the presence of a *licensor* to allow its occurrence in a sentence. The circuit consists of 4 suites on *Negative polarity items* (2 of them with modifiers) and 6 on *Reflexive pronouns* (all of them with modifiers), also from Marvin and Linzen (2018).

(15) No teacher that the ministers hated has failed any student.

(16) ? No teacher that no ministers hated has failed any student.

(17) * The teacher that the ministers hated has failed any student.

(18) * The teacher that no ministers hated has failed any student.

Long-distance dependencies (LDDs). LDDs occur when two constituents that are syntactically related do not appear adjacent to one another, but at a longer distance from one another. The circuit includes 6 suites on *Filler-gap dependencies* (2 with modifiers and 4 addressing extraction and hierarchy) from Wilcox et al. (2018) and Wilcox et al. (2019c), and 2 suites on *Cleft structure* that were first introduced in (Hu et al., 2020a).

(19) * My neighbor told me what the dog caught the mouse in full view of the neighbors yesterday.

(20) My neighbor told me that the dog caught the mouse in full view of the neighbors yesterday.

(21) My neighbor told me what the dog caught in full view of the neighbors yesterday.

(22) * My neighbor told me that the dog caught in full view of the neighbors yesterday.

4.1.2 SyntaxGymES: SyntaxGym for Spanish

For Spanish, we expand the tests in (Hu et al., 2020a) so as to cover language-specific phenomena[2]. In this section, we detail which of the original tests we retained, which ones we modified, and which ones we

[2]SyntaxGymES has been developed in collaboration with Alba Táboas García.

added within each original circuit. A whole new circuit regarding the linear order of a sentence's basic constituents was also added, since flexibility in this respect is a characteristic that distinguishes Spanish (and other Romance languages) from English. SyntaxGymES will be published in the SyntaxGym platform http://syntaxgym.org.
Each test consists of a list of ITEMS that vary in a controlled way according to a set of CONDITIONS determined by the experimental design. A series of PREDICTIONS compare surprisal values at specific regions of the items across conditions. Some tests have versions with MODIFIERS that increase the linear distance between two co-varying items, making the task more demanding.
The test suites are arranged in terms of *circuits* of related syntactic phenomena. Each of the following sections corresponds to one of these circuits.

4.1.2.1 Agreement

Agreement is a morpho-syntactic phenomenon that occurs when the features of an item constrain another item to adopt a specific form. Unlike English, Spanish is a morphologically rich language, and as such it presents many morpho-syntactic phenomena related to agreement. For this reason, out of the six original circuits, **Agreement** was the one that underwent the most changes.
Regarding verbal agreement (constraints imposed on the verb by the subject), we adapted two existing test suites, **Subject-Verb Agreement with Object Relative Clause** and **Subject-Verb Agreement with Subject Relative Clause**, and created a new one, **Basic Subject-Verb Agreement**, in which both person and number features were taken into consideration.

Basic Subject-Verb Agreement. New suite. Spanish finite verbs in any tense/mood have six inflected forms according to person and number features. The verb's features the subject's, otherwise the result is ungrammatical.

(23) Tú cocinas
you.2SG cook.2SG

(24) * Tú cocinais/cocino/cocinan
you.2SG cook.2PL/1SG/3PL

Predictions: The surprisal at the verb region is expected to be lower when it matches the subject than in any other condition. It is also expected to be lower when at least one of the features (person or number) agrees than when both disagree.

Subject-Verb Agreement with Subject Relative Clause. Adapted from English. This test focuses on number agreement. The subject relative clause includes a *distractor* NP differing in number with the subject.

(25) El fontanero que ayudó a los albañiles
the.SG plumber that helped.3SG to thePL bricklayers
trabaja/*trabajan los sábados.
work.3SG/3PL the saturdays.
'The plumber who helped the bricklayers works/*work on saturdays.'

(26) Los fontaneros que ayudaron al albañil
the.PL plumbers that helped.3SG to.thePL bricklayer
*trabaja/trabajan los sábados.
work.3PL/3SG the saturdays.
'The plumbers who helped the bricklayer *works/work on saturdays.'

Predictions: A successful model should place higher probability to the verb agreeing with the subject (instead of the distractor) both in singular and in plural.

Subject-Verb Agreement with Object Relative Clause. Adapted from English. Equal to the previous one, but with an object relative clause.

Nominal agreement was the basis for the following 6 new test suites. All of them share the same predictions: the surprisals should be lower when

both gender and number features in the second word of the agreement relation match those in the first word. They should also be lower when only one of the features agrees than when both disagree.
As for nominal agreement (constraints that a noun's gender and number features can impose on the form of other words in the sentence), we also created several new test suites: **Determinant-Noun Agreement** simply pairs a noun with the four possible forms of the definite article (*el*, *la*, *los*, *las*), while **Adjective-Noun Agreement** pairs a noun with the four possible forms of an adjective that modifies it (we excluded articles to avoid providing extra information).

Determiner-Noun Agreement. New suite. The four possible forms of the definite article are paired with different nouns.

(27) El/*La/*Los/*Las gato
the.M.SG/*F.SG/*M.PL/*F.PL cat

Adjective-Noun Agreement. New suite. The test pairs a noun with the four possible forms of an adjective that modifies it (we used constructions without determiner to avoid providing the models with extra information).

(28) La tienda vende discos usados/*usado/*usadas/*usada
the store sells discs used.M.PL/M.SG/F.PL/F.SG
'The store sells second-hand discs.'

In addition to these two suites, we built similar ones for **Attribute Agreement** in copulative constructions, to which we added two versions with object or subject relative clauses as modifiers, and also for **Predicative Agreement** in constructions with subject or object predicative complement. The only difference here is that the two words that must agree are not adjacent anymore. In terms of predictions, the verb/noun with matching features should have a lower surprisal than the others, and the verb/noun that matches only one feature should have a lower surprisal that the one that doesn't match any.

Attribute Agreement. New suite. Here, a noun is paired with and adjective through a copulative construction. This suite has 2 versions with object or subject relative clauses as modifiers.

(29) El piso está vacío/*vacía/*vacíos/*vacías
the flat is empty.M.SG/*F.SG/*M.PL/*F.PL

Predicative Agreement. New suite. The subject or the object is paired with an adjective functioning as a predicative complement.

(30) Los niños llegaron cansados/*cansado/*cansadas/*cansada
the children arrived tired.M.PL/*M.SG/*F.PL/*F.SG
'The children arrived tired.'

4.1.2.2 Center Embedding

A center embedded clause is a subordinate clause that sits in the middle of its superordinate clause, creating nested dependencies that may be challenging for the models. For this circuit, we adapted to Spanish the two existing test suites in English, creating **Center Embedding** and **Center Embedding with PP modifier**.

Center Embedding. Adapted from English. A relative clause is center embedded after the subject of the main clause. Verb transitivity and subject-verb plausibility are used to test if the models are capable of retaining the relevant information and predicting the verbs in the correct order.

(31) La tormenta que el capitán [capeó amainó]/?[amainó capeó].
'The storm the captain [weathered abated]/?[abated weathered].'

Prediction: The surprisal of the combination of verbs should be smaller when their relative order creates a plausible sentence than when it creates an implausible one.

Center Embedding with modifier. In the version with modifier, a prepositional phrase is inserted after the subject of the subordinate clause.

4.1.2.3 Gross Syntactic State

From the four original suites in this circuit, we adapted three of them: **Subordination**, and two of its versions with modifiers, **Subordination with Object Relative Clause** and **Subordination with Subject Relative Clause**. Given a sentence that starts with a typically subordinating adverb or conjunction, these suites test the models' ability to maintain the expectation for the onset of a matrix clause for as long as the subordinate one lasts.

Subordination. Adapted from English. A sentence starting with a subordinate clause creates the expectation for the onset of a matrix clause for as long as the subordinate one lasts.

(32) ?(Mientras) ella miraba los resultados, el doctor entró en la habitación.
'While she looked at the results, the doctor entered the room.'

(33) (*Mientras) ella miraba los resultados.
'(*While) she looked at the results.'

Predictions: The surprisal for the lack of a second clause should be higher when there is a subordinating conjunction or adverb than where there is not. But having two clauses joined by a conjunction/adverb should be less surprising than their juxtaposition.

Subordination with Object Relative Clause. Adapted from English. Version of the previous suite but with a modifier.

Subordination with Subject Relative Clause. Adapted from English. Version of the previous suite but with a modifier.

4.1.2.4 Long-distance Dependencies

LDDs occur when two syntactically related groups do not appear adjacent to one another but at a longer distance from one another. Filler-gap dependencies are an example of LDDs. They occur when a phrase (the filler) is realised somewhere in the sentence, but is semantically interpreted at some other point (the gap). For this circuit, we created a **Basic Filler-Gap Dependencies** test and adapted from the original English circuit a version that includes modifiers, **Filler-Gap Dependencies with Three Sentencial Embeddings**. Embedding three sentences between filler and gap makes the task more challenging. We also adapted to Spanish the novel **Pseudo-Cleft Structures** suite introduced in (Hu et al., 2020a).

Basic Filler-Gap Dependencies. New suite, a simplified version of the existing FGD tests for English. FGDs occur when a phrase (the filler) is realised somewhere in the sentence but is semantically interpreted at some other point (the gap).

(34) Yo sé [lo que]/*que tu amigo tiró _ al suelo.
'I know what/*that your friend threw _.'

(35) Yo sé *[lo que]/que tu amigo tiró una colilla al suelo.
'I know *what/that your friend threw a cigarette butt.'

Predictions: The overt object should be more surprising when there is a filler when there is not. We also expect lower surprisal when the sentence has a filler later followed by gap than when it has a conjunction instead but the gap remains.

Filler-Gap Dependencies with Three Sentencial Embeddings. Adapted from English. It is a version of the previous test that includes a modifier (three sentential embeddings) between filler and gap. This makes the task more challenging. The predictions, though, remain the same.

Pseudo-Cleft Structures. Adapted from English. A pseudo-cleft or wh-cleft is formed by a wh-element extracting content from a relative clause joined by a copula to a constituent that provides the content requested by the wh-element. The extracted constituent can be a NP or a VP. In the VP case, the verb in the relative clause must be an inflected form of 'hacer' ('to do').

(36) Lo que tú difundiste/?hiciste fue un rumor.
'What you spread/*did was a rumor.'

(37) Lo que tú *difundiste/hiciste fue confirmar un rumor.
'What you *spread/did was confirm a rumor.'

Predictions: The surprisal should be lower for the extracted VP when the verb in the relative clause is a light verb (*hacer* – 'to do') than when it is not, but it should be higher for the extracted NP when the verb is light than when it is semantically heavier and matches the NP. In addition, the difference in the first case should be more important than in the second one. This happens because the light verb admits a wider range of objects, whereas in the first case, one of the options is syntactically incorrect.

4.1.2.5 Garden Path Effects

Garden-path effects emerge when an incorrect but locally likely parse needs to be abandoned in favor of the correct one. In the NP/Z garden path, an NP is initially interpreted as the object in a subordinate clause, but when the main verb appears, this NP should be reinterpreted as its subject. The effect can be prevented by adding a comma, but also by placing an overt object in the subordinate clause, or by substituting its verb with a purely intransitive one. These are the basis for the next two suites.
The Garden Path effect can be created by several syntactic ambiguities that differ cross-linguistically. The Main Verb/Reduced Relative garden path effect was the subject of two suites in the original English circuit, but it does not translate to Spanish, so those suites were not included in Spanish SyntaxGym.

On the other hand, the ambiguity responsible for NP/Z also holds for Spanish. Here, an NP is initially interpreted as the object in a subordinate clause when it actually is the subject of the main clause (the subordinate clause having a Zero/null object). The ambiguity can be prevented with a comma, but also by placing an overt object in the subordinate clause, as is done in **NP/Z Garden Path Effect (with Overt Object)**, or by substituting its verb with a pure intransitive verb, as is done in **NP/Z Garden Path Effect (with Intransitive Verb)**. Both suites correspond to Spanish adaptations of the two original suites regarding this effect.

NP/Z Garden Path Effect (Overt Object). Adapted from English.

NP/Z Garden Path Effect (Intransitive Verb). Adapted from English.

(38) !Mientras ella leía sus manuscritos se volaron por la ventana.
!'While she read her manuscripts went out the window.'

(39) Mientras ella [dormía]/[leía un libro]/[leía,] sus manuscritos se volaron por la ventana.
'While she [slept]/[read a book]/[read,] her manuscripts went out the window.'

Predictions: The main verb should be more surprising in the garden path condition than when the effect has been prevented either by the comma or by interfering with the verb. Moreover, the difference in surprisal should be bigger when the comma is essential to solve the garden path effect than when it is not.

4.1.2.6 Licensing

In natural language, some words or constructions need the presence of a licensor to allow their occurrence in a sentence. This happens with NPIs (Negative polarity items) and subjunctive mood, for instance.
Negative polarity items (NPIs), like *any* or *ever* in English, are examples of words that need to be licensed by negation. Since Spanish NPIs do not

function exactly in the same way, we took the original NPI Licensing test as inspiration and created two new suites: **Negative Polarity Items** and **NPIs and Polarity Agreement**.
Constructions with verbs in subjunctive mood also require the presence of a licensor. In Spanish, a verb expressing feelings (e.g. of joy, surprise, pleasantness) in the main clause, creates the expectation for subjunctive mood in the subordinate clause. This was the basis for a new test suite: **Subjunctive Mood and Verbs that Express Feeling**.

Negative Polarity Items and Polarity Agreement. New suite. In Spanish, NPIs that follow the verb (such as *nunca* 'never', *nadie* 'nobody', and *nada* 'nothing') need to be licensed by negation. This 'double negative' does not result in an affirmative, it is a sort of polarity agreement.

(40) Yo no bebo nunca/?siempre.
I NEG drink never/always
'I never drink./I don't drink always.'

(41) Yo bebo *nunca/siempre.
'I *ever/always drink.'

Predictions: We expect the surprisals in both agreeing conditions (negative-NPI, positive-PPI) to be lower than in any of the non-agreeing conditions (negative-PPI, positive-NPI).

Negative Polarity Items. New suite. NPIs also need to be in the scope of the negation to be licensed by it. This suite compares between a negative particle that "commands" the NPI and one that doesn't.

(42) Tú, como no mirabas por la ventana, *(no) has visto a
You, as NEG looked by the window, NEG have seen at
nadie.
nobody
'As you weren't looking through the window, you have *(not) seen anybody.'

(43) Tú, como mirabas por la ventana, *(no) has visto a nadie.
You, as looked by the window, NEG have seen at nobody
'As you were looking through the window, you have *(not) seen anybody.'

Predictions: The NPI should be more surprising when there isn't a negative particle that commands it, independently of the presence of another one that does not command it.

Subjunctive Mood and Verbs that Express Feeling. New suite. Feeling verbs that introduce a subordinate clause serve as licensors for subjunctive mood, whereas other type of verbs do not.

(44) Espero que mañana llueva/*lloverá.
(I)hope that tomorrow rain.SUB/will.rain.IND
'I hope it rains/*[will rain] tomorrow.'

(45) Sé que mañana *llueva/lloverá.
(I)know that tomorrow rain.SUB/will.rain.IND
'I know it [will rain]/rains tomorrow.'

Predictions: Subjunctive mood should be less surprising than indicative mood when the verb in the main clause expresses feelings. But when it doesn't, subjunctive should be more surprising than indicative mood. Moreover, subjunctive mood should also be more surprising with a feeling verb than with a non-feeling verb.
The other new suite in this circuit, **Subjunctive Mood, Negation and Belief Verbs**, relies on the fact that belief verbs can also license subjunctive mood, but only when combined with negation:

Subjunctive Mood, Negation and Belief Verbs. New suite. Belief verbs can also license subjunctive mood, but only when combined with negation.

(46) No creo que mañana llueva/*lloverá.
NEG believe that tomorrow rain.SUB/will.rain.IND

’I don’t think it rains/[will rain] tomorrow.’

(47) Creo que mañana no *llueva/lloverá.
(I)believe that tomorrow NEG rain.SUB/will.rain.IND
’I think it rains/[won’t rain] tomorrow.’

Predictions: The subordinate verb should be less surprising in subjunctive than in indicative mood when the main clause is negated. However, the contrary should hold when the subordinate clause is negated but the main one is not. In addition, subjunctive mood should be less surprising when the negation is in the main clause than when it is in the subordinate clause.

4.1.2.7 Linearisation

One of the main syntactic distinctions between languages is constituent order within the sentence. But, in addition to the canonical order in which these elements appear, languages also differ in their flexibility to alter that order. Spanish allows some flexibility, which was the basis for three new test suites.
For **Subject–Auxiliary Verb–Main Verb Linearisation**, the possibility to postpone the subject is compared with the rigidity of the relation between main and auxiliary verb, which must be adjacent and do not allow inversion:

Subject – Auxiliary Verb – Main Verb Linearisation. New suite. Subject-verb order admits inversion in Spanish but main and auxiliary verb do not and they must be adjacent.

(48) Juan ha comido. / Ha comido Juan
’John has eaten. / Has eaten John.’

(49) *Juan comido ha. / *Ha Juan comido.
’John eaten has. / Has John eaten.’

Predictions: The postposed subject should be less surprising than any of the alterations involving auxiliary and main verb. The canonical SV

order, however, should be less surprising than postposing the subject, and the difference in this case should be less important than the differences in the first two cases.
In the **Subject–Verb–Object Linearisation** test, we compare the phenomenon in affirmative versus interrogative sentences. In Spanish, word order flexibility holds for affirmative sentences, but not for interrogative ones, where subject-verb inversion is compulsory:

Subject – Verb – Object Linearisation. New test. In Spanish, word order flexibility holds for affirmative sentences but not for interrogative ones, where subject-verb inversion is compulsory.

(50) Ana compró un libro/Compró un libro Ana.
'Ann bought a book. / Bought a book Ann.'

(51) ¿Qué compró Ana? / ¿Qué Ana compró?
'What did Ana buy? / 'What Ana did buy?'

Predictions: A postposed subject in an affirmative sentence should be less surprising than lack of SV inversion in an interrogative one. The canonical SV order in the affirmative sentence, however, should be less surprising than postposing the subject, and the difference in this case should be less important than the difference in the first one.
Word order variations also appear within the NP, as captured by the **Noun-Adjective and Noun-PP Linearisation** test. Contrary to English, Spanish adjectives usually come after the noun. But again, the language allows for some flexibility and they can be swapped. This possibility, however, does not apply to other noun modifiers like prepositional phrases:

Noun-Adjective and Noun-PP Linearisation. New suite. Spanish adjectives usually come after the noun, but this order can be inverted. Other noun modifiers like prepositional phrases cannot.

(52) Construyó una [mesa robusta]/[robusta mesa].
'He built a [sturdy table]/[table sturdy].'

(53) Construyó una [mesa de madera]/*[de madera mesa].
’He built a [wooden table]/*[table wooden].’

Predictions: A PP preceding the noun should be more surprising than one following it. An adjective preceding the noun should also be more surprising than one following it, but the difference in this case should be less important than in the first one.

4.2 Targeted syntactic evaluation

4.2.1 Experimental Setup

Drawing upon the models from the HuggingFace Transformers library (Wolf et al., 2020b), we test the base cased versions of BERT and mBERT, RoBERTa and XLM-R on the English SyntaxGym and BETO (Canete et al., 2020), mBERT and XLM-R on the Spanish SyntaxGym. To run the experiments, we use the SyntaxGym toolkit (Gauthier et al., 2020a).

• **BERT** (Devlin et al., 2019b). Bidirectional Transformer trained with MLM and NSP. Monolingual (16 GB of English data from Book Corpus and Wikipedia). 30k WordPiece vocabulary. 110M parameters.

• **RoBERTa** (Liu et al., 2019b). Bidirectional Transformer trained with MLM. Monolingual (160 GB of English data). 50k BPE vocabulary. 125M parameters. Compared with BERT, RoBERTa is trained with dynamic masking (instead of static) on much more data and without NSP loss.

• **XLM-R** (Conneau et al., 2020). Bidirectional Transformer trained with MLM. Multilingual (2 TB filtered CommonCrawl data; 100 languages). 250k Sentence Piece vocabulary. 270M parameters.

