FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

DOCTORAL THESIS

Tomasz Limisiewicz

Interpreting and Controlling Linguistic Features in Multilingual Language Models

Institute of Formal and Applied Linguistics

Supervisor: David Mareček

Study Program:	Computer Science
Specialization:	Computational Linguistics

Prague 2024

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

Prague, July 6, 2024 Tomasz Limisiewicz

Title: Interpreting and Controlling Linguistic Features in Multilingual Language Models

Author: Tomasz Limisiewicz

Department: Institute of Formal and Applied Linguistics

Supervisor: David Mareček,
Institute of Formal and Applied Linguistics

Abstract:

Language models based on neural networks have become the foundation for solving diverse tasks, yet their inner workings remain opaque. This dissertation investigates which components of language models are crucial for representing and processing information from texts. We mainly focus on multilingual models that can reuse representation for processing tasks in different languages. We hypothesize that understanding how models represent linguistic phenomena is necessary to control their function and alleviate issues hindering models' performance. To this end, we propose novel interpretability techniques to analyze specific components of language models. Our approaches enable the localization of the representation of distinct types of signals within the language models. The localization allowed us to contain our analysis to specific modules and apply precise interventions to mitigate gender bias or improve cross-lingual transfer.

Keywords: natural language processing, language models, multilinguality, interpretable machine learning

Název práce: Interpretace a přizpůsobování jazykových jevů ve vícejazyčných modelech

Autor: Tomasz Limisiewicz

Katedra: Ústav formální a aplikované lingvistiky

Vedoucí práce: David Mareček,
Ústav formální a aplikované lingvistiky

Abstrakt:

Jazykové modely založené na neuronových sítích se staly základem pro řešení nejrůznějších úloh, jejich vnitřní fungování však zůstává nejasné. Tato disertační práce zkoumá, které komponenty jazykových modelů jsou klíčové pro reprezentaci a zpracování textových informací. Zaměřujeme se především na vícejazyčné modely, které mohou využívat reprezentaci pro zpracování úloh napříč různými jazyky. Předpokládáme, že pochopení toho, jak modely reprezentují jazykové jevy, je nezbytné pro jejich následné přizpůsobování a zmírňování problémů, kterými tyto modely trpí. Za tímto účelem navrhujeme nové techniky pro interpretaci jednotlivých komponent jazykových modelů. Naše metody umožňují lokalizovat reprezentaci různých typů signálů v těchto modelech. Tato lokalizace nám umožňuje omezit naši analýzu na konkrétní komponenty a aplikovat cílené opravy modelů ke zmírnění genderové zaujatosti nebo zlepšení mezijazykového přenosu.

Klíčová slova: zpracování přirozeného jazyka, jazykové modely, vícejazyčnost, interpretovatelné strojové učení

Acknowledgements

Special thanks to:

and all others not pictured who helped me immensely along the way.

This work has been using language resources and tools developed and/or stored and/or distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071). My research and studies have been supported by: Grant 23-06912S of the Czech Science Foundation, Grant 338521 of the Charles University Grant Agency, Fellowship from Paul G. Allen School of Computer Science, Karel Urbánek Fellowship, and the Mobility Fund of Charles University.

Contents

1
Introduction

The recent success of neural networks in the field of natural language processing (NLP) has brought about their wide adaptation in society. Applications relying on language models based on neural networks are today ubiquitous, to name a few: machine translation, text generators, chatbots, and information search. However, the fact that the way these models process language is not understood is a cause of concern (Rudin, 2019; Bender et al., 2021; Bommasani et al., 2021). The models can be affected by the unwanted harmful information and biases stemming from the data they were trained on (Caliskan et al., 2017; Bolukbasi et al., 2016; Van Der Wal et al., 2022). To control unwanted functionality of the models, we first need to identify their extent and the components responsible for encoding them (Bau et al., 2019; Brandl et al., 2023).

A major obstacle in pursuing the interpretation of NLP models is the fast-paced development in NLP. Notably, the field has massively changed since the beginning of my PhD studies four years ago. However, within the broad pool of ever-rising metrics: terabytes of textual data, billions of model parameters, and scores across diverse benchmarks, there are a few aspects that remain constant. First of them is perhaps surprisingly the structure of neural networks, which are still based on the Transformer architecture (Vaswani et al., 2017). The second is the training objective, which is a prediction of the unseen tokens (or simply words) in a sentence (Bengio et al., 2003). The objective is realized either by a prediction of the sentence completion (autoregression, Radford and Narasimhan, 2018) or by recovering the masked tokens inside the sentence (autoencoding, Devlin et al., 2019).