• **mBERT** (Devlin et al., 2019b). Bidirectional Transformer trained with MLM and NSP. Multilingual (top 104 languages with the largest Wikipedia). 110k WordPiece vocabulary. 177M parameters.

• **BETO** (Canete et al., 2020). Bidirectional Transformer trained with MLM and NSP. Monolingual (3B words). 31k BPE vocabulary. 110M parameters.

4.2.2 Encoding unidirectional context with bidirectional models

The SyntaxGym test suites are designed from the perspective of sentence generation, i.e., with the hypothesis that if a model has correctly learned some relevant syntactic generalisation, it should assign higher probability to grammatical and natural continuations of sentences. This requires asking the models to predict the next token given a context of previous tokens, in a left-to-right generative fashion. However, BERT-based and RoBERTa-based families of models (in our case, BERT and mBERT on the one side, and RoBERTa and XLM-R on the other side) are bidirectional, they are trained with a MLM objective to predict a word given its left and right context. We follow Wang and Cho (2019)'s sequential sampling procedure to evaluate the probability of a text sequence, encoding unidirectional context in the forward direction. To compute the probability distribution for a sentence with N tokens, we start with a sequence of $N + 2$ tokens: a *begin_of_sentence* token plus $N + 1$ *mask* tokens, where the last *mask* corresponds to the *end_of_sentence* token. For each token position *i* in $[1, N]$, we compute the probability distribution over the vocabulary given the left context of the original sequence, and select the probability assigned by the model to the original word. Note that this setup allows the models to know how many tokens there are in the sentences, and therefore the results are not directly comparable with those of unidirectional models, that do not have any information regarding the length of the sequence.

For example, in an agreement test with the sentence '*The girls run fast.*', a model that has properly learned agreement should assign a higher probability to *run* than to *runs* for the third word. In order to test it, we feed the tokens sequence *[[bos] [The] [girls] [mask] [mask] [mask] [mask]]* to the model, and compare the probabilities assigned by the model to *run*

	Average SG performance	
Model	**English**	**Spanish**
BERT	77.80	—
RoBERTa	82.04	—
mBERT	77.55	72.31
XLM-R	71.84	78.50
BETO	—	67.92

Table 4.2: Average SG score by model class for the English and Spanish tests.

and *runs* for position 4.

4.2.3 Evaluation results

This section summarises the results of our experiments that aim to: 1– contrast the performance of monolingual and multilingual models on English and Spanish; and 2– provide insights on the performance of the multilingual models across languages.

Table 4.2 shows the average SyntaxGym (SG) performance of the evaluated monolingual and multilingual models on the English and Spanish SyntaxGyms. Figures 4.1 and 4.2 zoom in on the performance of the tested models with respect to specific circuits for English and Spanish respectively.

Six of the English test suites (Center Embedding, Cleft structure, MVRR, NPZ-Verb, NPZ-Object, Subordination) and five of the Spanish test suites (Attribute Agreement, Basic Subject-Verb Agreement, Subordination, Center Embedding, Basic Filler-Gap Dependencies) include tests with and without modifiers, i.e,. intervening content inserted before the critical region. Figures 4.3 and 4.4 show the models' average scores in these test suites, without modifiers (dark bars) and with modifiers (light bars), evaluating how robust each model is with respect to the corresponding content.

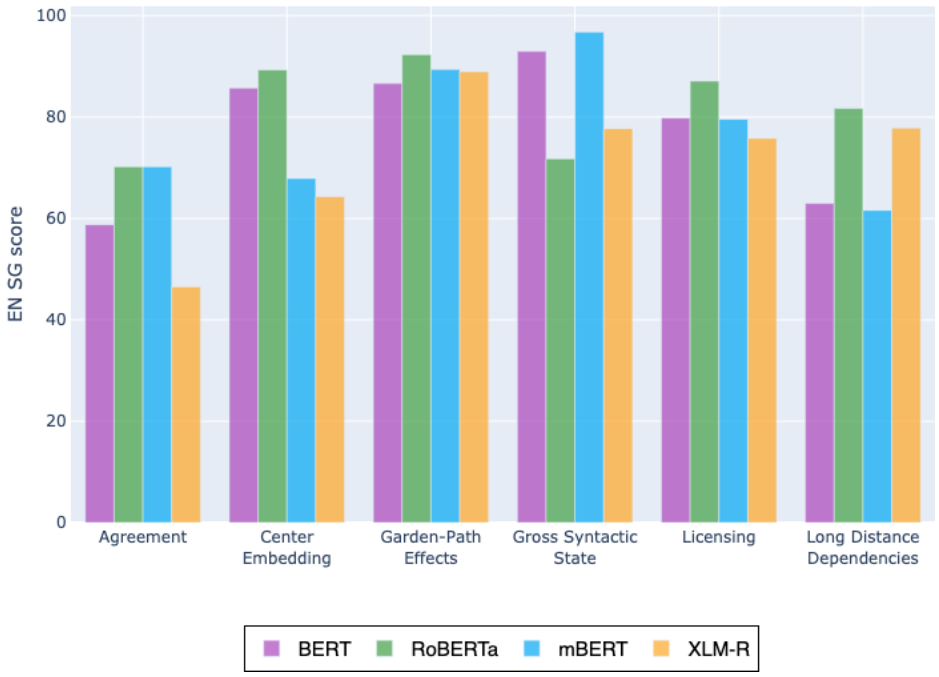


Figure 4.1: Performance accuracy across English circuits

4.3 Results analysis

Let us assess in detail the results of the experiments from above. In what follows, we compare the performance of monolingual with the performance of multilingual models and analyse the cross-language performance of multilingual models, as well as the stability of the individual models with respect to modifiers.

4.3.1 Monolingual vs multilingual models

RoBERTa shows an overall higher performance than the other models for English (Table 4.2). This is not surprising since it is trained on 10 times more data than BERT, and it has been shown to improve over BERT in many NLU tasks. However, while mBERT does not seem to lose perfor-

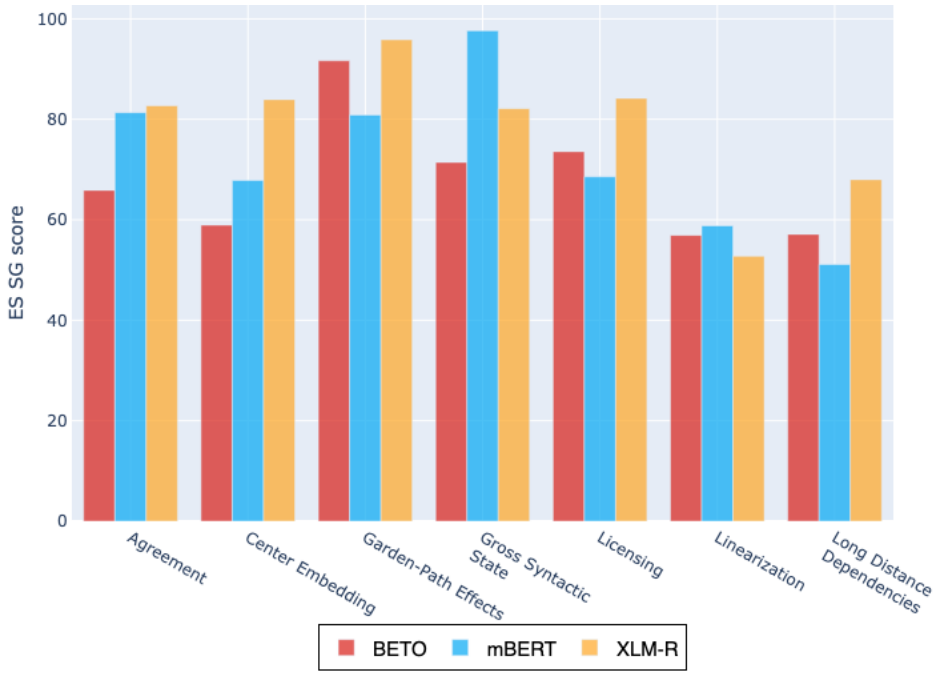


Figure 4.2: Performance accuracy across Spanish circuits

mance compared to BERT, XLM-R loses around 10 points compared to RoBERTa. As XLM-R is specifically designed to offer a more balanced performance across languages, with a special focus on low-resource languages, it appears natural that it loses some performance on high-resource languages such as English. For Spanish, the multilingual models clearly outperform the monolingual model. This is likely due to the fact that while BETO and mBERT are of comparable size and are trained with the same amount of data (16GB), BETO is only trained with a MLM objective, and mBERT is trained on MLM and NSP. On the other hand, XLM-R is also only trained on MLM, but it is trained on more than 2TB of data, 53 GB corresponding to Spanish data.

RoBERTa outperforms all other models in all the English circuits (cf. Figure 4.1), except in Gross Syntactic State, in which BERT-based models clearly outperform RoBERTa-based models, and the multilingual model

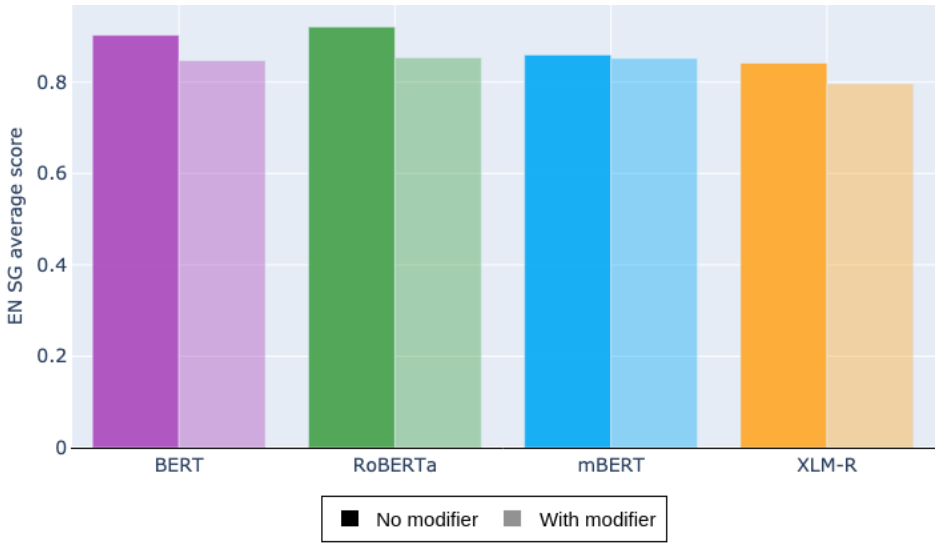


Figure 4.3: Models average English SG score in Center Embedding, Cleft structure, MVRR, NPZ-Verb, NPZ-Object and Subordination, with and without modifiers.

outperforms the monolingual one in both families. Intuitively, we believe that the NSP training objective of BERT-based models helps them to better understand the relation between two sentences, and this knowledge can also be applied to the relation between two clauses (which is the basis of the Gross Syntactic State circuit). Comparing the BERT and RoBERTa model families, it is interesting to notice that while RoBERTa outperforms XLM-R in all circuits except Gross Syntactic State, BERT only outperforms mBERT in 3 of them.

Interestingly, all models seem to struggle with Agreement in English. This observation is aligned with Mueller et al. (2020)’s hypothesis that language models learn better hierarchical syntactic generalisations in morphologically complex languages (such as, e.g., Spanish), which frequently provide overt cues to syntactic structure, than in morphologically simpler languages (such as, e.g., English). Indeed, the fact that XLM-R offers the lowest performance may be related to the fact that the model has been more exposed to more complex languages than the others. For Long Dis-

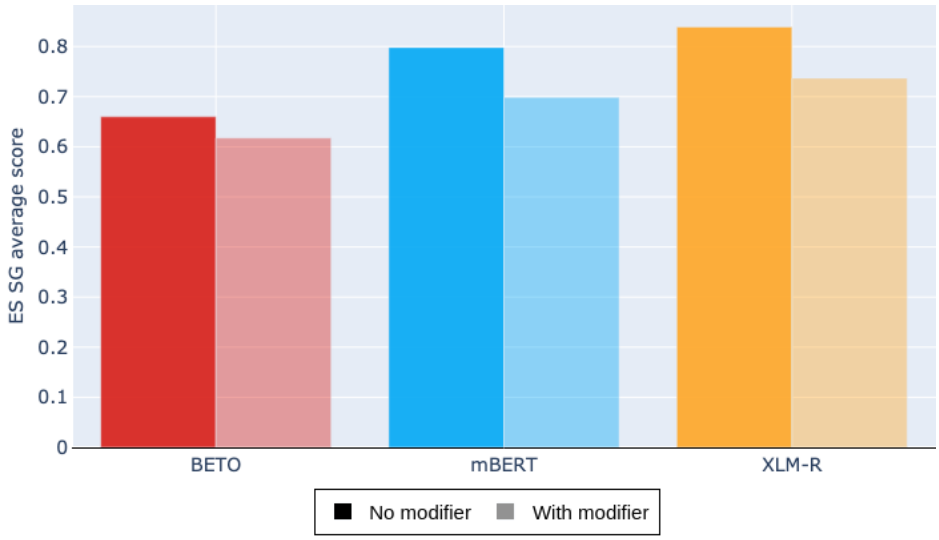


Figure 4.4: Models average Spanish SG score in Attribute Agreement, Subject-Verb Agreement, Subordination, Center Embedding and Filler-Gap Dependencies, with and without modifiers.

tance Dependencies, BERT-based models show a low performance compared to RoBERTa-based models. This might be due to the different training procedures adopted in both model families (i.e., that RoBERTa does not include the NSP task (as BERT does) and introduces dynamic masking).

On the other hand, in specific circuits for Spanish (cf. Figure 4.2) XLM-R outperforms the other two models in 5 out of 7 circuits. As observed for English, the BERT-based models struggle with the Long Distance Dependencies tests, and mBERT offers an outstanding performance in Gross Syntactic State. The monolingual model, BETO, is outperformed by mBERT in 4 out of 7 tests, and by XLM-R in all 6 out of 7 tests. As mentioned before, these differences may be related to the fact that, unlike BERT, BETO is not trained with the NSP objective; but also to the difference in training data size: 16GB for BETO vs. more than 2TB (of which 53GB of Spanish data) for XLM-R.

All models offer a low performance in the new Linearisation test for Span-

ish. A more in-depth investigation is necessary to explain this. The test has been designed with literary Peninsular Spanish in mind, and it is possible that the training data may not contain enough samples that show the targeted word order varieties, or may contain data from American Spanish sources, which may show differences in canonical word order with respect to Peninsular Spanish.

4.3.2 Cross-language multilingual models performance

As shown in Table 4.2, multilingual models do not syntactically generalise equally well in both languages. While mBERT offers a better generalisation in English, outperforming XLM-R by almost 6 points, XLM-R generalises better in Spanish, outperforming mBERT by 6 points. This observation corroborates our intuition that XLM-R sacrifices performance in high-resource languages (e.g., English, with 300GB of training data) to be able to offer a more balanced performance across languages (e.g., Spanish, with 53GB of training data).
Comparing Figures 4.1 and 4.2, we observe improvements in the Spanish tests for XLM-R in 4 out of 6 circuits, particularly noticeable in Agreement and Center Embedding, while it loses around 10 points in Long Distance Dependencies. On the other hand, mBERT also shows a big improvement in the Spanish tests in Agreement, while it loses performance in Garden Path Effects, Licensing and Long Distance Dependencies.

4.3.3 Model stability with respect to modifiers

Since modifiers increase the linear distance between the elements in a dependency structure, thus making the task more demanding, stability in this respect indicates that models have robustly learnt the appropriate syntactic generalisation and do not depend that much on adjacency. Figures 4.3 and 4.4 show the models' average scores in those test suites that have two versions: without modifiers (dark bars) and with modifiers (light bars). As was intuitively expected, all the models offer a higher performance in the tests without modifiers. While for English the multilingual models

are the less affected, for Spanish BETO seems to be more robust than the multilingual models, even though it offers a lower performance.

4.4 Insights

We assessed the syntactic generalisation potential of selected transformer-based language models on English and Spanish. We have shown that multilingual models do not generalise equally well across languages: mBERT generalises better for phenomena in English, while XLM-R does it better for phenomena in Spanish. We have also shown that the answer to the question whether monolingual or multilingual models generalise better is equally language-specific: the monolingual RoBERTa generalises better on English, while the multilingual XLM-R generalises better on Spanish. While it is possible that the multilingual abstractions captured by XLM-R become useful for morphologically rich languages such as Spanish, this difference may also be related to the difference in the amount of training data used to train BETO and XLM-R, and therefore it is possible that a monolingual model trained with a comparable amount of data could outperform the multilingual models.

The performance of all models is affected by the presence of modifiers, which shows that the complexity of the syntactic structure is still a challenge. In general, each syntactic phenomenon deserves attention. For instance, Agreement in English is hard to learn, given the scarcity of cues (especially if compared to a morphologically rich language), and so is Linearisation in Spanish.

As far as the nature of the training procedures of the models is concerned, the lack of NSP objective in the RoBERTa model family seems to harm BETO, but not XLM-R; this suggests that the performance of BETO may be improved with (much) more training data. It also seems to harm in the case of the Gross Syntactic State circuit, suggesting that RoBERTa-based models may also benefit from complementary training objectives in their pretraining procedure.

Chapter 5

IMPACT OF PRETRAINING DATA SIZE ON THE SYNTACTIC ABILITIES OF LANGUAGE MODELS

The use of unsupervised pretrained language models in the context of supervised tasks has become a widely spread practice in NLP, with Transformer-based models such as BERT (Devlin et al., 2019b) and RoBERTa (Liu et al., 2019b) achieving outstanding results in many well-known NLU benchmarks such as GLUE (Wang et al., 2018) and SQuAD (Rajpurkar et al., 2018a). Consequently, several studies investigate the types of knowledge learned by BERT, how and where this knowledge is represented and what the best methods to improve it are; see, e.g., Rogers et al. (2020). There is evidence that, among other information (e.g., part-of-speech, syntactic chunks and roles (Tenney et al., 2019b; Lin et al., 2019; Belinkov et al., 2017), morphology in general (Peters et al., 2018a), or sentence length (Adi et al., 2017)), BERT representations implicitly embed entire syntax trees (Hewitt and Manning, 2019).

Language models are traditionally assessed by information-theoretical metrics such as perplexity, i.e., the probability of predicting a word in

its context. The general wisdom is that the more pretraining data a model is fed, the lower its perplexity gets. However, while pretraining methods are very convenient, they are expensive in terms of time and resources, and large volumes of pretraining data may not always be available. This calls for a study of the impact of pretraining data size on the knowledge of the models.

In this Chapter, we explore the relation between the size of the pretraining data and the syntactic capabilities of RoBERTa by means of the MiniBERTas models, a set of 12 RoBERTa models pretrained from scratch by Warstadt et al. (2020b) on incremental sizes of raw text data ranging from 1M to 1B words. In particular:

- We draw upon the syntactic structural probes from Hewitt and Manning (2019) to determine whether the models pretrained on more data encode a higher amount of syntactic information;
- We analyse the generalisation performance of the different models using SyntaxGym (Gauthier et al., 2020b) and the targeted syntactic tests presented in (Hu et al., 2020a);
- We compare the performance of the different models on two morpho-syntactic tasks (PoS tagging and dependency parsing), and a non-syntactic task (paraphrase identification);
- We conduct a cost-benefit trade-off analysis (Strubell et al., 2019; Bhattacharjee et al., 2020) of the models training.

The remainder of the chapter is structured as follows. Section 5.1 presents the MiniBERTas models. The next three sections correspond to the different experiments: Section 5.2.2 presents the structural probing experiments, Section 5.3 presents a targeted syntactic evaluation and Section 5.4 presents a downstream tasks evaluation. Section 5.5 offers a cost-benefit analysis of the pretraining of the different models, and Section 5.6 summarises our findings.

Model Size	L	AH	HS	FFN	P
BASE	12	12	768	3072	125M
MED-SMALL	6	8	512	2048	45M

Table 5.1: Hyperparameters per model sizes. AH = number of attention heads; HS = hidden size; FFN = feedforward network dimension; P = number of parameters.

5.1 The MiniBERTas models

The MiniBERTas are a set of 12 RoBERTa models pretrained from scratch by Warstadt et al. (2020b) on 4 datasets containing 1B, 100M, 10M and 1M tokens, available through HuggingFace Transformers.[1] The datasets are sampled from Wikipedia and Smashwords – the two datasets that make up the original pretraining dataset of BERT and that are included in the RoBERTa pretraining data. For each dataset size, pretraining is run 25 times (10 times for 1B) with varying hyperparameter values; the three models with the lowest development set perplexity are released. For the smallest dataset, a smaller model size is used to prevent over-fitting. We refer to models trained on the same amount of data as a *family* of models, and models inside a family as *intra-family members* (e.g.,the *roberta-base-100M-1* model is a member of the *roberta-base-100M* family). Table 5.1 offers an overview of the hyperparameters per model size.

5.2 Structural probing

As reviewed in Section 3.1, a commonly used method to test models for the presence of a wide range of linguistic phenomena is supervised probing (Conneau et al., 2018; Liu et al., 2019a; Tenney et al., 2019b; Voita and Titov, 2020; Elazar et al., 2020; Lepori and McCoy, 2020), that is, training supervised models to predict properties from representations extracted from a model. Here, we draw upon the syntactic structural probes

[1] https://huggingface.co/nyu-mll

from Hewitt and Manning (2019) to determine whether the models pretrained on more data encode a higher amount of syntactic information.

5.2.1 Hewitt and Manning structural probe

Hewitt and Manning (2019)'s structural probes assess how well syntax trees are embedded in a linear transformation of the network representation space applying two different evaluations: Tree distance evaluation, in which squared L2 distance encodes the distance between words in the parse tree, and Tree depth evaluation, in which squared L2 norm encodes the depth in the parse tree.

Tree distance evaluation. Evaluates how well the predicted distances between all pairs of words in a model reconstruct gold parse trees by computing the Undirected Unlabeled Attachment Score (*UUAS*). It also computes the Spearman correlation between true and predicted distances for each word in each sentence, averaging across all sentences with lengths between 5 and 50 (we refer to as *DSpr.*).

Tree depth evaluation. Evaluates the ability of models to recreate the order of words specified by their depth in the parse tree, assessing their ability to identify the root of the sentence as the least deep word (*Root %*) and computing the Spearman correlation between the predicted and the true depth ordering, averaging across all sentences with lengths between 5 and 50 (we refer to as *NSpr*).

5.2.2 Probing results

We use Hewitt and Manning's syntactic structural probes to determine whether the MiniBERTa models pretrained on more data encode a higher amount of syntactic information than those trained on less data. Following the original work, we probe layer 7 of all models, as it was shown to encode most of the syntax. Results are shown in Table 5.2.

Tree distance evaluation results. The models trained with more data encode better syntactic information (as measured by the probe metrics). While *DSpr.* shows a less pronounced variability between family mem-

	Tree distance eval.		Tree depth eval.	
Model	**UUAS**	**Dspr.**	**Root %**	**Nspr.**
1b-1	70.75	78.82	83.92	85.38
1b-2	72.93	79.86	83.53	85.92
1b-3	77.23	82.66	85.13	86.87
100m-1	68.46	76.95	81.21	84.06
100m-2	70.02	78.11	81.25	84.53
100m-3	69.35	78.73	79.88	84.59
10m-1	61.48	73.19	70.88	81.65
10m-2	62.01	73.78	70.07	81.89
10m-3	60.12	72.58	67.14	80.62
1m-1	56.96	71.70	57.12	74.16
1m-2	55.78	71.33	56.56	74.74
1m-3	55.84	71.33	57.41	74.46

Table 5.2: Structural probing with Hewitt and Manning's syntactic structural probes. '1b-*' corresponds to the family roberta-base-1B, '100M-*' to roberta-base-100M, '10M-* to roberta-10M, and '1M-*' to roberta-med-small-1M.

bers, and smaller differences across families, *UUAS* shows a higher intra-family variability and bigger differences between families. Noticeably, for the *roberta-base-1B* family, there is a 7 points difference in *UUAS* between model 1 and model 3, which have a difference of only 0.09 points in perplexity, highlighting the importance of training hyperparameters for the performance of the models.

Tree depth evaluation results. As for the distance metrics, the models trained on more data show a better encoding of syntactic information. Again, the correlation shows less variability between family members and smaller differences between families, while *Root %* shows a higher intra-family variability (especially noticeable for *roberta-base-10M*).

5.3 Targeted syntactic evaluation

We test the MiniBERTas on the syntactic tests assembled by Hu et al. (2020a), accessible through the SyntaxGym toolkit (Gauthier et al., 2020b). The tests require asking the models to predict the next token given a context of previous tokens, in a left-to-right generative fashion. In order to test the bidirectional MiniBERTas models, we follow Wang and Cho (2019)'s sequential sampling procedure, described in Section 4.2.2, to encode unidirectional context in the forward direction.

5.3.1 Syntactic test suites

The tests are divided into 6 syntactic circuits, thoroughly detailed in Section 4.1.1 and briefly summarised here:

• **Agreement:** Tests a language model for how well it predicts the number marking on English finite present tense verbs. It is composed of 3 Subject-Verb Number Agreement tests from Marvin and Linzen (2018),

• **Center Embedding:** Tests the ability to embed a phrase in the middle of another phrase of the same type. The circuit is composed of 2 tests from Wilcox et al. (2019b).

• **Garden-Path Effects:** Measures the syntactic phenomena that result from tree structural ambiguities that give rise to locally coherent but globally implausible syntactic parses. The circuit is composed of 2 Main Verb / Reduced Relative Clause (MVRR) tests and 4 NP/Z Garden-paths (NPZ) tests, all from Futrell et al. (2018).

• **Gross Syntactic Expectation:** Tests the ability of the models to distinguish between coordinate and subordinate clauses: introducing a subordinator at the beginning of the sentence should make an ending without a second clause less probable, and should make a second clause more probable. The circuit is composed of 4 Subordination tests from Futrell et al. (2018).

• **Licensing:** Measures when a particular token must exist within the scope of an upstream licensor token. The circuit is composed of 4 Negative Polarity Item Licensing (NPI) tests and 6 Reflexive Pronoun Licens-

ing tests, all from Marvin and Linzen (2018).
• **Long-Distance Dependencies:** Measures covariations between two tokens that span long distances in tree depth. The circuit is composed of 6 Filler-Gap Dependencies (FGD) tests from Wilcox et al. (2018) and Wilcox et al. (2019c), and 2 Cleft tests from (Hu et al., 2020a).

5.3.2 Evaluation results

We assess the syntactic generalisation performance of the different MiniBERTas models using Hu et al. (2020a)'s test suites (cf. Section 5.3) to answer the following questions: Do models pretrained on more data generalise better? Do models with lower perplexity perform better in the syntactic tests? Do models with more pretraining or better perplexity perform better in all circuits?

Average SG Score. Figure 5.1 shows the performance of each model averaged across all 6 circuits. We observe a variability between family members, especially for *roberta-base-100M*, with a difference of 15 points between models 1 and 2. As intuitively expected, the smallest family of models, *roberta-med-small-1M*, performs clearly worse than the other families. However, it is interesting to observe that more training data does not always imply better syntactic generalisation: model *roberta-base-100M-1* performs worse than the whole *roberta-base-10M* family, and model *roberta-base-100M-2* performs better than the whole *roberta-base-1B* family.

Stability with respect to modifiers. Five of the test suites (Center Embedding, Cleft structure, MVRR, NPZ-Verb, NPZ-Object) include tests with and without modifiers, i.e,. intervening content inserted before the critical region. These additional clauses or phrases increase the linear distance between two co-varying items, making the task more difficult, and sometimes they also include a distractor word in the middle of a syntactic dependency, which can lead the models to misinterpret the dependency. Figure 5.2 shows the models' average scores on these test suites, without

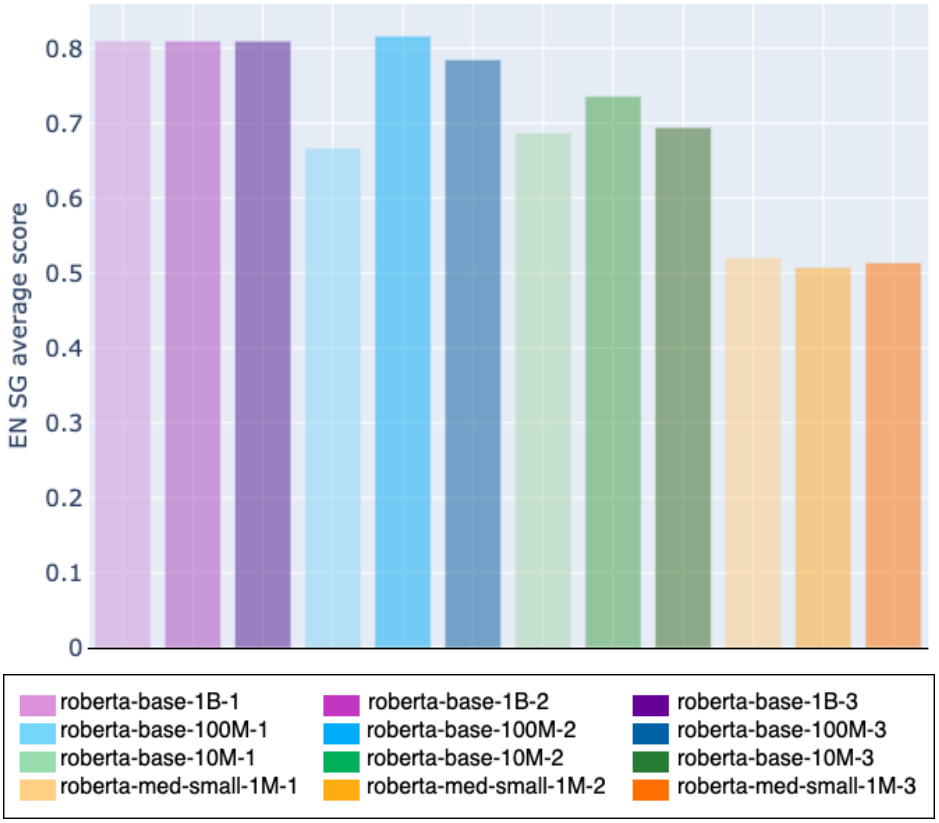


Figure 5.1: Syntactic generalisation evaluation. Average SyntaxGym score.

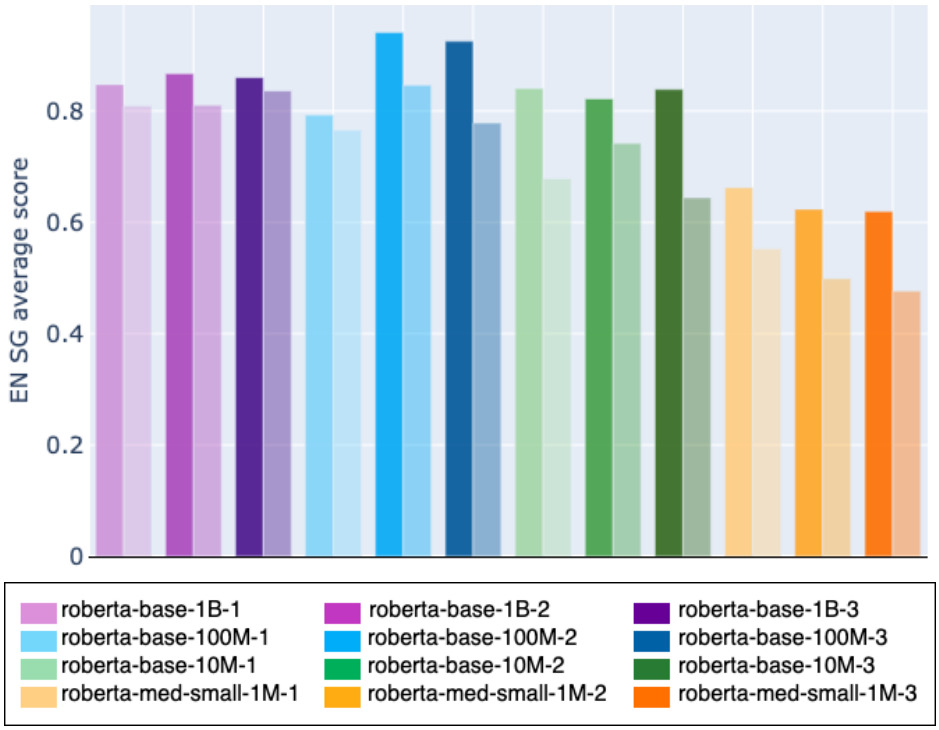


Figure 5.2: Syntactic generalisation evaluation. SyntaxGym score on Center Embedding, Cleft structure, MVRR, NPZ-Verb, and NPZ-Object, without (dark bars) and with (light bars) modifiers.

modifiers (dark bars) and with modifiers (light bars), evaluating how robust each model is with respect to the intervening content. We observe that all models are affected by the presence of modifiers, but the difference is narrower for *roberta-base-1b*, which offers the best stability.

Perplexity vs. SG Score. Figure 5.3 shows the relation between the average score across all circuits (*SG score*) and the perplexity of the models. As previously observed in (Hu et al., 2020a), even though there is a (not perfect) negative correlation between the two metrics when comparing different families, when comparing points corresponding to the same family of models (with equal architecture and training data size, points

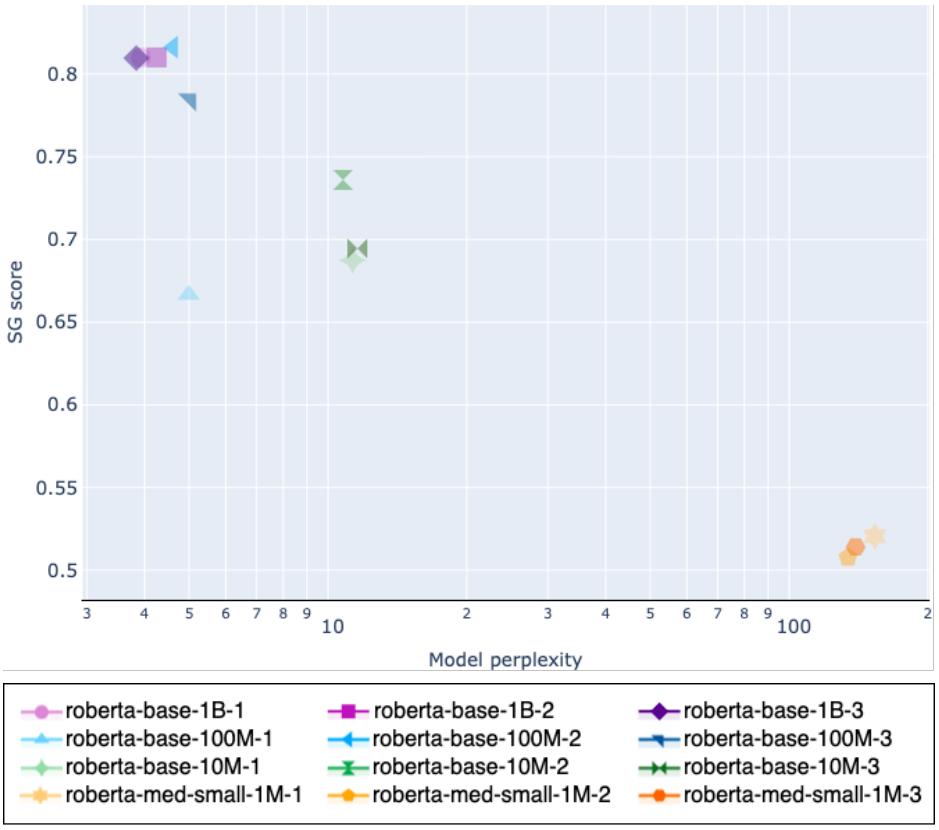


Figure 5.3: Relationship between average SyntaxGym score and model perplexity.

of the same color in Figure 5.3), there is no clear relation between them. This suggests that both metrics capture different aspects of the knowledge of the models.

Syntactic generalisation of the models. Figure 5.4 offers an overview of the syntactic capabilities of all the models on the different syntactic circuits. The family with more pretraining data, *roberta-base-1B*, outperforms all other families in 3 out of 6 circuits, but offers a surprisingly low performance in Gross Syntactic State, clearly outperformed by *roberta-base-100M* and *roberta-base-10M*, and matched by the *roberta-*

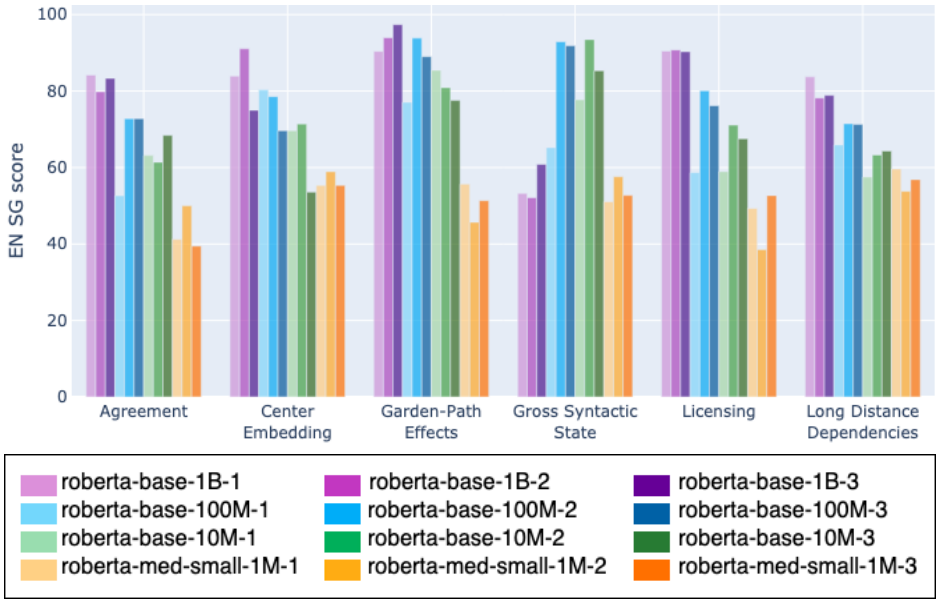


Figure 5.4: SyntaxGym evaluation across circuits.

med-small-1M. Again, the smallest family offers the lowest performance across all circuits, with individual models outperforming isolated models of other families in Center Embedding, Gross Syntactic State and Long Distance Dependencies. There is a high variability between the scores achieved by the models of the same family in the same circuit, with the exception of *roberta-base-1B* in Licensing, where all models offer a similar performance. Interestingly, there is not a single model for any family that performs best (nor worst) across all tests.

5.4 Downstream tasks evaluation

To compare the performance of the models on downstream applications, we analyse their learning curves along the fine-tuning process on two morpho-syntactic tasks (PoS tagging and dependency parsing) and a non-

syntactic task (paraphrase identification):
• **PoS tagging.** We fine-tune RoBERTa with a linear layer on top of the hidden-states output for token classification.[2] Dataset: Universal Dependencies Corpus for English (UD 2.5 EN EWT Silveira et al. (2014a)).
• **Dependency parsing.** We fine-tune a Deep Biaffine neural dependency parser (Dozat and Manning, 2017). Dataset: UD 2.5 English EWT (Silveira et al., 2014b).
• **Paraphrase identification.** We fine-tune RoBERTa with a linear layer on top of the pooled sentence representation.[3] Dataset: Microsoft Research Paraphrase Corpus (MRPC, Dolan and Brockett 2005a).

5.4.1 Experimental setup

Each task is fine-tuned for 3 epochs, with the default learning rate of $5e^{-5}$. To mitigate the variance in performance induced by weight initialisation and training data order (Dodge et al., 2020; Reimers and Gurevych, 2017), we repeat this process 5 times per task with different random seeds and average results.[4]

5.4.2 Evaluation results

We compare the performance of the different models on three different downstream tasks: PoS tagging (Figure 5.5), dependency parsing (Figure 5.6) and paraphrase identification (Figure 5.7) to determine if models pretrained on more data perform better on downstream applications. We observe the same tendency for all tasks: models with more training data perform better, and the model with the smallest architecture (*roberta-med-small-1M*) performs remarkably worse. Although note that while

[2]Source: `https://github.com/Tarpelite/UniNLP/blob/master/examples/run_pos.py`

[3]Source: `https://github.com/huggingface/transformers/blob/master/examples/text-classification/run_glue.py`.