My PhD research pursued the goal of understanding the astounding success of the language models following these two principles. To simplify the task, we divide the Dissertation into analyses of each component of the models. Our studies focus on the following aspects, which we analyze for each of the components:

Linguistic Background We studied how neural networks learn language-relevant information and represent it. We find it crucial to focus on linguistic phenomena in language corpora and the ways language models apprehend them (Tenney et al., 2019), such as understanding syntactic, semantic, and discourse cues. The understanding of linguistics properties serves as an intermediate step for resolving the more complex tasks, that define the utility of the NLP models.

Multilinguality Another motivation for our research is the investigation of the applicability of the models across multiple languages. It has been observed, that the models can be adapted to tasks in unseen languages with a relatively small amount of in-language data (Wu and Dredze, 2019; Pires et al., 2019). We particularly focused on the cross-lingual transfer of knowledge, i.e. sharing of information across languages.

Methods for Improvement Lastly, we focus on the practical outcomes of the research. Specifically, we propose to make precise interventions in the models to improve their performance in specific domains and tasks. We focus on the applications to low-resource languages and underrepresented domains that cannot be simply resolved by the increase of the model size or data amount.

1.1 Structure

The motivation of the research articulated above, is addressed step by step through in-depth analysis and adaptions of the models. To structure the outcomes of our research, we formulate the general guiding questions that are successively answered in the following chapters. The first two questions help us define the scope of the models and the tools we use throughout our experiments. To comprehensively answer them we refer to the past literature:

1. **How are language models built?** We go over the Transformer architecture in detail and overview the design choices characteristic to recent language models implementations. We take a closer look at the training objective of language modeling: token prediction based on context (Bengio et al., 2003; Mikolov, 2014). We delineate the two main types of neural language models that realize this objective: autoregressive (Radford and Narasimhan, 2018) and autoencoding models (Devlin et al., 2019; Liu et al., 2020b).

2. **What do language models learn?** We present the common evaluation benchmark of information learned by the models. We introduce the benchmarks gradually from the ones evaluating the presence of linguistic features: syntax, lexicon, and entity recognition. We then move to the more complex tasks involving semantics, and discourse, and finally, discuss language understanding challenges that require world knowledge and basic reasoning.

The following two chapters provide a thorough survey of the literature on language models. To address the first question in Chapter 2 we described the typical structure of Transformer model and how it changed in the recent implementations. Subsequently, in Chapter 3 we turn our attention to the second question. To answer this question, we survey language features that LMs learn and the benchmark used to evaluate them. We then move to the open research questions that are addressed by our novel methods and experiments. The answers to these questions make up the core of the dissertation and a comprehensive summary of my PhD research:

3. **How do language models learn?** The main motivation of the dissertation is to shed light on the learning process of the models. We specifically, focus on disentangling and tracing the signals to identify the components responsible for the model's "understanding" of particular aspects of language. For that purpose, we apply and develop methods for model interpretation: attention head heatmaps (Raganato and Tiedemann, 2018; Marecek and Rosa, 2019), probing tasks (Hewitt and Manning, 2019; Vig and Belinkov, 2019; Belinkov, 2022b), and causal tracing and intervention (Vig et al., 2020; Meng et al., 2022). We have further developed these methods to identify the signals with higher resolution and accuracy. Throughout the thesis, we answer this question by analyzing the components of the models and noting down the findings in the following format:

Finding 0

The key observations of our research are spelled in red text boxes.

4. **How can we improve language models?** We build upon the understanding of the model components to propose the targeted improvement of the models. As stated in the research motivations above, we focus on the specific artifacts of the models that cannot be improved by the fitting-all approach, such as scaling the model size or data amount. We aim to improve the multilingual capabilities of the analyzed models. Another focus is knowledge modification, which has an impactful aspect in the erasure of unwanted harmful or stereotypical signals learned by the model, most importantly gender bias (Stanczak and Augenstein, 2021; Nangia et al., 2020b; De-Arteaga et al., 2019). In the following sections, we present our contributions in the following format:

Innovation 0

Our methods and improvements are distinguished in green text boxes.

Starting from Chapter 4, we walk through each component and representation of Transformer: latent embeddings (Chapter 4), feed-forwards (Chapter 5), attention heads (Chapter 6), and input and output layers (Chapter 7). In each of these chapters, we describe our methodologies for interpreting encoded information and its significance to language model quality, addressing the third research question. In reference to the fourth question, we propose model adaptation methods aimed at improving their performance throughout diverse languages or mitigating biases.