[4]The implementation relies in the Transformers library (Wolf et al., 2020a) and AllenNLP (Gardner et al., 2018a). For implementation details, pretrained weights and hyperparameter values, cf. the documentation of the libraries.

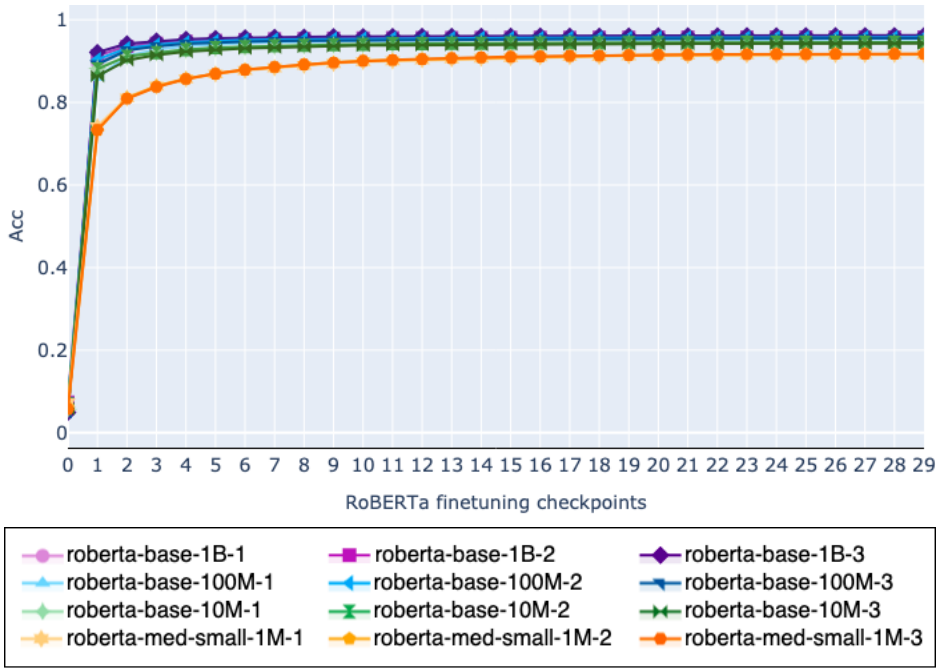


Figure 5.5: Downstream task evaluation. PoS tagging accuracy evolution.

the increase of training data between families is exponential (1M, 10M, 100M, 1B), the performance grows at a slower rate. This observation suggests that there may be a limit to the amount of data that we can feed into a RoBERTa model and the knowledge that the model can acquire.

5.5 Cost-benefit analysis

For the sake of a more holistic view on the quality of the models, we perform a cost–benefit analysis of the performance gains in the different tasks, with an estimate of the financial and environmental cost of developing the models. As the resources used to train the MiniBERTas are not publicly available, we rely on the data provided in (Strubell et al., 2019) to estimate the cost of developing each individual model based on the costs

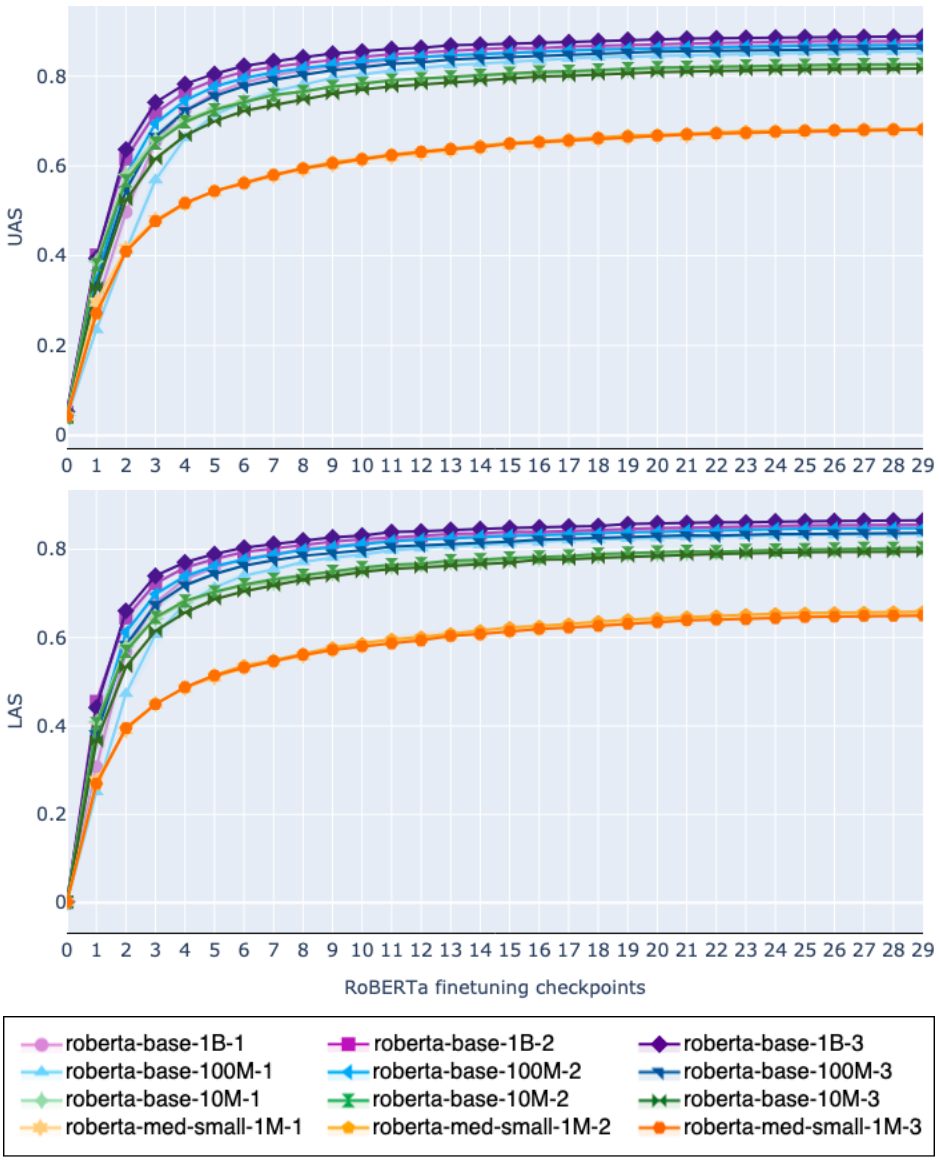


Figure 5.6: Downstream tasks evaluation. Dependency parsing UAS and LAS evolution.

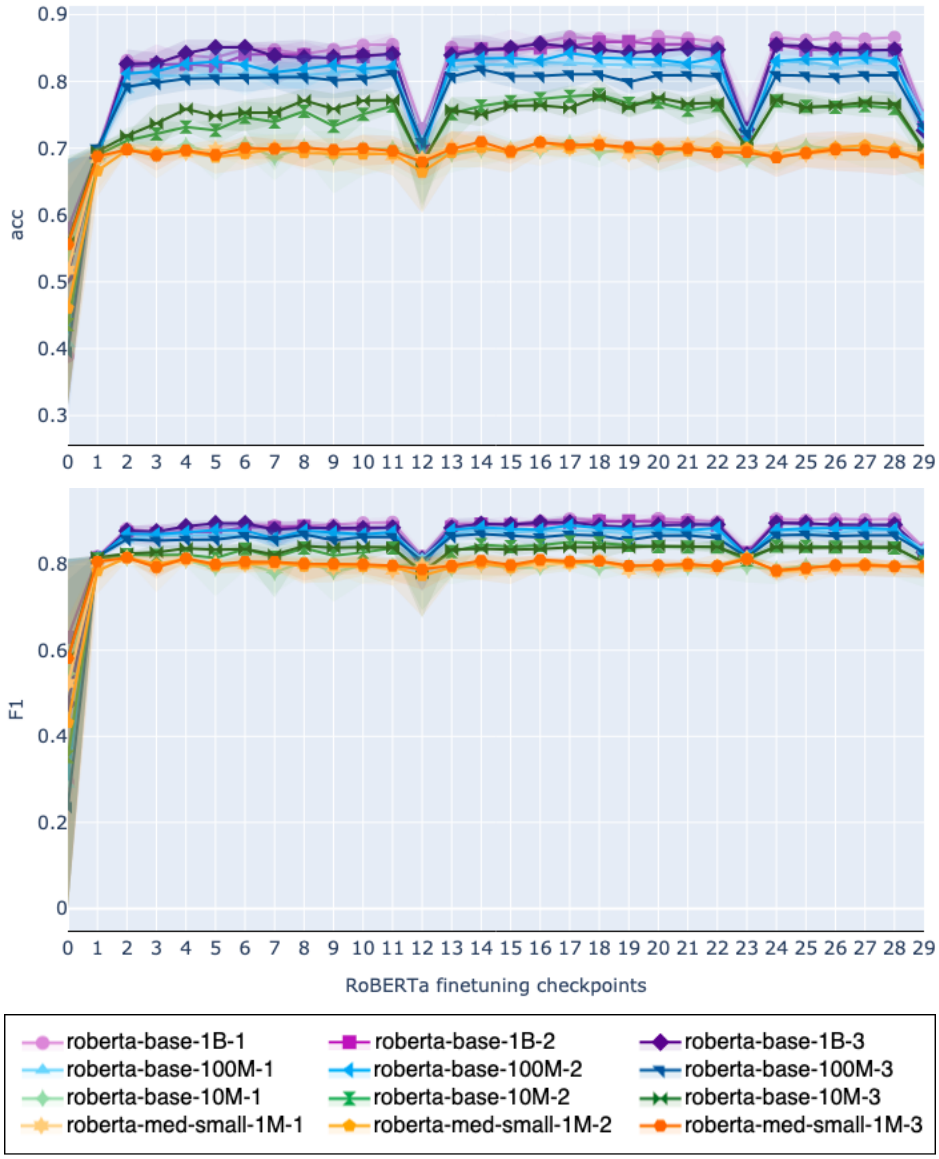


Figure 5.7: Downstream tasks evaluation. Paraphrase identification accuracy and F1 evolution.

Model family	Cost	CO_2e	PoS	Dep. parsing	Paraphrase id.
roberta-base-1B	$20320	2330	96.03 (+0.5%)	85.73 (+1.76%)	89.59 (+2.02%)
roberta-base-100M	$5075	582.5	95.53 (+1.11%)	83.97 (+4.04%)	87.57 (+2.79%)
roberta-base-10M	$500	58.25	94.42 (+2.73%)	79.93 (+14.48%)	84.78 (+5.34%)
rob-med-small-1M	$50	5.825	91.69 (base)	65.45 (base)	79.44 (base)

Table 5.3: Comparison of the estimated cost of developing the different MiniBERTas families in terms of cloud compute cost (USD) and CO_2 emissions (lbs) and their averaged performances on PoS tagging (acc), Dep. Parsing (LAS), and Paraphrase identification (F1). In parentheses, we show the increment with respect to the previous smaller model.

of RoBERTa, trained on 30B words, in proportion to the amount of words used to train each family of models.

Financial cost. As RoBERTa base was trained on 1024 Nvidia V100 GPUs for 24 hours (i.e., 24,576 GPU hours), and the price per hour of Nvidia V100 (on-demand) is $2.48 (Strubell et al., 2019), the cost of training RoBERTa base amounts to $60,948, and the cost of training a MiniBERTas model can be estimated to be $60,948 / 30B words * #TrainingWords. E.g., for the *roberta-base-1b* model: $60,948 / 30B words * 1B words = $2,032.

CO_2 Emissions. Using Strubell et al. (2019), we extrapolate that Nvidia V100 GPUs emit 0.28441456 lbs of CO_2 per GPU per hour, which means that the training of RoBERTa base emitted 6,990 lbs of CO_2. We estimate the emissions of the training of each MiniBERTas model as 6,990 lbs / 30B * #TrainingWords.

To develop each MiniBERTas models, Warstadt et al. run the pretraining 10 times for the bigger family (roberta-base-1B), and 25 times for the other three families (roberta-base-100M, roberta-base-10M and roberta-med-small-1M) with varying hyperparameters. Therefore, to compute the cost of developing each family of models, we multiply the cost of training a single model by the number of pretraining runs needed to obtain

it. Table 5.3 lists the estimated costs and CO_2 emissions of the development of each MiniBERTas family, along with their averaged performance on the three studied downstream applications. We see that small performance gains come at high financial and environmental costs. E.g., for *roberta-base-1B*, a performance increase of 0.5%–2.02% on downstream applications has a cost of $20K in computing resources and significant carbon emissions, higher than the estimated 1984 lbs generated by a single passenger flying between New York and San Francisco (Strubell et al., 2019).

5.6 Insights

Our experiments shed light on the impact of pretraining data size on the syntactic capabilities of RoBERTa. Our results indicate that models pretrained with more data encode better syntactic information (as measured by Hewitt and Manning's structural probes) and are more robust to the presence of modifiers in the syntactic tests, i.e,. intervening content inserted before the critical region. However, they do not always generalise better over the different syntactic phenomena covered by the tests assembled in (Hu et al., 2020a). As was already observed in (Hu et al., 2020a), there is no simple relationship between the perplexity of the models and the SyntaxGym score: the variance in intra-family SG score is not explained by the perplexity differences. When zooming in on the different test circuits, probing different linguistic phenomena, we observe that there is a high variability between the scores achieved by the models of the same family, with no single model for any family performing best across all tests. While the family pretrained with more data outperforms all the models of the other families on 3 out of 6 circuits, it offers a surprisingly low performance in Gross Syntactic State, clearly outperformed by the smaller models.

We also compare the performance of the different models fine-tuned on PoS tagging, dependency parsing and paraphrase identification, observing that models with more training data offer a better performance, and the

model with the smallest architecture (roberta-med-small-1M) performs remarkably worse. However, while the amount of training data between families grows exponentially, we observe that the performance grows at a much slower rate, suggesting that there may be a limit to the knowledge that a RoBERTa model can acquire solely from raw pretraining data.
We complement our findings with a financial and environmental cost–benefit analysis of pretraining models on different amounts of data. We show that while models pretrained on more data encode more syntactic information and perform generally better on downstream applications, small performance gains come at a huge financial and environmental cost. Thus, when developing and training new models we should weigh between the benefit of making models bigger and pretraining them on huge datasets and the costs this implies, prioritising computationally efficient hardware and algorithms.

Chapter 6

IMPACT OF FINE-TUNING ON THE SYNTACTIC KNOWLEDGE ENCODED IN LANGUAGE MODELS

As shown all along this thesis, adapting unsupervised pretrained language models (LMs) to solve supervised tasks has become a widely spread practice in NLP, with with Transformer-based models such as BERT (Devlin et al., 2019b) and RoBERTa (Liu et al., 2019b) achieving state-of-the-art results in many well-known NLU benchmarks like GLUE (Wang et al., 2019a) and SQuAD (Rajpurkar et al., 2018b).

Even though pretrained language models can be used as frozen feature extractors, they are often fine-tuned to solve downstream tasks (Peters et al., 2019), and therefore it is important to understand how the encoded knowledge evolves along the fine-tuning process. In this Chapter, we aim to understand how syntax trees implicitly embedded in the geometry of deep models (Hewitt and Manning, 2019) evolve along the fine-tuning process of BERT on different supervised tasks, and shed some light on the importance of the syntactic information for those tasks. Intuitively, we expect morpho-syntactic tasks to clearly reinforce the encoded syn-

tactic information, while tasks that are not explicitly syntactic in nature should maintain it in case they benefit from syntax (Kuncoro et al., 2020) and lose it if they do not. In order to cover the three main levels of the linguistic description (morphology, syntax and semantics), we select six different tasks: PoS tagging, constituency parsing, syntactic dependency parsing, semantic role labeling (SRL), QA and paraphrase identification. The first three inherently deal with (morpho-)syntactic information while the latter three, which traditionally draw upon the output of syntactic parsing (Carreras and Màrquez, 2005; Björkelund et al., 2010; Strubell et al., 2018; Wang et al., 2019b, inter-alia), deal with higher level, semantic information. Almost all of our experiments are on English corpora; one is on multilingual dependency parsing.

The remainder of the chapter is structured as follows. Section 6.1 describes our experimental setup, presenting the tasks that we use to fine-tune BERT on and the probe that we use to analyse the evolution of the syntactic knowledge encode in the models. Section 6.2 presents our analysis on the evolution of syntactic knowledge along the fine-tuning of the model on the different tasks, and Section 6.3 complements the results with the performance curves of the target tasks for which the models are fine-tuned, along with the performance curves of the Hewitt and Manning's structural probes metrics, facilitating the comparison of the evolution of the encoded syntax trees information and the target tasks performances. Finally, Section 6.4 summarises our findings.

6.1 Experimental setup

We study the evolution of the syntactic structures discovered during pre-training along the fine-tuning of BERT-base (cased)[1] on six different tasks,

[1] Our experiments are implemented in PyTorch, using two open-source libraries: the Transformers library (Wolf et al., 2020b) and AllenNLP (Gardner et al., 2018b). Implementation details, pretrained weights and full hyperparameter values can be found in the libraries documentation.

drawing upon the structural probe of Hewitt and Manning (2019).[2] We fine-tune the whole model on each task outlined below for 3 epochs, with a learning rate of $5e^{-5}$, saving 10 evenly-spaced checkpoints per epoch. The output of the last layer is used as input representation for the classification components of each task. To mitigate the variance in performance induced by weight initialisation and training data order (Dodge et al., 2020), we repeat this process 5 times per task with different random seeds and average results.

6.1.1 Hewitt and Manning structural probe.

Hewitt and Manning (2019)'s structural probe, thoroughly described in Section 5.2.1, evaluates how well syntax trees are embedded in a linear transformation of the network representation space, performing two different evaluations: 1– Tree distance evaluation, in which squared L2 distance encodes the distance between words in the parse tree; and 2– Tree depth evaluation, in which squared L2 norm encodes the depth of the parse tree. Using their probe, Hewitt and Manning show that the 7th layer of BERT-base is the layer that encodes more syntactic information. Therefore, to analyse the evolution of the encoded syntax trees, we train the probes on the 7th layer of the different checkpoint models generated along the fine-tuning process of each task.

6.1.2 Downstream tasks description

To analyse the impact of the fine-tuning process on the syntactic information encoded by BERT, we analyse the evolution of the syntax trees implicitly embedded in its geometry (Hewitt and Manning, 2019) along the fine-tuning on six different supervised tasks:

• **PoS tagging.** We fine-tune BERT with a linear layer on top of the

[2]We use the same experimental setup used by the authors. Source: https://github.com/john-hewitt/structural-probes

hidden-states output for token classification.[3] Dataset: Universal Dependencies Corpus for English (UD 2.5 EN EWT Silveira et al. (2014a)).

• **Constituency parsing.** Following Vilares et al. (2020), we cast constituency parsing as a sequence labeling problem, and use a single feed-forward layer on top of BERT to directly map word vectors to labels that encode a linearised tree. Dataset: Penn Treebank (Marcus et al., 1993).

• **Dependency parsing.** We fine-tune a Deep Biaffine neural dependency parser (Dozat and Manning, 2017) on three different datasets: 1– UD 2.5 English EWT (Silveira et al., 2014a); 2– a multilingual benchmark generated by concatenating the UD 2.5 standard data splits for German, English, Spanish, French, Italian, Portuguese, and Swedish (Nivre et al., 2017), with gold PoS tags; 3– PTB SD 3.3.0 (de Marneffe et al., 2006).

• **Semantic role labeling.** Following Shi and Lin (2019), we decompose the task into 1– predicate sense disambiguation and argument identification; and 2– classification. Both subtasks are casted as sequence labeling, feeding the contextual representations into a one-hidden-layer Multi-Layer Perceptron (MLP) for the first, and a one-layer BiLSTM followed by a one-hidden-layer MLP for the latter. Dataset: OntoNotes corpus (Weischedel et al., 2013).

• **Question answering.** We fine-tune BERT with a linear layer on top of the hidden-states output to compute span start logits and span end logits.[4] Dataset: SQuAD, Stanford Question Answering Dataset (Rajpurkar et al., 2018b).

• **Paraphrase identification.** We fine-tune BERT with a linear layer on top of the pooled sentence representation.[5] Dataset: MRPC (Dolan and Brockett, 2005b).