Finally, in Chapter 8 we discuss our findings in the context of previous and subsequent research outcomes in the field. Chapter 9 summarizes our contributions, outlines the direction for future research, and lists the contribution of collaborators in the experimental part of the thesis.

1.2 Contributions

The contribution of this work is organizing and synthesizing our research outcomes presented in the array of research articles. We list them here in chronological order:

1. **Head ensembling** the method aggregating the attention heads into the sets capturing specific syntactic phenomena in English and across languages (Limisiewicz et al., 2020). The method is described in Chapter 6. The original paper was published in ACL Findings 2020.

2. **Orthogonal Probing** in this method we propose to probe for structural linguistic signals and show how their representations are distributed in latent representation (Limisiewicz and Mareček, 2021b). Further, we use this method to explore information sharing across languages (Limisiewicz and Mareček,

2021a) and the ability to erase unwanted signals without harming the models' overall performance (Limisiewicz and Mareček, 2022). Chapter 4 summarizes orthogonal probing and its applications. The mentioned papers were published in ACL 2021, EMNLP 2021, and Gender Bias in NLP workshop respectively.

3. **Multilingual Vocabulary Analysis** we thoroughly studied the multilingual vocabularies obtained by popular tokenizers (Kudo and Richardson, 2018; Sennrich et al., 2016) Our findings highlight the importance of allocating tokens across languages for better performance throughout end-tasks. We also show that sharing vocabulary entries across languages has varied impacts on cross-lingual transfer across different tasks (Limisiewicz et al., 2023a). The contribution is presented in Chapter 7, the mentioned work was published in Findings of ACL 2023.

4. **Causal tracing and model adaptation for debiasing** the method for identifying the components responsible for encoding gender bias with higher resolution and accuracy. We show that this observation allows us to surgically intervene in the models' feed-forward layers in order to erase the unwanted signals (Limisiewicz et al., 2023b). We describe the analysis and debiasing method in Chapter 5. The research article appeared at ICLR 2024.

2
How Are Language Models Build?

In this chapter, we introduce the basic technical concepts and components of the Transformer model (Vaswani et al., 2017), which is at the time of writing this thesis the dominant architecture used for language models and other natural language processing systems. We survey the key components of the model and describe how they are combined in the model. Subsequently, we present some of the most popular implementations of the Transformer models that vary in architecture and applications.

2.1 Terminology

In this section, we will denote input and output sequences of a language model (LM) as $\vec{x}= (x_1, x_2, \ldots, x_n)$ and $\vec{y}= (y_1, y_2, \ldots, y_m)$, respectively.

We use $\vec{h}= (h_1, h_2 \ldots, h_n)$ to denote the sequence of latent embeddings. We will refer to the embedding at position t as h_t, and in some cases, we will use $h_{t,i}$ to denote the i-th dimension of the hidden representation at position t. When we consider a model layer with latent embeddings in input and output, we will use $\vec{h}'$ to denote the output representation of the layer. We will denote linear transformation matrices in a network, as W and biases as b, i.e., a dense layer is given by equation: $h' = W \cdot h + b$.

2.2 Transformer

A Transformer is a deep neural network, i.e., composed of multiple layers. Similarly to previous deep architecture, it consists of a series of linear matrix multiplication and non-linear activation functions between them. The most distinct feature of Transformers is their reliance on the attention mechanism (Bahdanau et al., 2015a):

2.2.1 Attention

The role of the attention (Att.) is to efficiently disseminate the information across the context in sequential input. Attention allows passing the information across the representations further apart in a sequence and in a more efficient way than the recurrent neural networks (Hochreiter and Schmidhuber, 1997; Cho et al., 2014), which are sequential.

In the attention layers, three types of representation are computed for each element of the sequence's latent representation $\vec{h}$: query, key, and value.

$$\begin{aligned} q_t &= W_Q \cdot h_t, \\ k_t &= W_K \cdot h_t, \\ v_t &= W_V \cdot h_t \end{aligned} \tag{2.1}$$

The key and queries are used to compute the attention scores. The softmax is applied row-wise to the dot product matrix of keys and queries to obtain the weights (that sum to one) for averaging value vectors at each position of the sequence.

$$h' = \text{Attention}(Q, K, V)(h) = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_{\text{model}}}}\right) \cdot V \tag{2.2}$$

In this equation, d_{model} is the dimensionality of the hidden vectors. Matrices Q, K, V are column concatenations of the query, key, and value vectors in Equation 2.1.