[3] Source: https://github.com/Tarpelite/UniNLP/blob/master/examples/run_pos.py

[4] Source: https://github.com/huggingface/transformers/tree/master/examples/question-answering.

[5] Source: https://github.com/huggingface/transformers/blob/master/examples/text-classification/run_glue.py.

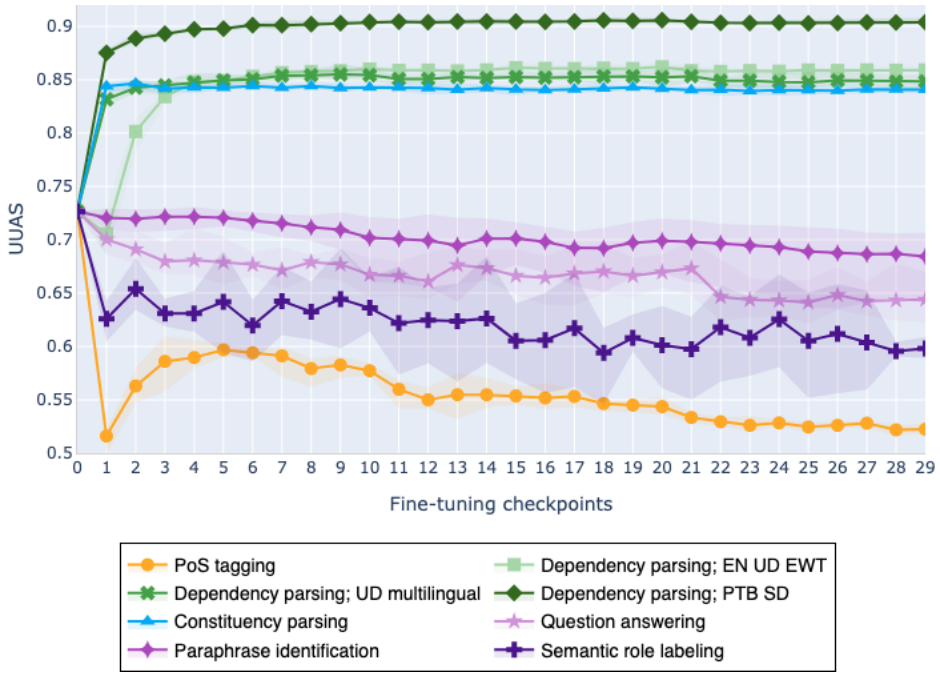


Figure 6.1: Tree distance evaluation. *UUAS* evolution.

6.2 Evolution of the syntactic knowledge during fine-tuning

6.2.0.1 Tree distance evaluation

The probe evaluates how well the predicted distances between all pairs of words in a model reconstruct gold parse trees by computing the Undirected Unlabeled Attachment Score (*UUAS*). It also computes the Spearman correlation between true and predicted distances for each word in each sentence, averaging across all sentences with lengths between 5 and 50 (henceforth referred to as *DSpr.*).

Morpho-syntactic tasks. As shown in Figures 6.1 and 6.2, both metrics follow a similar behaviour (shades represent the variability across

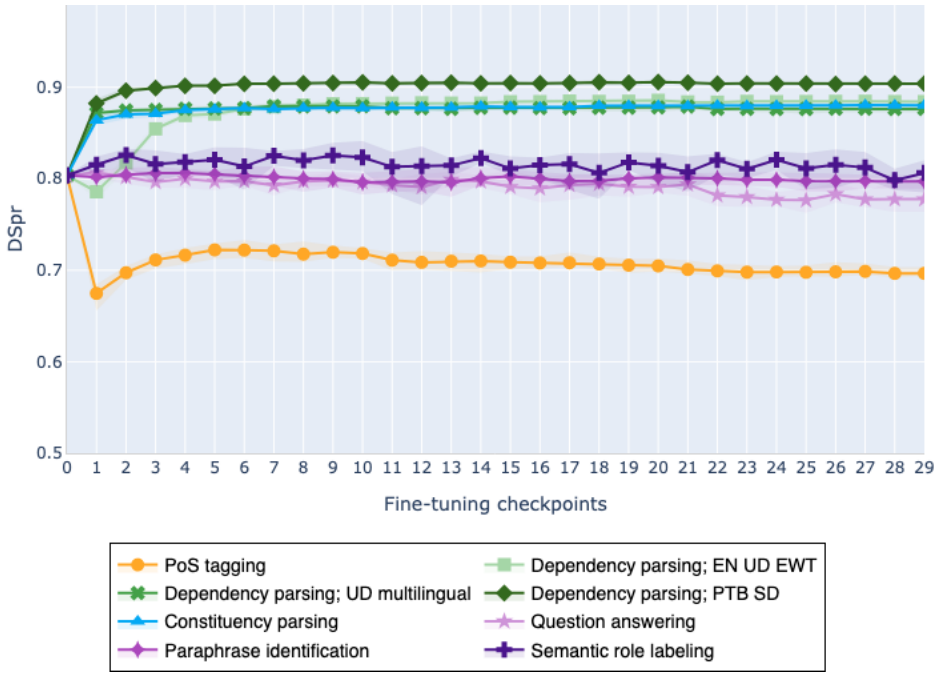


Figure 6.2: Tree distance evaluation. *Dspr* evolution.

the 5 model runs). PoS tagging shows an important loss of performance all along the fine-tuning process, especially noticeable for UUAS (Figure 6.1), suggesting that distance-related syntactic information is of less relevance to PoS tagging than could be intuitively assumed. As many words have a clear preference towards a specific PoS, especially in English, and most of the ambiguous cases can be resolved using information in the close vicinity (e.g., a simple 3-gram sequence tagger is able to achieve a very high accuracy (Manning, 2011)), syntactic structure information may not be necessary and, therefore, the model does not preserve it. This observation is aligned with Pimentel et al. (2020b), who found that PoS-tagging is not an ideal task for contemplating the syntax contained in contextual word embeddings. The loss is less pronounced on depth-related metrics, maybe because the root of the sentence usually corresponds to the verb, which may also help in identifying the PoS of

surrounding words.

Constituency parsing and dependency parsing share a very similar tendency, with a big improvement in the first fine-tuning steps preserved along the rest of the process. As both tasks heavily rely on syntactic information, this improvement intuitively makes sense. Dependency parsing fine-tuned on the Penn Treebank (PTB) shows even higher results since the probing is trained on the same dataset. Interestingly, the probe performs similarly even if the parsing task is modeled as a sequence labeling problem (as in constituency parsing), suggesting that the structure of syntax trees emerges in such models even when no tree is explicitly involved in the task. The initial drop observed for PoS tagging and monolingual dependency parsing with UD, trained on UD EN EWT, may be related to the size of the dataset, since UD EN EWT is significantly smaller than the other datasets and therefore the models see less examples per checkpoint.

Semantics-related tasks. As shown in Figures 6.1 and 6.2, both metrics follow different behaviours (again, shades represent the variability across the 5 model runs). Paraphrase identification shows a small but constant UUAS loss along the fine-tuning, while QA shows a slightly steeper loss trend. Initially, SRL loses around 12 points, suggesting that it discards some syntactic information right at the beginning, and follows a similar downward trend afterwards. Those three tasks show a stable performance along the fine-tuning for the DSpr metric, which implies that even if there is a loss in UUAS information it does not impact the distance ordering.

6.2.0.2 Tree depth evaluation

The probe evaluates models with respect to their ability to recreate the order of words specified by their depth in the parse tree, assessing their ability to identify the root of the sentence as the least deep word (*Root %*) and computing the Spearman correlation between the predicted and the true depth ordering, averaging across all sentences with lengths between 5 and 50 (henceforth referred to as *NSpr*).

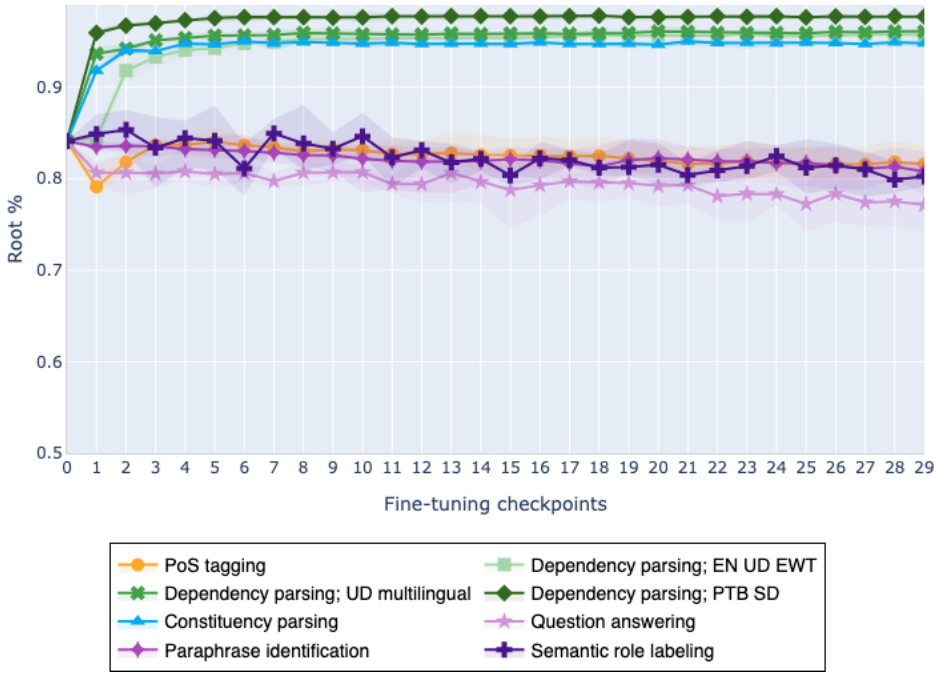


Figure 6.3: Tree depth evaluation. *Root %* evolution.

Morpho-syntactic tasks Again, both metrics follow a similar behaviour, as shown in Figures 6.3 and 6.4. PoS tagging shows a sustained loss of performance, though softer than the loss observed for the distance metrics. This loss is slightly less pronounced for *Root %* than for *Nspr*, suggesting that while depth-related syntactic information may be of less relevance to PoS tagging than it is to the other morpho-syntactic tasks, identifying the root of the sentence may be important, as the root of the sentence is likely to become one of the ambiguous tags and therefore identifying it may help to select the correct label. Constituency parsing and dependency parsing share a similar tendency, with a big improvement in the first steps preserved along the rest of the fine-tuning process, reinforcing the intuition previously introduced in Section 6.2.0.1 about the structure of syntax trees emerging in models even when no tree is explicitly involved in the task. Again, an initial drop can be observed for PoS tagging and

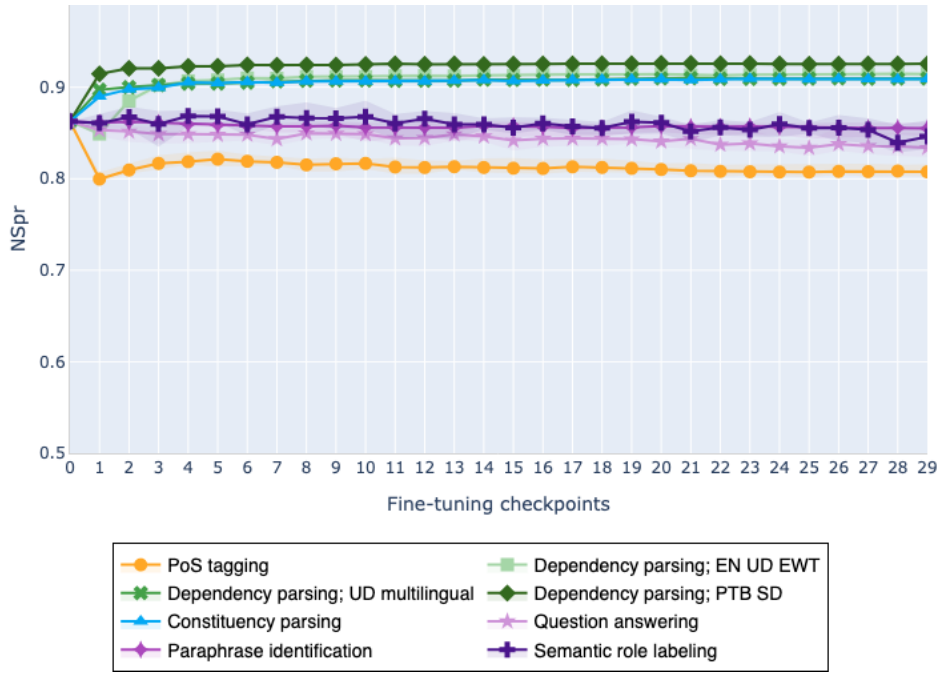


Figure 6.4: Tree depth evaluation. *Nspr* evolution.

monolingual dependency parsing with UD, most probably related to the smaller size of the UD EN EWT dataset used in both tasks.

Semantics-related tasks Both metrics follow a similar behaviour, as shown in Figures 6.3 and 6.4, with all tasks following a soft but sustained loss of performance until the end of the fine-tuning process, specially noticeable for *Root %*.

6.3 Target tasks performance evolution

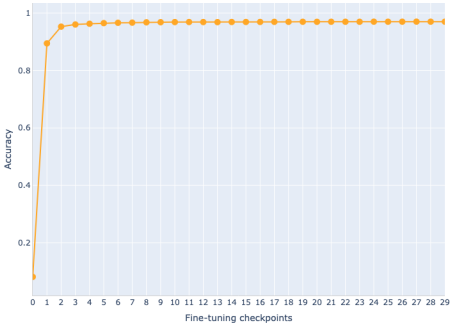
To complement the results from the previous section, we include here the performance curves of the target tasks for which the models are fine-tuned, along with the performance curves of the structural probes metrics,

facilitating the comparison of the evolution of the encoded syntax trees information and the target tasks performances. Along with the performance curves of the four structural probes metrics (*UUAS*, *Nspr*, *Root %* and *Dspr*), the following figures include the performance curves of the target tasks.

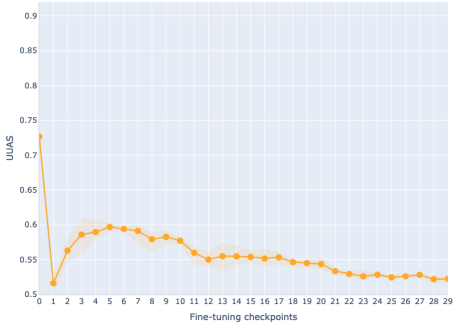
PoS tagging. As shown in Figure 6.5, it reaches a 0.95 accuracy in only two checkpoints, ending up with a 0.97 on the last checkpoint (Figure 6.5a). It shows a loss of accuracy for the four probing metrics all along the fine-tuning process, especially noticeable for *UUAS* (Figure 6.5b) and *Root %* (Figure 6.5d), suggesting that syntactic information is of less relevance to PoS tagging than could be intuitively assumed. The loss is less pronounced on depth-related metrics, maybe due to the fact that the root of the sentence usually corresponds to the verb, which may also help in identifying the PoS of surrounding words.

Dependency parsing with PTB SD. Figure 6.6 shows a steep learning curve for the Labeled Attachment Score (LAS), as shown in Figure 6.6b, reaching a performance of 0.90 LAS on the third checkpoint, up to a final 0.94. All four probing metrics show an important improvement in the first fine-tuning step (Figures 6.6c to 6.6f), which is preserved along the rest of the process. As the task heavily relies on syntactic information, this improvement intuitively makes sense. Compared to the result of the other dependency parsing experiments, this one show bigger improvements because the probing is trained on the same dataset.

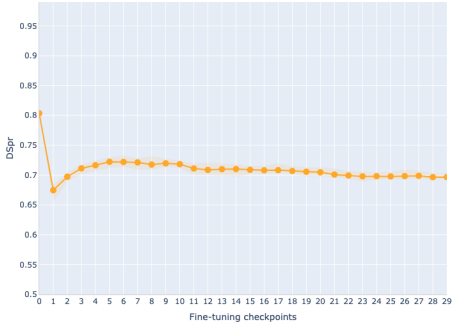
Dependency parsing with EN UD EWT. As show in Figure 6.7, it shows a shallower learning curve than other experiments (Figure 6.7b), as the dataset is significantly smaller than the multilingual and PTB and therefore the models see less examples per checkpoint, ending up with a high performance of 0.9. After an initial drop (probably due to the dataset size, as mentioned before), the probing metrics show a big improvement in the first fine-tuning steps, preserved along the rest of the process (Fig-



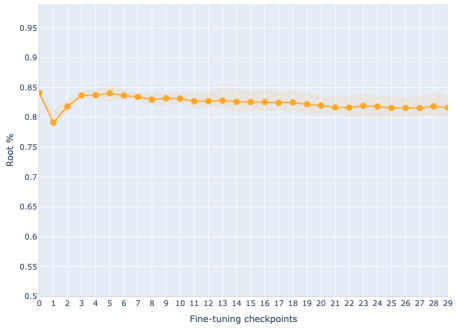
(a) Fine-tuning. Accuracy



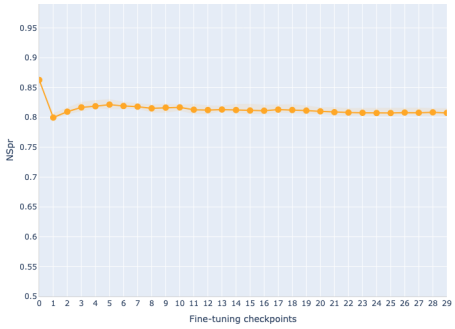
(b) Structural probes tree distance evaluation. *UUAS*



(c) Structural probes tree distance evaluation. *Dspr*

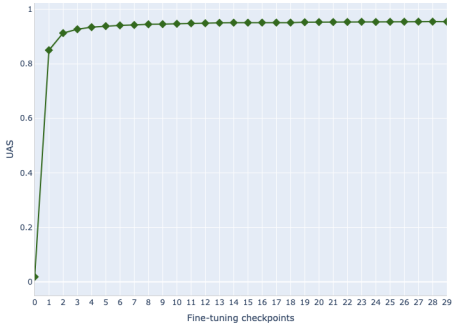


(d) Structural probes tree depth evaluation. *Root %*

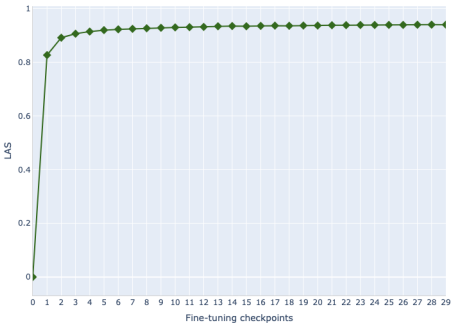


(e) Structural probes tree depth evaluation. *Nspr*.

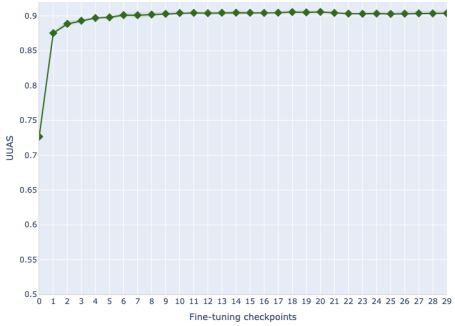
Figure 6.5: POS Tagging. Fine-tuning & probing metrics evolution.



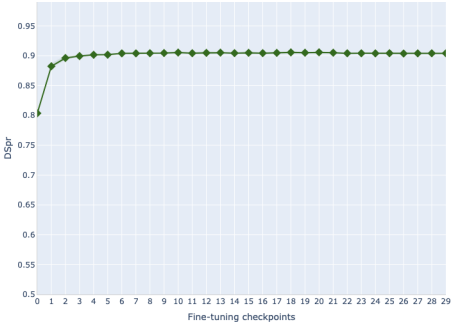
(a) UAS



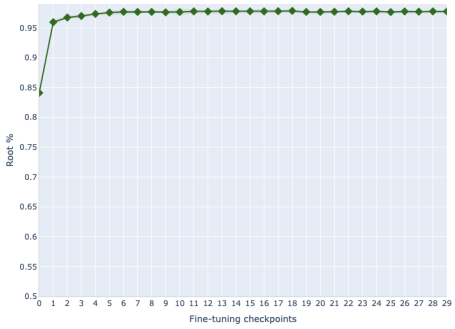
(b) Fine-tuning. LAS



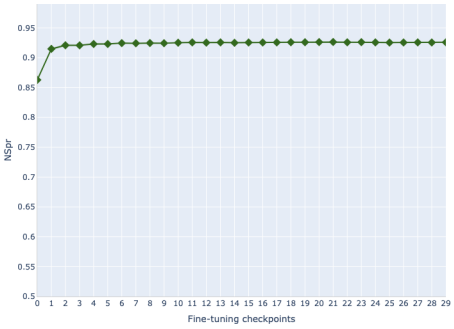
(c) Structural probes tree distance evaluation. *UUAS*



(d) Structural probes tree distance evaluation. *Dspr*.



(e) Structural probes tree depth evaluation. *Root %*



(f) Structural probes tree depth evaluation. *Nspr*

Figure 6.6: Dependency Parsing PTB SD. Fine-tuning & probing metrics evolution.

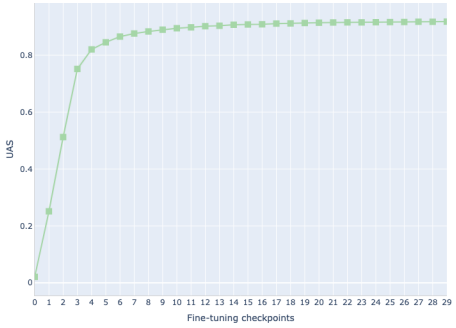
ures 6.7c to 6.7f). As the task heavily relies on syntactic information, this improvement intuitively makes sense.

Multilingual dependency parsing. As show in Figure 6.8, it shows a steeper learning curve than dependency parsing with EN UD EWT, as it is trained with a larger dataset (Figure 6.8b), reaching a performance of 0.87 in LAS. All four probing metrics show a big improvement in the first fine-tuning step, preserved along the rest of the process (Figures 6.8c to 6.8f). As the task heavily relies on syntactic information, this improvement intuitively makes sense.

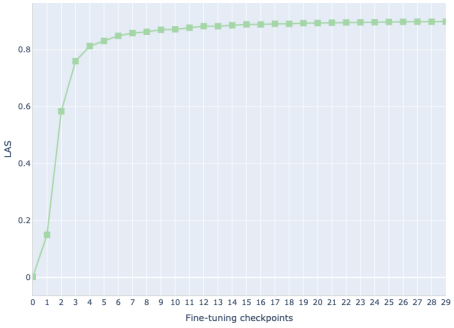
Constituency parsing. As shown in Figure 6.9, fine-tuning follows a steep curve, quickly reaching an Accuracy of 0.87 that is further improved to 0.9 in the last checkpoint (Figure 6.9a). All four probing metrics show a big improvement in the first fine-tuning steps, preserved along the rest of the process (Figures 6.9b to 6.9e). As the task heavily relies on syntactic information, this improvement intuitively makes sense. Interestingly, even though the task is modeled as a sequence labeling problem, the probe performs similarly to the dependency parsing tasks, suggesting that the structure of syntax trees emerges in such models even when no tree is explicitly involved in the task.

Question answering. As shown in Figure 6.10, fine-tuning quickly hits an F1 score of 0.73 on the first step, which is further improved to 0.88 in the last checkpoint (Figure 6.10a). All four probing metrics show a clear loss trend (Figures 6.10b to 6.10e). The loss is specially noticeable for *UUAS* and *Root %*, and more stable for the Spearman correlations, suggesting that even if there is a loss of information it does not impact the distance and depth orderings.

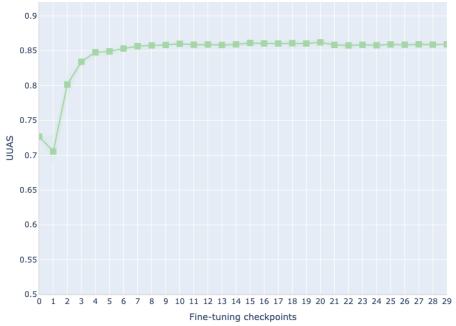
Paraphrase identification. As shown in Figure 6.11, fine-tuning starts with an F1 score of 0.81 on the first step that is further improved to 0.90 in the last checkpoint (Figure 6.11a). Regarding accuracy, after reaching



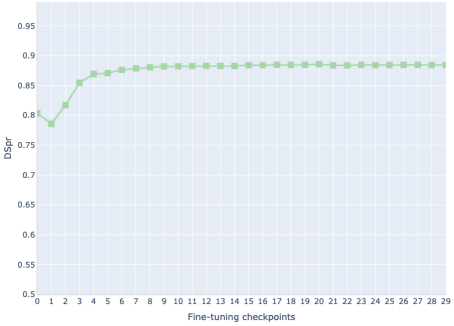
(a) UAS



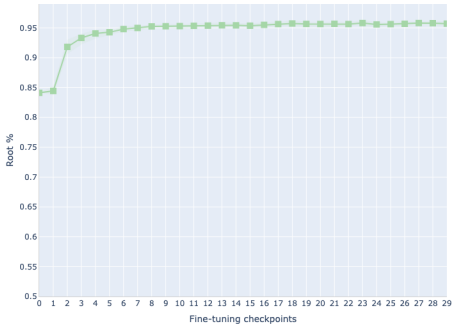
(b) Fine-tuning. LAS



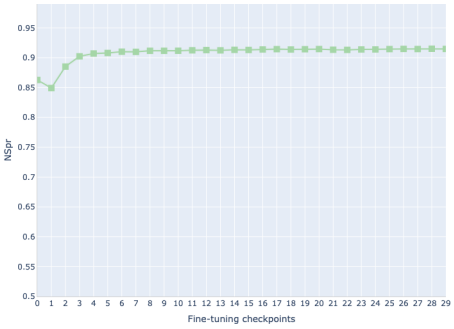
(c) Structural probes tree distance evaluation. *UUAS*



(d) Structural probes tree distance evaluation. *Dspr*.

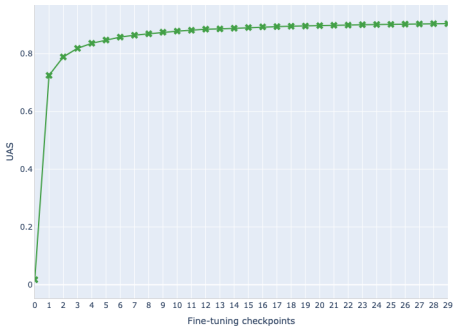


(e) Structural probes tree depth evaluation. *Root %*

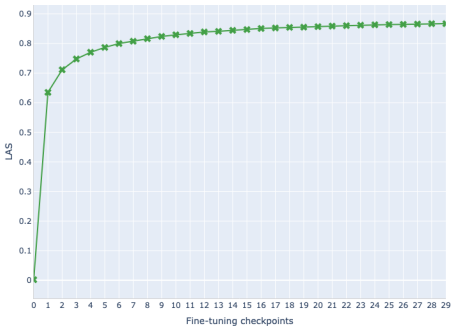


(f) Structural probes tree depth evaluation. *Nspr*

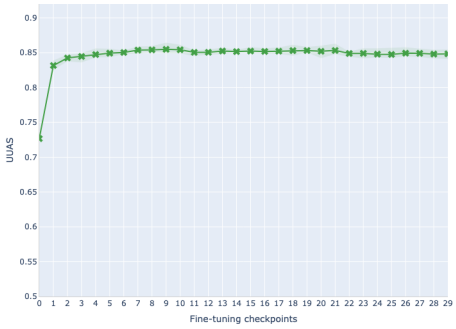
Figure 6.7: Dependency Parsing EN UD EWT. Fine-tuning & probing metrics evolution.



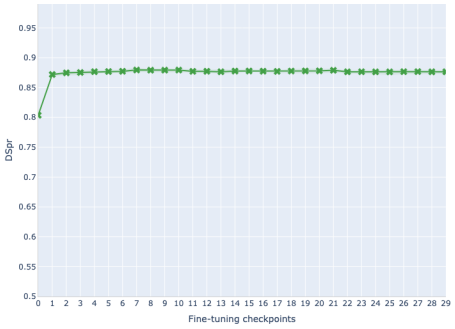
(a) UAS



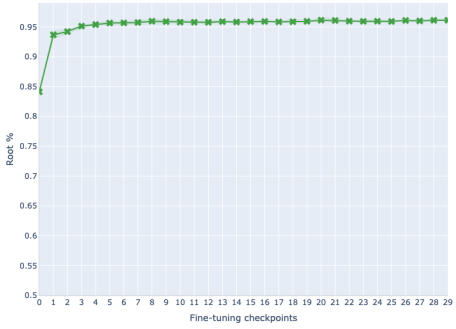
(b) Fine-tuning. LAS



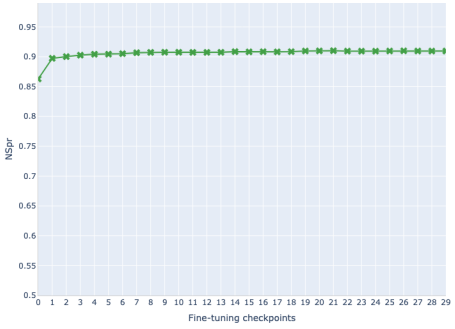
(c) Structural probes tree distance evaluation. *UUAS*



(d) Structural probes tree distance evaluation. *Dspr*.

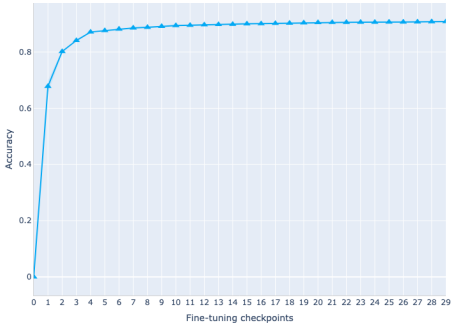


(e) Structural probes tree depth evaluation. *Root %*

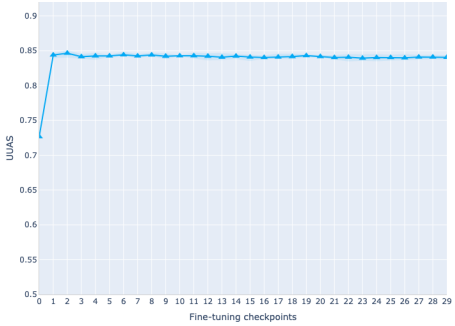


(f) Structural probes tree depth evaluation. *Nspr*

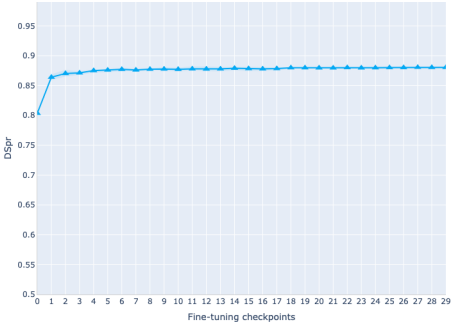
Figure 6.8: Dependency Parsing UD Multilingual. Fine-tuning & probing metrics evolution.



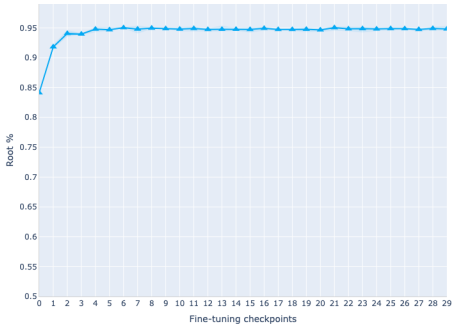
(a) Fine-tuning. Accuracy



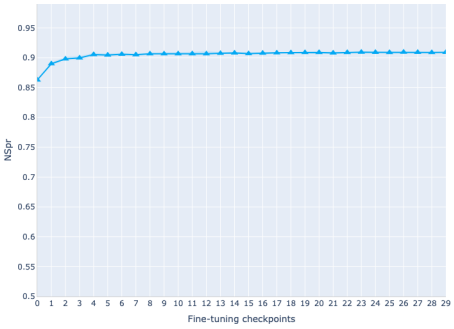
(b) Structural probes tree distance evaluation. *UUAS*



(c) Structural probes tree distance evaluation. *Dspr*.

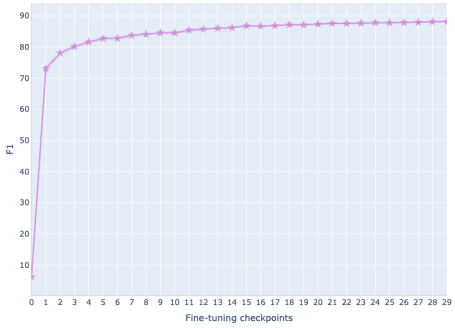


(d) Structural probes tree depth evaluation. *Root %*

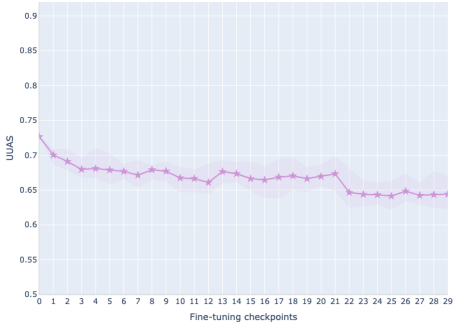


(e) Structural probes tree depth evaluation. *Nspr*

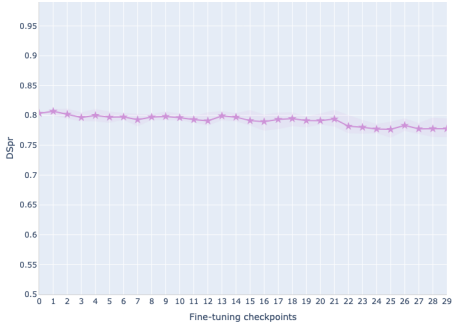
Figure 6.9: Constituent Parsing. Fine-tuning & probing metrics evolution.



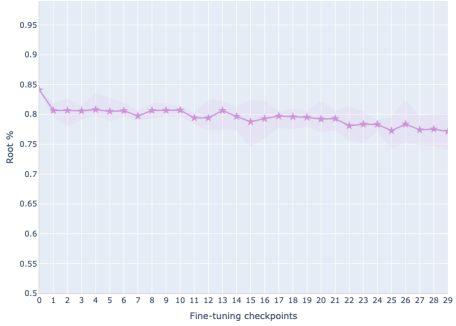
(a) Fine-tuning. F1



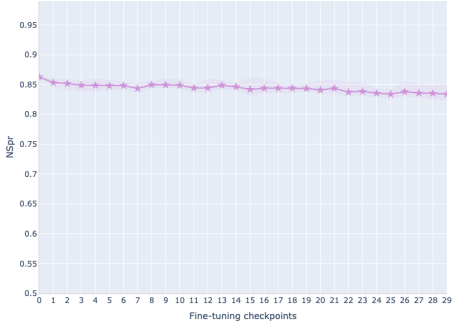
(b) Structural probes tree distance evaluation. *UUAS*



(c) Structural probes tree distance evaluation. *Dspr*.



(d) Structural probes tree depth evaluation. *Root %*



(e) Structural probes tree depth evaluation. *Nspr*

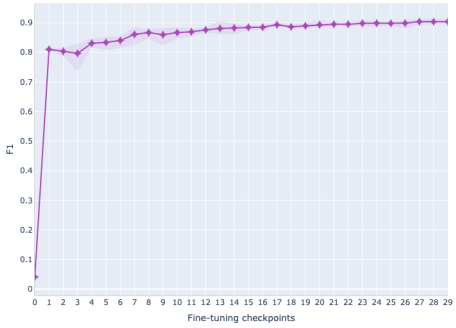
Figure 6.10: Question Answering. Fine-tuning & probing metrics evolution.

0.69 on the first checkpoint it follows a shallower curve to a final 0.86 (Figure 6.11b). All four probing metrics follow a loss trend (Figures 6.11c to 6.11f). The loss is specially noticeable for *UUAS* and *Root %*, and more stable for the Spearman correlations, suggesting that even if there is a loss of information it does not impact the distance and depth orderings.

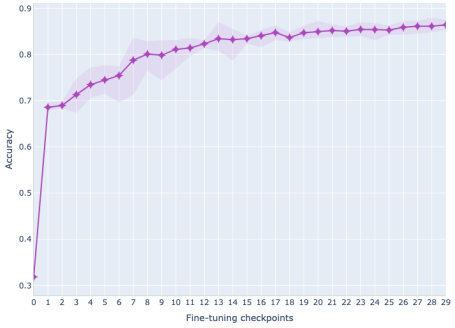
Semantic Role Labeling. As shown in Figure 6.12, fine-tuning follows a steep curve for F1, quickly reaching an F1 score of 0.71 on the first step that is further improved to 0.82 in the last checkpoint (Figure 6.12a). All four probing metrics follow a loss trend (Figures 6.12b to 6.12e). The loss is specially noticeable for *UUAS*, which initially loses around 12 *UUAS* points, and more stable for the Spearman correlations, suggesting that even if there is a loss of information it does not impact the distance and depth orderings.

6.4 Insights

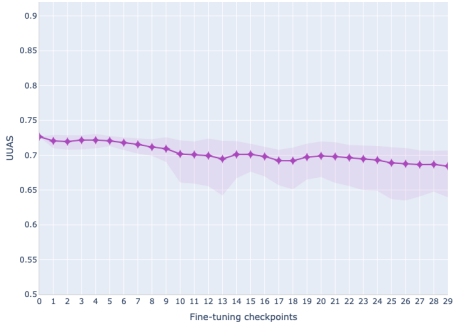
We show that fine-tuning is not always a conservative process. Rather, the syntactic information initially encoded in the models is forgotten (PoS tagging), reinforced (parsing) or preserved (semantics-related tasks) in different (sometimes unexpected) ways along the fine-tuning, depending on the task. We expected that morpho-syntactic tasks clearly reinforce syntactic information. However, PoS tagging forgets it, which, on the other side, can also be justified linguistically (cf. Section 6.2.0.1). In contrast, tasks closer to semantics mostly preserve the syntactic knowledge initially encoded. This interesting observation reinforces recent findings that models benefit from explicitly injecting syntactic information for such tasks (Sachan et al., 2021b; Xu et al., 2021).



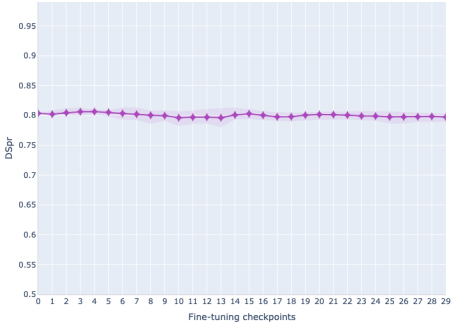
(a) Fine-tuning. F1



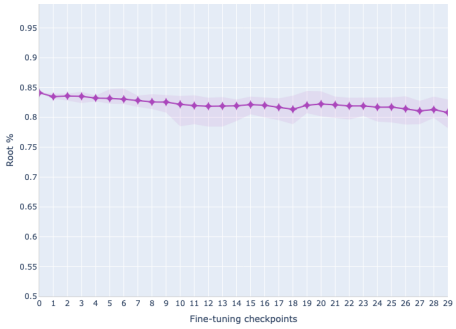
(b) Accuracy



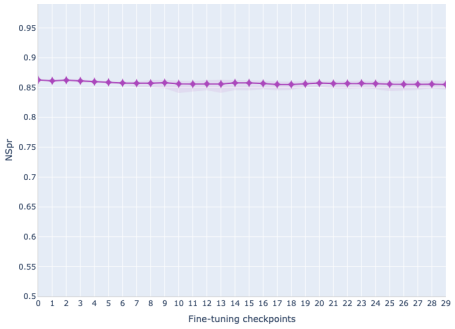
(c) Structural probes tree distance evaluation. *UUAS*



(d) Structural probes tree distance evaluation. *Dspr*.