In each layer, the attention is applied n times in parallel, as so-called *attention head*s. The outputs of all heads are concatenated into a single output vector:

$$\begin{aligned} h' = \text{MultiHeadAttention}(Q, K, V)(h) = \\ = \text{Concat}(\text{Attention}(Q_i, K_i, V_i)(h) \dots \text{Attention}(Q_n, K_n, V_n)(h)) \end{aligned} \tag{2.3}$$

Therefore, the whole Transformer contains $H \times L$ *attention head*s, where L is the number of layers and H is the number of heads.

Due to the necessity to compute the attention value for each pair of sequence elements the complexity of the attention is $O(n^2 \cdot d_{\text{model}})$. The presence of the n^2 term makes the attention computationally costly for longer sequences. However, multiple solutions have been proposed to mitigate this issue, to name a few: caching, flash attention (Dao et al., 2022), or linear approximations (Choromanski et al., 2021).

Cross-Attention In sequence-to-sequence models, cross-attention is used to pass the representation from the encoder to the decoder. This is done by taking the query from the decoder corresponding layer and the key and value from the encoder layer.

Self-Attention In other cases, all of the query, key, and value vectors are computed based on the same hidden vector, i.e. output of the previous layer.

2.2.2 Feed Forward

Feed-forward (FF) are the most common type of layer applied in neural networks. In Transformer, feed-forwards have wider inner activation dimension. Dimension widening is realized by the following steps: first, the hidden vectors ($\vec{h}$) are projected into a higher dimension; second, a non-linear activation function is applied; and last, the latent representation is projected back to the original dimensionality ($\vec{h'}$). The reason for such a design is to use more expressive use of the non-linear activations in the more dimensional space. The mathematical formula for the feed-forward layers is straightforward:

$$h'_t = \text{FF}(h_t) = W_{out} \cdot \sigma(W_{in} \cdot h_t + b_1) + b_2 \tag{2.4}$$

Where $W_{in} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$ projects the hidden vector to a higher dimension, while $W_{out} \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$ restoes the original dimensionality. b_1, b_2 are bias vectors and σ is a non-linear activation function, ReLU (Glorot et al., 2011) or other function depending on the particular implementation of Transformer architecture.

In recent large language models, the feed-forward usually takes up most of the model parameters. Past works have investigated feed-forward as potentially the most capable of storing the world knowledge learned by models (Geva et al., 2021; Meng et al., 2022).

2.2.3 Layer Normalization

Layer normalization (LN) is a technique used to stabilize the training thanks to the normalization of the output distribution in intermediate layers (Ba et al., 2016). Layer normalization (LN) is among the basic components of the Transformer, applied after each Att. and FF layer. Its coefficients are computed across the hidden dimension for each latent embedding in a sequence h_t:

$$\begin{aligned} \mu_{LN,t} &= \frac{1}{d_{\text{model}}} \sum_{i=1}^{d_{\text{model}}} h_{t,i} \\ \sigma_{LN,t} &= \sqrt{\frac{1}{d_{\text{model}}} \sum_{i=1}^{d_{\text{model}}} (h_{t,i} - \mu_{LN,t})^2} \end{aligned} \tag{2.5}$$

The normalization is then applied in each forward pass:

$$h'_t = \text{LayerNorm}(h_t) = \frac{h_t - \mu_{LN,t}}{\sigma_{LN,t}} \tag{2.6}$$

Unlike batch normalization, layer normalization is computed for each element and can be applied in an online regime, i.e. one element at a time.

2.2.4 Vocabulary and Input Embeddings

Each of the tokens in the input sequence is looked up in the input embedding matrix $W_{\text{embedding}}$. The embedding matrix maps each element of the a priori set vocabulary to a fixed-dimensional vector, so $W_{\text{embedding}}$ dimensionality is $\mathbb{R}^{v \times d_{\text{model}}}$. The lookup is easily performed by multiplying one-hot vectors (with ones at the token index in the vocabulary) by the embedding matrix:

$$we_t = \text{Embedding}(x_t) = W_{\text{embedding}} \cdot \mathbb{1}_{i=index(x_t)} \tag{2.7}$$

In Chapter 7, we will take a closer look at the methods for allocating constructing vocabularies and allocating parameters in the embedding matrix.