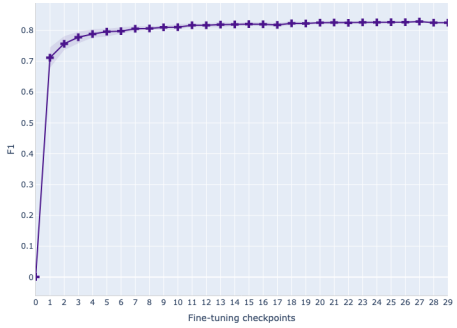


(e) Structural probes tree depth evaluation. *Root %*

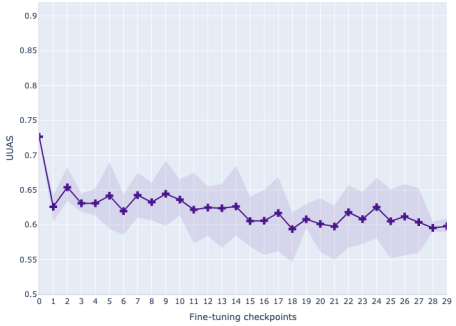


(f) Structural probes tree depth evaluation. *Nspr*

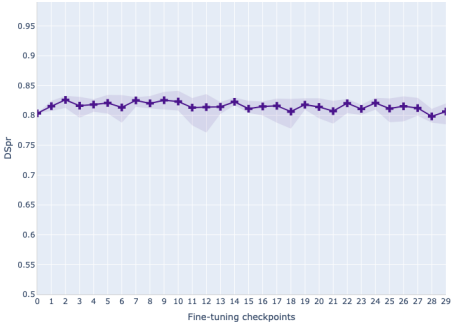
Figure 6.11: Paraphrase identification. Fine-tuning & probing metrics evolution.



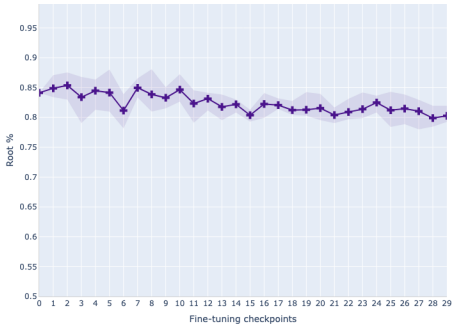
(a) Fine-tuning. F1



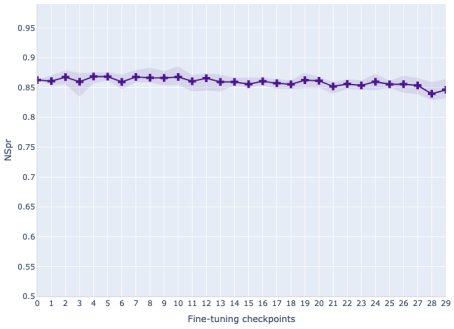
(b) Structural probes tree distance evaluation. *UUAS*



(c) Structural probes tree distance evaluation. *Dspr*.



(d) Structural probes tree depth evaluation. *Root %*



(e) Structural probes tree depth evaluation. *Nspr*

Figure 6.12: Semantic Role Labeling. Fine-tuning & probing metrics evolution.

Chapter 7

CONCLUSIONS AND FUTURE WORK

In this last chapter, we offer a summary of the findings and contributions of this dissertation, answering the research questions stated in Section 1.2. Next, we review relevant lines of future work related to the research topics explored along this thesis. To conclude, we offer some final remarks about the opportunities and dangers presented by big pretrained models, along with new research directions given rise by them.

7.1 Summary of findings and contributions

Since the publication of the Transformer in 2017, the NLP field has experienced a total revolution. In particular, the publication of BERT marked an inflection point, and traditional approaches to model NLP tasks were quickly replaced by new architectures relying mainly on pretrained Transformer-based models able to generate powerful contextual embeddings. The accessibility and usability of BERT-based pretrained models, available through many well-known libraries such as Tensorflow, PyTorch and MXNet, contributed to the application of the models to many tasks, pushing the state of the art to new levels. Two main questions emerge: what do these models learn, and how do they use this knowledge? This thesis aims

at shedding light on these questions by offering an extensive empirical comparison of the morpho-syntactic capabilities of different pretrained Transformer-based autoencoding models. We thoroughly explored the three main related dimensions of pretrained language models, namely: 1– language: monolingual (English and Spanish) and multilingual models; 2– pretraining objectives: (masked) language modeling and sentence-based tasks such as next sentence prediction; and 3– amount of training data. We now recapitulate how our methods addressed the research objectives and summarise our contributions and findings.

Syntactic generalisation abilities of pretrained models (Chapter 4). While multilingual models achieve outstanding results in a wide range of cross-lingual transfer tasks, it remained unknown whether the optimisation for different languages conditions the capacity of the models to generalise over syntactic structures, and how languages with syntactic phenomena of different complexity are affected.

- **Do multilingual models generalise equally well across languages?** We show that multilingual models do not generalise equally well across languages: while mBERT generalises better for phenomena in English, XLM-R does it better for phenomena in Spanish.

- **How well do monolingual models generalise over syntactic phenomena compared to multilingual models?** We show that there is a substantial difference between the syntactic generalisation potential of monolingual and multilingual models. But this difference depends on the language: while for English monolingual models (BERT and RoBERTa) offer a higher syntactic generalisation than multilingual models (mBERT and XLM-R), this is not the case for Spanish, for which multilingual models (XLM-R) generalise better. While it is possible that the multilingual abstractions captured by XLM-R become useful for morphologically rich languages such as Spanish, this difference may also be related to the huge difference in training data between the two models.

- **Does the presence of modifiers affect the generalisation capabilities of the models?** We show that performance of all models is affected by the presence of modifiers, indicating that modeling the complexity of the syntactic structure is still very challenging.

- **Does the nature of the training procedures employed to train the models affect the generalisation capabilities of the models?** The lack of NSP objective in RoBERTa-based models seems to harm BETO but not XLM-R, reinforcing our intuition that BETO may be improved with (much) more training data. Also, we show that BERT-based models outperform all RoBERTa-based models in the particular case of the Gross Syntactic State circuit, suggesting that RoBERTa-based models may also benefit from complementary training objectives in their pretraining procedure.

Impact of the pretraining data size on the syntactic abilities of the models (Chapter 5). Training language models is expensive in terms of time and resources, with two factors having a major impact in this cost: the size of the model (in millions of parameters) and the amount of data used to train it. We analysed the impact of pretraining data size on the morpho-syntactic knowledge of the MiniBERTa models, a set of RoBERTa models trained on incremental amounts of data.

- **Do models pretrained with more data encode more syntactic information and generalise better over syntactic phenomena? Are they more robust to the presence of modifiers?** We show that models pretrained with more data encode better syntactic information (as measured by Hewitt and Manning’s structural probes) and are more robust to the presence of modifiers. However, while models pretrained with more data generalise better over syntactic phenomena on average, they only offer a higher syntactic generalisation on half of the studied phenomena. Importantly, we show that there is a high variability in the performance of models of equal size and training data, initialised with different random seeds.

- **Do models pretrained with more data offer a better performance on downstream tasks such as dependency parsing and paraphrase identification?** Indeed, we show that models with more training data offer a better performance. However, while the amount of training data between families grows exponentially, the performance grows at a much slower rate, suggesting that there may be a limit to the knowledge that a RoBERTa model can acquire solely from raw training data.

- **Is there a correlation between the language modeling abilities of the models and their syntactic generalisation abilities?** Corroborating findings in (Hu et al., 2020a), we show that there is no simple relationship between the perplexity of the models and their average syntactic generalisation capabilities, suggesting that it may be possible to complement information-theoretical metrics such as perplexity with metrics measuring specific types of knowledge, e.g., syntax, in order to develop and select NLU models that are potentially more robust and efficient.

Evolution of syntactic knowledge during fine-tuning (Chapter 6). Although pretrained models can be used frozen as feature extractors, they are often fine-tuned on downstream tasks, and therefore it becomes increasingly important to understand how the knowledge initially encoded in the models evolves along the process. We studied how the encoded syntactic knowledge (as measured by Hewitt and Manning's structural probes) evolves along the fine-tuning process of BERT on six different tasks, covering all levels of the linguistic structure.

- **Is the syntactic information initially encoded in the models forgotten, preserved, or reinforced along the fine-tuning process? Does it depend on the task in which the models are fine-tuned?** We show that morpho-syntactic tasks experiment substantial changes in the initial phases: while PoS tagging forgets a high amount of syntactic information, dependency parsing clearly reinforces it. On the other hand, semantics-related tasks maintain a more stable trend,

mostly preserving the syntactic information. This finding highlights the importance of syntactic information in tasks that are not explicitly syntactic in nature.

7.2 Future work

In what follows, we review relevant lines of future work related to the research topics explored along this dissertation, and outline some interesting subsequent research lines.

Testing the syntactic knowledge of models. In this thesis we have studied the differences between monolingual and multilingual models, and between the abilities of multilingual models across different languages. Overall, we have shown the importance of testing models on a wider range of languages, particularly morphologically rich ones. However, we limited our experiments to English and Spanish, covering a wide yet possibly incomplete range of phenomena. As part of our future work, we plan to assess the actual coverage of the test suites, extending them to ensure that important phenomena are not left out. For example, we will extend the Spanish tests to cover pronoun-dropping, a phenomena in which certain classes of pronouns may be omitted when they can be pragmatically or grammatically inferred. Moreover, we plan to develop SyntaxGyms for a number of other selected languages, such as Portuguese, Russian and Italian, extending the suites to cover language-specific phenomena, as we already did for Spanish by expanding the agreement suite and adding a whole new circuit regarding the linear order of a sentence's basic constituents.

Complementing stopping criteria during pretraining. Several works, including this dissertation, have shown that the syntactic structure emerges in pretrained language models, even though they are trained on raw text without supervision and are never exposed to any type of syntactic signal. The models are trained to maximize perplexity, a metric evaluating the

language modeling capabilities of the models. However, we have shown that there is no simple relationship between the perplexity of the models and their average syntactic generalisation capabilities: the variance in intra-family SyntaxGym score is not explained by the perplexity differences. Thus, an interesting question emerges: is it possible to complement information-theoretical metrics such as perplexity with metrics measuring specific types of knowledge, e.g., syntax? Can we find a specific set of probes covering different linguistic phenomena to be used as a pretraining stopping criteria? We hypothesise that complementing perplexity with complementary metrics could lead to an improvement in the encoding of the linguistic information on pretrained models, facilitating the development of more robust and efficient models to solve NLU tasks.

Multi-task pretraining In this work, we have analysed the benefits of MLM and NSP by comparing BERT-based and RoBERTa-based models in several downstream tasks. Our findings suggest that larger and more diverse training data can effectively compensate the lack of sentence-based objectives, as proved by RoBERTa-based models outstanding performance. However, our experiments also suggest that there may be a limit to the amount of data that can be feed into a RoBERTa model and the knowledge that the model can acquire. An alternative interesting work line to improve the linguistic capabilities of Transformer-based models is leveraging the knowledge form different pretraining objectives. For instance, in (Aribandi et al., 2021), a study of the effect of multi-task pretraining at the largest scale to date is presented. The authors argue that a massive and diverse collection of pretraining tasks is generally preferable to an expensive search for the best combination of pretraining tasks, because manually curating an ideal set of tasks for multi-task pretraining is not straightforward. However, we wonder whether it is possible to find a linguistically motivated combination of tasks that maximises the knowledge of the models. While manually curating a selection of tasks is indeed expensive, it may lead to the development of more robust models with a stronger encoding of linguistic knowledge, while allowing us to escape the *bigger is better* paradigm in which the field seems to be trapped.

Adding a linguistic dimension to tokenizers Tokenizers are in charge of splitting a text into words or subwords, which then are converted to ids through a look-up table. Transformer-based models use a hybrid approach between word-level and character-level tokenization called “ subword tokenization”, which allows them to have a reasonable vocabulary size while being able to learn meaningful context-independent representations. It also allows the models to process words they have never seen before, by decomposing them into known subwords. Subword tokenizers are trained in an unsupervised manner along with the model, and rely on two basic principles: 1) frequently used words should not be split into smaller subwords; and 2) rare words should be decomposed into meaningful subwords. However, even though it has been shown that tokenizers play an important role in the downstream performance of pretrained models (Rust et al., 2020), little is known about the linguistic differences between the most commonly used tokenization techniques, namely, WordPiece, SentencePiece and BPE. We hypothesise that a linguistically motivated tokenizer could help the models to better exploit the linguistic cues of the text, e.g. by correctly splitting roots from prefixes and suffixes, declination, etc, leading to an improvement of the linguistic knowledge encoded in the models. In order to shed some light on this matter, we plan to conduct a comparative study of current tokenization techniques to assess whether there are linguistic differences between them, and whether different techniques may be more useful when applied to process some languages than others. In addition, we aim at developing language-specific linguistically-motivated tokenizers and comparing them with current unsupervised ones.

7.3 Final remarks on the opportunities, dangers and limitations of pretrained models

The popularization of self-supervised learning with language models, in particular with Transformer-based architectures, marked the beginning of a revolution in the NLP field. This new successful transfer learning

paradigm of training one model on a huge amount of data and adapting it to many other applications is, of course, not exclusive of the NLP field, and has been widely adopted by the AI community. In an attempt to underscore their critically central yet incomplete character, such models are generally referred to as foundation models. Indeed, their use is so extended that the Stanford Institute for Human-Centered Artificial Intelligence (HAI) has created the Center for Research on Foundation Models (CRFM), an interdisciplinary initiative that aims to make fundamental advances in the study, development, and deployment of foundation models. In an extensive report entitled *On the opportunities and risks of foundation models* (Bommasani et al., 2021), the CRFM provides a thorough account of the opportunities and risks of such models, ranging from their capabilities (e.g., language, vision, robotics, reasoning, human interaction) and technical principles (e.g., model architectures, training procedures, data, systems, security, evaluation, theory) to their applications (e.g., law, healthcare, education) and societal impact (e.g., inequity, misuse, economic and environmental impact, legal and ethical considerations).

7.3.1 Dangers and limitations of big language models

As language models grow in terms of parameters and training data, so does the body of research concerned with the possible risks associated with this technology. A comprehensive study of the opportunities and risks of big language models can be found in (Bommasani et al., 2021). In particular, the authors warn about current lack of understanding of how foundation models work, when do they fail, and what are they capable of, warning about the dangers of homogenisation, as the defects of the foundation model are inherited by all the adapted models downstream. Along the same lines, Bender et al. (2021) offer a critical overview of the risks of relying on ever-increasing size of LMs as the primary driver of language technology, arguing for a reallocation of efforts towards approaches that avoid some of the associated risks while still benefiting of improvements to language technology. They provide recommendations including weighing the environmental and financial costs first, investing

resources into curating and carefully documenting datasets, evaluating how the planned approach fits into research and development goals and supports stakeholder values, and encouraging research directions beyond ever larger language models. In what follows, we offer a brief overview of the main concerns raised, and provide some useful pointers to further readings.

Misuse of language models. There are many harmful activities that rely on text, such as misinformation, fake-news, spam, phishing and fraudulent writing. Until not long ago, these applications relied on human beings to write plausible and natural text. However, current language models such as GPT-3 are already able to generate text that is difficult to distinguish from human-written text, and could lower existing barriers to perform these activities and increase their efficacy (Brown et al., 2020).

Economical and environmental cost. The computational costs of state-of-the art AI research has increased 300,000x in recent years, leading to a surprisingly large carbon footprint (Schwartz et al., 2019). Indeed, training and developing big language models requires large amounts of computation. In the last years, different works have highlighted the need for energy efficient model architectures and training paradigms to reduce negative environmental impact and inequitable access to resources, encouraging authors to report the financial cost of developing, training, and running models in order to provide baselines for the investigation of increasingly efficient methods; cf., e.g., Strubell et al. (2019); Schwartz et al. (2019); Bender et al. (2021). Since then, several recent works have embraced these guidelines, including reports on their energy usage and cost/benefit analysis (Brown et al., 2020; Pérez-Mayos et al., 2021; Austin et al., 2021; Wei et al., 2021). Also, Strubell et al. (2019) proposes actionable recommendations to reduce costs and improve equity, namely 1) reporting training time and sensitivity to hyperparameters; 2) a government-funded academic compute cloud to provide equitable access to all researchers; and 3) prioritising computationally efficient hardware and algorithms. To facilitate the detection of energy bottlenecks and the evaluation of the en-

ergy impact of different architectural choices, Cao et al. (2021) presents IrEne, an interpretable and extensible energy prediction system that accurately predicts the inference energy consumption of a wide range of Transformer-based NLP models.

Diversity, social views and bias in training data. The recent success of language models relies to a large extent in the huge amount of data used to train them, mostly extracted from the Internet. For example, the Common Crawl [1] consists of petabytes of data collected over 8 years of web crawling, and a filtered version of it is included in the GPT-3 training data. However, commonly used training data has been shown to have problematic characteristics resulting in models that encode stereotypical and derogatory associations along gender, race, ethnicity, and disability status (Hutchinson et al., 2020; Lu et al., 2020; Brown et al., 2020; Bender et al., 2021; Hovy and Prabhumoye, 2021). Bender et al. (2021) offers an in-depth analysis on how large, uncurated, static datasets from the Internet encode hegemonic views that are harmful to marginalised populations, and recommend significant resource allocation towards dataset curation and documentation practices. Several works analyse different sources of bias and explore ways of recognizing and mitigating them (Sun et al., 2019; Hovy and Prabhumoye, 2021; Bender et al., 2021).

Lack of accountability. Relying on ever larger datasets implies the risk of incurring in *documentation debt*, that is, where the datasets are both undocumented and too large to document, perpetuating the above-mentioned harms without recourse and difficulting their mitigation (Bender et al., 2021).

Are language models just stochastic parrots? As argued in (Bender and Koller, 2020), it is important to understand the capabilities and limitations of LMs. Although it may appear otherwise, LMs are not able to

[1] https://commoncrawl.org

perform Natural Language Understanding (NLU), and their success is restricted to tasks that can be approached by manipulating linguistic form (Bender and Koller, 2020; Bender et al., 2021). Pushing the criticism further, Noam Chomsky expressed his skepticism about GPT-3's scientific value: "It's not a language model. It works just as well for impossible languages as for actual languages. It is therefore refuted, if intended as a language model, by normal scientific criteria. [...] Perhaps it's useful for some purpose, but it seems to tell us nothing about language or cognition generally.".[2]

7.3.2 New research directions

Foundation models have drastically changed the practice of NLP (Section 2.5), and many tasks are now solved to an almost-human level using mainly foundation models such as BERT and RoBERTa. At the same time, foundation models have given rise to many new research directions:

Language variation. There are thousands of different languages in the world, with important variations even within one language (cf. dialects) or within one speaker (e.g., informal conversation compared to written language). However, it is not clear how successfully current pretrained models handle language variation, and it remains an open question whether it is possible to make foundation models that robustly and equitably represent language variations and their subtleties.

Multilinguality. A major challenge to modeling the more than 6.000 languages in the world is the lack of enough training data. Multilingual models rely on the assumption that the shared structures and patterns between languages can lead to sharing and transfer from the high-resourced languages (e.g., English) to the low-resources ones, making foundation models possible for languages where it is not possible to train a monolingual model. However, it remains unclear how much models trained

[2] https://www.youtube.com/watch?v=c6MU5zQwtT4

on this data can represent aspects of other languages that are drastically different from English, and it is not clear how much variation can fit in a single model.

Grounded language acquisition. Compared to human language acquisition, machine language acquisition is extremely inefficient, with current models being trained on around three to four orders of magnitude more data than most humans will ever hear or read. One of the main differences is the fact that human language is grounded to the real world, e.g. we learn the words to refer to common objects while pointing at them, and we learn new languages with the support of images and sounds. Thus, while ungrounded statistical learning from raw text is important, advancing grounded language learning for foundation models remains an important direction to improve language acquisition efficiency.

Generalisation. As humans, we learn language in a way that allows us to slot new knowledge into existing abstractions, and productively create new grammatical sentences. In contrast, foundation models often do not acquire such systematic abstractions. For example, when a model produces a linguistic construction accurately there is no guarantee that future uses of that construction will be consistent, and this is aggravated when applied to different domains. Making adaptable models able to mirror human-like linguistic adaptation and language evolution without relying on rigid linguistic rules is an open research area for the future of foundation models.

Following the current trend, we anticipate that foundation models will keep evolving and scaling over the next years, fueled by their huge commercial incentives. However, as warned in (Bommasani et al., 2021; Bender et al., 2021), this progress should be led not from industry alone, but in collaboration with governments and academia, with the common goal of establishing the norms that will enable the responsible research and deployment of foundation models, promoting their social benefit and mitigating their social harms.