Positional Embeddings The position of the token in the sequence is encoded by adding a fixed positional embedding to the token embedding. The positional embeddings for each position are fixed (i.e., not changed during training) across all the positions t up to the maximum sequence length, by the following formulas:

$$
\begin{aligned}
pe_{t,2i} &= \sin\left(\frac{t}{10000^{2i/d_{\text{model}}}}\right) \\
pe_{t,2i+1} &= \cos\left(\frac{t}{10000^{2i/d_{\text{model}}}}\right)
\end{aligned}
\tag{2.8}
$$

Please note how the values for odd $(2i+1)$ and even $(2i)$ dimensions of the embedding are computed by distinct functions. The token embeddings and positional embedding are summed together, we denote it as the first hidden layer of the model:

$$h_t = \text{Embedding}(x_t) + pe_t \tag{2.9}$$

The usage of position encoding is crucial in the Transformer models, as it is the only way to distinguish repeating instances of the same token in the attention mechanism (e.g. this sentence would have four instances of the word *the*). The positional embeddings also allow identifying the order of the tokens in the input sequence. The main drawback of positional embeddings is the inability to generalize their values for sequences longer than the a priori set maximum sentence length.

2.2.5 Output Linear Layer

The last layer maps the hidden representation to the output spaces. In language modeling tasks, the output space is the vocabulary, and we denote its size as v. The computation is straightforward, a linear layer followed by a softmax (SM) activation function.

$$l_t = W_{\text{softmax}} \cdot h_t \tag{2.10}$$

Where $W_{\text{softmax}} \in \mathbb{R}^{d_{\text{model}} \times v}$ and l corresponds to the logits for each token in the vocabulary. Usually, the softmax layer is the same as the token embedding layer, i.e. $W_{\text{softmax}} = W_{\text{embedding}}$. This procedure called *weight tying* reduces the number of parameters in the model and was proven to be beneficial for the model's performance (Press and Wolf, 2017). Subsequently, the softmax function is applied to obtain the probability distribution for the predicted token:

$$P(y_t|y_{<t}) = \text{softmax}(l) = \frac{e^{l_{t,i}}}{\sum_{j=1}^{v} e^{l_{t,j}}} \tag{2.11}$$

when applying the language model to the classification task through fine-tuning, the output layer is replaced with a linear layer mapping the hidden representation to the number of classes.

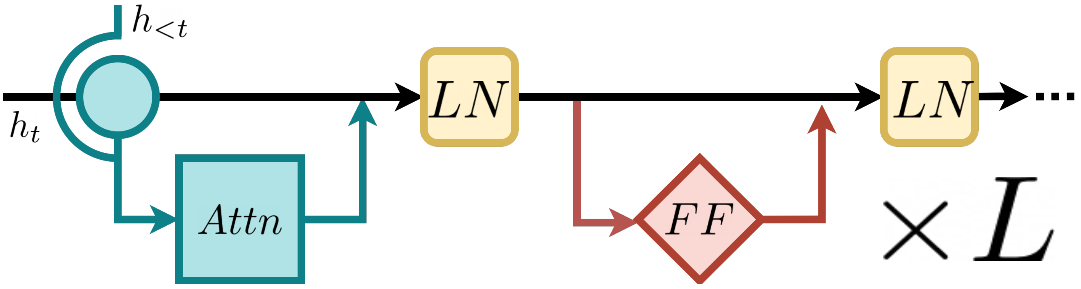


Figure 2.1: A schema showing the typical autoregressive Transformer layer. It consists of a self-attention layer and a feed-forward layer. Each of those components is followed by a layer normalization and a parallel residual connection. Such layer repeats L times and makes up the core of Transformer models.