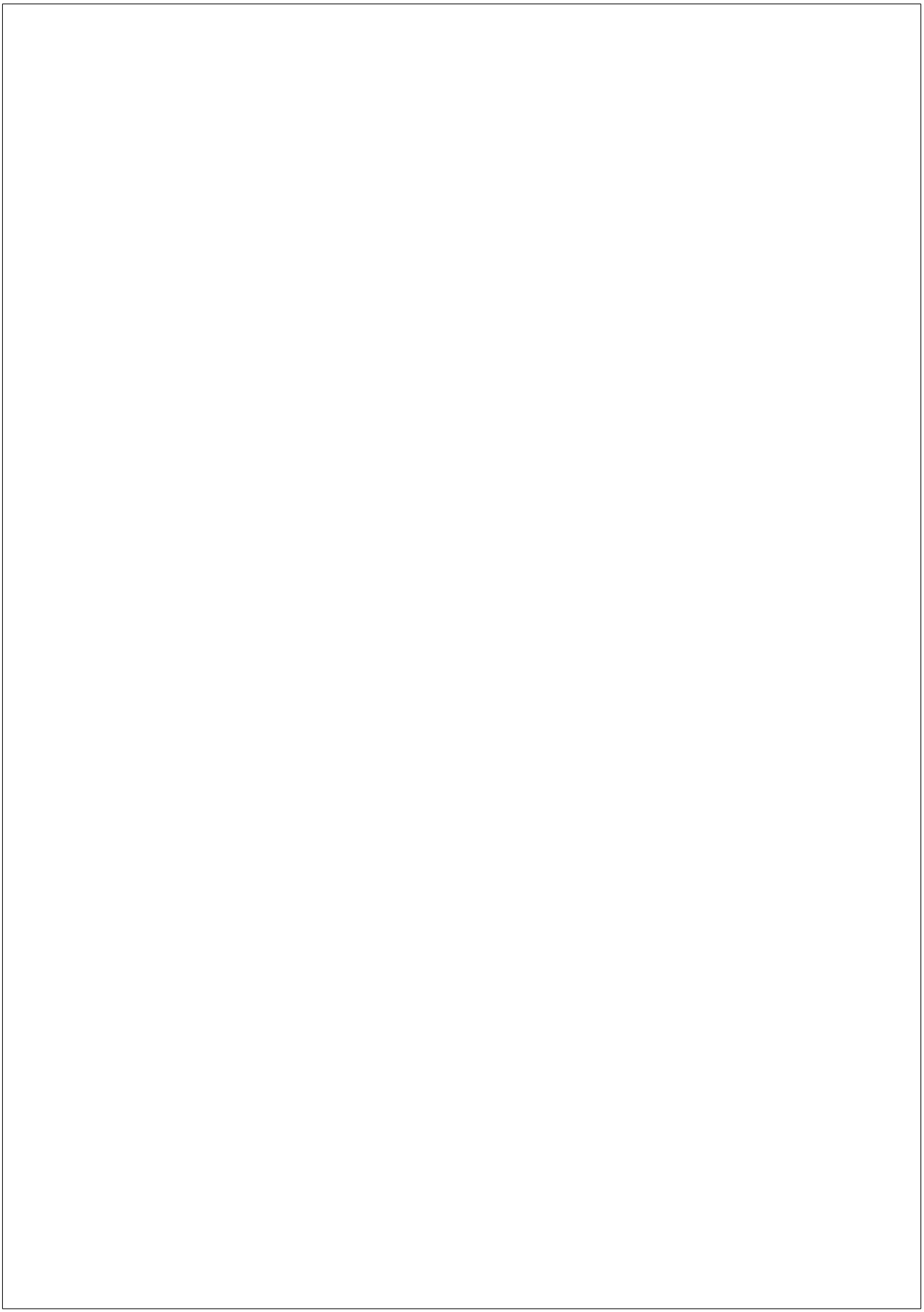
7.4 Publications

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

- **Pérez-Mayos, L.**; Carlini, R.; Ballesteros, M.; Wanner, L. On the evolution of syntactic information encoded by BERT's contextualised representations. In proceedings of the European Chapter of the Association for Computational Linguistics (EACL 2021).
- **Pérez-Mayos, L.**; Táboas García, A.; Mille, S.; Wanner, L. Assessing the Syntactic Capabilities of Transformer-based Multilingual Language Models. In findings of the Associ- ation for Computational Linguistics (ACL 2021).
- **Pérez-Mayos, L.**; Ballesteros, M.; Wanner, L. How much pretraining data do language models need to learn syntax?. In proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021).

Additionally, while not directly related to the contents of this dissertation, the following articles have also been published over the course of the doctorate program:

- Fortuna, P.; Cortez, V.; Sozinho Ramalho, M.; **Pérez-Mayos, L.** MIN_PT: An European Portuguese Lexicon for Minorities Related Terms. In proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021) on the ACL Anthology.
- Fortuna, P.; **Pérez-Mayos, L.**; AbuRa'ed, A.; Soler-Company, J.; Wanner, L. Cartography of Natural Language Processing for Social Good: Definitions, Statistics and White Spots. In proceedings of the 1st Workshop on NLP for Positive Impact (2021) on the ACL Anthology.



Bibliography

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.

Eleni Adamopoulou and Lefteris Moussiades. 2020. Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:100006.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Rodrigo Agerri, Iñaki San Vicente, Jon Ander Campos, Ander Barrena, Xabier Saralegi, Aitor Soroa, and Eneko Agirre. 2020. Give your text representation models some love: the case for Basque. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4781–4788, Marseille, France. European Language Resources Association.

Maria Antoniak and David Mimno. 2018. Evaluating the stability of embedding-based word similarities. *Transactions of the Association for Computational Linguistics*, 6:107–119.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q Tran, Dara Bahri, Jianmo Ni, et al. 2021. Ext5: Towards extreme multi-task scaling for transfer learning. *arXiv preprint arXiv:2111.10952*.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *ArXiv preprint*, abs/2108.07732.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv preprint*, abs/2004.05150.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 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*, pages 610–623.

Emily M. Bender and Alexander Koller. 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.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 932–938. MIT Press.

Kasturi Bhattacharjee, Miguel Ballesteros, Rishita Anubhai, Smaranda Muresan, Jie Ma, Faisal Ladhak, and Yaser Al-Onaizan. 2020. To BERT or not to BERT: Comparing task-specific and task-agnostic semi-supervised approaches for sequence tagging. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7927–7934, Online. Association for Computational Linguistics.

Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. 2010. A high-performance syntactic and semantic dependency parser. In *Coling 2010: Demonstrations*, pages 33–36, Beijing, China. Coling 2010 Organizing Committee.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4349–4357.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *ArXiv preprint*, abs/2108.07258.

Léon Bottou. 2012. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer.

Y LeCun L Bottou and Muller GO. 1998. K.: Efficient backprop. *Neural Networks: Tricks of the trade, Springer*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the tenth conference on computational natural language learning (CoNLL-X)*, pages 149–164.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

José Canete, Gabriel Chaperon, Rodrigo Fuentes, and Jorge Pérez. 2020. Spanish pre-trained BERT model and evaluation data. *PML4DC at ICLR*, 2020.

Qingqing Cao, Yash Kumar Lal, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. 2021. IrEne: Interpretable energy prediction for transformers. 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 2145–2157, Online. Association for Computational Linguistics.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems.

Wenlin Chen, David Grangier, and Michael Auli. 2016. Strategies for training large vocabulary neural language models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1975–1985, Berlin, Germany. Association for Computational Linguistics.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 160–167. ACM.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Bernard Comrie. 1989. *Language universals and linguistic typology: Syntax and morphology*. University of Chicago press.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 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.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems 28: Annual*

Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 3079–3087.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 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, Florence, Italy. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. 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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. 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.

Sumanth Doddapaneni, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M Khapra. 2021. A primer on pretrained multilingual language models. *ArXiv preprint*, abs/2107.00676.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *ArXiv preprint*, abs/2002.06305.

William B. Dolan and Chris Brockett. 2005a. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

William B. Dolan and Chris Brockett. 2005b. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880, Online. Association for Computational Linguistics.

Daniel Edmiston. 2020. A systematic analysis of morphological content in bert models for multiple languages.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2020. When bert forgets how to pos: Amnesic probing of linguistic properties and mlm predictions.

Allyson Ettinger. 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.

Richard Futrell, Ethan Wilcox, Takashi Morita, and Roger Levy. 2018. Rnns as psycholinguistic subjects: Syntactic state and grammatical dependency.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018a. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018b. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020a. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020b. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Asso-*

ciation for Computational Linguistics: System Demonstrations, pages 70–76, Online. Association for Computational Linguistics.

Jon Gauthier and Roger Levy. 2019. Linking artificial and human neural representations of language. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 529–539, Hong Kong, China. Association for Computational Linguistics.

Yoav Goldberg. 2017. *Neural Network Methods for Natural Language Processing*, volume 37 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool, San Rafael, CA.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *ArXiv preprint*, abs/1901.05287.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Vikrant Goyal, Anoop Kunchukuttan, Rahul Kejriwal, Siddharth Jain, and Amit Bhagwat. 2020. Contact relatedness can help improve multilingual NMT: Microsoft STCI-MT @ WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 202–206, Online. Association for Computational Linguistics.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *ArXiv preprint*, abs/1308.0850.

Yuxian Gu, Zhengyan Zhang, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. 2020. Train no evil: Selective masking for task-guided pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6966–6974, Online. Association for Computational Linguistics.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 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.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Abram Handler. 2014. An empirical study of semantic similarity in wordnet and word2vec.

James Henderson. 2020. The unstoppable rise of computational linguistics in deep learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6294–6306, Online. Association for Computational Linguistics.

John Hewitt and Percy Liang. 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.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *ArXiv preprint*, abs/1503.02531.

Daniel Hládek, Ján Staš, and Matúš Pleva. 2020. Survey of automatic spelling correction. *Electronics*, 9(10):1670.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Dirk Hovy and Shrimai Prabhumoye. 2021. Five sources of bias in natural language processing. *Language and Linguistics Compass*, 15(8):e12432.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020a. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020b. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882.

Ben Hutchinson, Vinodkumar Prabhakaran, Emily Denton, Kellie Webster, Yu Zhong, and Stephen Denuyl. 2020. Social biases in NLP models as barriers for persons with disabilities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5501, Online. Association for Computational Linguistics.

Folasade Olubusola Isinkaye, Yetunde O Folajimi, and Bolande Adefowoke Ojokoh. 2015. Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal*, 16(3):261–273.

Anna A Ivanova, John Hewitt, and Noga Zaslavsky. 2021. Probing artificial neural networks: insights from neuroscience. *ArXiv preprint*, abs/2104.08197.

Derry Jatnika, Moch Arif Bijaksana, and Arie Ardiyanti Suryani. 2019. Word2vec model analysis for semantic similarities in english words. *Procedia Computer Science*, 157:160–167.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 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, Florence, Italy. Association for Computational Linguistics.

Mandar Joshi, Kenton Lee, Yi Luan, and Kristina Toutanova. 2020a. Contextualized representations using textual encyclopedic knowledge.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020b. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. inlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained

multilingual language models for indian languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4948–4961.

Minki Kang, Moonsu Han, and Sung Ju Hwang. 2020. Neural mask generator: Learning to generate adaptive word maskings for language model adaptation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6102–6120, Online. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. 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 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. Dependency parsing. *Synthesis lectures on human language technologies*, 1(1):1–127.

Artur Kulmizev, Vinit Ravishankar, Mostafa Abdou, and Joakim Nivre. 2020. Do neural language models show preferences for syntactic formalisms? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4077–4091, Online. Association for Computational Linguistics.

Adhiguna Kuncoro, Lingpeng Kong, Daniel Fried, Dani Yogatama, Laura Rimell, Chris Dyer, and Phil Blunsom. 2020. Syntactic structure distillation pretraining for bidirectional encoders. *Transactions of the Association for Computational Linguistics*, 8:776–794.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 41(5):1202–1241.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2020a. FlauBERT: Unsupervised language model pre-training for French. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2020b. FlauBERT: Unsupervised language model pre-training for French. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. 2006. A tutorial on energy-based learning. *Predicting structured data*, 1(0).

Yann LeCun and Fu Jie Huang. 2005. Loss functions for discriminative training of energy-based models. In *International Workshop on Artificial Intelligence and Statistics*, pages 206–213. PMLR.

Michael Lepori and R. Thomas McCoy. 2020. Picking BERT’s brain: Probing for linguistic dependencies in contextualized embeddings using representational similarity analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3637–3651, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2177–2185.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for

natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Junlong Li, Zhuosheng Zhang, Hai Zhao, Xi Zhou, and Xiang Zhou. 2020. Task-specific objectives of pre-trained language models for dialogue adaptation. *ArXiv preprint*, abs/2009.04984.

George James Lidstone. 1920. Note on the general case of the bayes-laplace formula for inductive or a posteriori probabilities. *Transactions of the Faculty of Actuaries*, 8(182-192):13.

Tomasz Limisiewicz, David Mareček, and Rudolf Rosa. 2020. Universal Dependencies According to BERT: Both More Specific and More General. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2710–2722, Online. Association for Computational Linguistics.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT’s linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Tal Linzen and Marco Baroni. 2020. Syntactic structure from deep learning. *ArXiv preprint*, abs/2004.10827.

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

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

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Pablo Loyola, Edison Marrese-Taylor, and Yutaka Matsuo. 2017. A neural architecture for generating natural language descriptions from source code changes. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 287–292, Vancouver, Canada. Association for Computational Linguistics.

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2020. Gender bias in neural natural language processing. In *Logic, Language, and Security*, pages 189–202. Springer.

Christopher D Manning. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *International conference on intelligent text processing and computational linguistics*, pages 171–189. Springer.

Christopher D Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.

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

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 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.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

James L McClelland, David E Rumelhart, and Geoffrey E Hinton. 1986. The appeal of parallel distributed processing. *MIT Press, Cambridge MA*, pages 3–44.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 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, Online. Association for Computational Linguistics.

Alessio Miaschi and Felice Dell'Orletta. 2020. Contextual and non-contextual word embeddings: an in-depth linguistic investigation. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 110–119, Online. Association for Computational Linguistics.

Vincent Micheli, Martin d'Hoffschmidt, and François Fleuret. 2020. On the importance of pre-training data volume for compact language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7853–7858, Online. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *ArXiv preprint*, abs/1301.3781.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

David Mimno and Laure Thompson. 2017. The strange geometry of skip-gram with negative sampling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2873–2878, Copenhagen, Denmark. Association for Computational Linguistics.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G Azzolini, et al. 2019. Deep learning recommendation model for personalization and recommendation systems. *arXiv preprint arXiv:1906.00091*.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Nils J Nilsson. 2009. *The quest for artificial intelligence*. Cambridge University Press.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The conll 2007 shared task on dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 915–932.

Joakim Nivre, Daniel Zeman, Filip Ginter, and Francis Tyers. 2017. Universal Dependencies. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Valencia, Spain. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 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.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. 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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018b. 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.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 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.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020a. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online. Association for Computational Linguistics.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020b. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online. Association for Computational Linguistics.

Laura Pérez-Mayos, Miguel Ballesteros, and Leo Wanner. 2021. How much pretraining data do language models need to learn syntax?

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018a. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018b. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Sello Ralethe. 2020. Adaptation of deep bidirectional transformers for Afrikaans language. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2475–2478, Marseille, France. European Language Resources Association.

Prajit Ramachandran, Peter Liu, and Quoc Le. 2017. Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 383–391, Copenhagen, Denmark. Association for Computational Linguistics.

Delip Rao, Paul McNamee, and Mark Dredze. 2013. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, multilingual information extraction and summarization*, pages 93–115. Springer.

Vinit Ravishankar, Artur Kulmizev, Mostafa Abdou, Anders Søgaard, and Joakim Nivre. 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, Online. Association for Computational Linguistics.

Reza Rawassizadeh, Taylan Sen, Sunny Jung Kim, Christian Meurisch, Hamidreza Keshavarz, Max Mühlhäuser, and Michael Pazzani. 2019. Manifestation of virtual assistants and robots into daily life: Vision and challenges. *CCF Transactions on Pervasive Computing and Interaction*, 1(3):163–174.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Samuel Rönnqvist, Jenna Kanerva, Tapio Salakoski, and Filip Ginter. 2019. Is multilingual BERT fluent in language generation? In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 29–36, Turku, Finland. Linköping University Electronic Press.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2020. How good is your tokenizer? on the monolingual performance of multilingual language models. *arXiv preprint arXiv:2012.15613*.

Devendra Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. 2021a. Do syntax trees help pre-trained transformers extract information? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2647–2661, Online. Association for Computational Linguistics.

Devendra Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. 2021b. Do syntax trees help pre-trained transformers extract information? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2647–2661, Online. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv preprint*, abs/1910.01108.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2019. Green ai. *ArXiv preprint*, abs/1907.10597.

Claude Elwood Shannon. 1948. A mathematical theory of communications. *Bell Syst. Tech. J.*, 27:379–423.

Wei Shen, Yuhan Li, Yinan Liu, Jiawei Han, Jianyong Wang, and Xiaojie Yuan. 2021. Entity linking meets deep learning: Techniques and solutions. *IEEE Transactions on Knowledge and Data Engineering*.

Wei Shen, Jianyong Wang, and Jiawei Han. 2014. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *ArXiv preprint*, abs/1904.05255.

Heung-Yeung Shum, Xiao-dong He, and Di Li. 2018. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1):10–26.

Clauirton Siebra, Walter Correia, Marcelo Penha, Jefte Macedo, Jonysberg Quintino, Marcelo Anjos, Fabiana Florentin, Fabio QB Da Silva, and Andre LM Santos. 2018. Virtual assistants for mobile interaction: A review from the accessibility perspective. In *Proceedings of the 30th Australian Conference on Computer-Human Interaction*, pages 568–571.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014a. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014b. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Anders Søgaard. 2021. Explainable natural language processing. *Synthesis Lectures on Human Language Technologies*, 14(3):1–123.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1640, Florence, Italy. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Marten van Schijndel, Aaron Mueller, and Tal Linzen. 2019. Quantity doesn't buy quality syntax with neural language models. 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 5831–5837, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

David Vilares, Michalina Strzyz, Anders Søgaard, and Carlos Gómez-Rodríguez. 2020. Parsing as pretraining. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9114–9121. AAAI Press.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3156–3164. IEEE Computer Society.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Alex Wang and Kyunghyun Cho. 2019. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, Minneapolis, Minnesota. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 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.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yufei Wang, Mark Johnson, Stephen Wan, Yifang Sun, and Wei Wang. 2019b. How to best use syntax in semantic role labelling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5338–5343, Florence, Italy. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020a. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020b. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *ArXiv preprint*, abs/2109.01652.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.

Laura Wendlandt, Jonathan K. Kummerfeld, and Rada Mihalcea. 2018. Factors influencing the surprising instability of word embeddings. 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 2092–2102, New Orleans, Louisiana. Association for Computational Linguistics.

Ethan Wilcox, Roger Levy, and Richard Futrell. 2019a. Hierarchical representation in neural language models: Suppression and recovery of expectations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 181–190, Florence, Italy. Association for Computational Linguistics.

Ethan Wilcox, Roger Levy, and Richard Futrell. 2019b. Hierarchical representation in neural language models: Suppression and recovery of expectations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 181–190, Florence, Italy. Association for Computational Linguistics.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Ethan Wilcox, Peng Qian, Richard Futrell, Miguel Ballesteros, and Roger Levy. 2019c. Structural supervision improves learning of non-local grammatical dependencies. 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 3302–3312, Minneapolis, Minnesota. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma,

Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020a. 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. Association for Computational Linguistics.

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

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv preprint*, abs/1609.08144.

Zenan Xu, Daya Guo, Duyu Tang, Qinliang Su, Linjun Shou, Ming Gong, Wanjun Zhong, Xiaojun Quan, Daxin Jiang, and Nan Duan. 2021. Syntax-enhanced pre-trained model. 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 5412–5422, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Yian Zhang, Alex Warstadt, Xiaocheng Li, and Samuel R. Bowman. 2021. When do you need billions of words of pretraining data? 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 1112–1125, Online. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society.