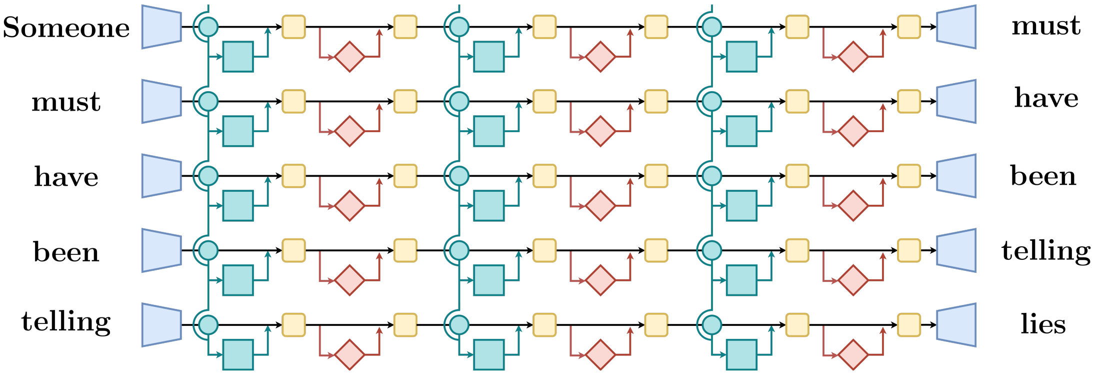


Figure 2.2: A schema showing the decoder of Transformer model trained for next token prediction (we assume that each word is represented as one token). The colors and shapes of the modules correspond to the ones in Figure 2.1. We also present input embeddings and output linear layers as blue trapezoids.

2.2.6 Bringing it All Together

Figure 2.1 presents how the aforementioned components are combined into a single layer of the Transformer model. The layers are stacked on top of each other, with the output of the previous layer being the input to the next one. Each attention and feed-forward is accompanied by a parallel residual connection, which adds the input of the layer to its output. The composition of multiple layers is presented in summary Figure 2.2. The figure also includes look-up layers, responsible for assigning input embeddings to the input tokens and the output linear layer with softmax that projects the hidden embeddings into the vocabulary domain.

The architecture of the Transformer layers can vary slightly across different models. For instance, in GPT-Neox (Black et al., 2022), attention and feed-forward are combined in parallel instead of sequentially.

2.3 Model Families

We distinguish three main Transformer model families based on the arrangement and usage of dense layers: encoder-decoder models, autoregressive models, and autoencoding models. We now give a brief overview of the most popular implementations for each of the presented model families.

2.3.1 Encoder-Decoder Models

The original Transformer introduced in (Vaswani et al., 2017) is an **encoder-decoder** model. That means it is composed of two separate stacks of layers, one for encoding input sequence and the other for predicting new sequence, which can be the sentence continuation or translation to another language. Specifically, this type of model employs cross-attention to pass information from the encoder to the decoder (see Section 2.2.1) This method of operation is also known as sequence-to-sequence processing because the model is fed a sequence of tokens and produces another sequence, allowing outputting texts of a different structure than the input.

Translation Models The original Transformer model was designed specifically for machine translation. Initially, Transformers have gained considerable popularity specifically in this task due to the attention mechanism capability of passing information across the sequence in a non-sequential manner (Bahdanau et al., 2015b), as opposed to recurrent neural networks (Bengio et al., 2003). This aspect of Transformers models is especially practical in translating between languages with distinct word order, e.g. English to German.

Since the Vaswani et al. (2017) work, there have been multiple implementations of translation models based on the Transformer architecture, for instance, MarianMT (Junczys-Dowmunt et al., 2018). They have not varied significantly in terms of the architecture, but rather in the training corpora and coverage of the languages.

BART Bidirectional and Auto-Regressive Transformers (BART) (Lewis et al., 2020) is the family of language models trained for the reconstruction of corrupted texts. In this pre-training task, the spans of input text are replaced by special labels (it's a similar process to token masking described in the following Section 2.3.2). Subsequently, the model's decoder is tasked to generate the spans hidden by specific labels. The multilingual version: mBART (Liu et al., 2020a) was shown to be a strong base model for translation in multiple directions and supporting up to 50 languages.

T5s Text-to-Text Transfer Transformer (T5) (Raffel et al., 2020) were also pre-trained on the task of reconstructing corrupted texts. Because of the sequence-to-sequence nature of pre-training, T5 models have been also widely applied as a foundation for fine-tuning to various end-tasks: translation, question answering, parsing etc. There have been multiple versions of T5 models differing by size, training corpora, and input representation. To name a few: original: trained on English data only T5 (Raffel et al., 2020), multilingual: trained for 100 languages mT5 (Xue et al., 2021), T5 with byte-level input: byT5 (Xue et al., 2022). The author of the thesis is also responsible for the development of morphologically enriched T5 model myT5 (Limisiewicz et al., 2024), yet this work is beyond the scope of the dissertation.

2.3.2 Encoder-only Models

Encoder (or **autoencoding**) models are composed of just a single stack of layers. It is used to encode a sentence into a fixed-size representation. Encoder models are used for masked language model (MLM), where the whole sentences are used with some tokens masked out. The model is trained to predict the masked tokens based on the context from the left and right contexts. This type of model is especially useful for representation learning for sequences and as a basis for fine-tuning downstream classification tasks.

BERT Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) was the first Transformer model to introduce the masked language model pre-training task. In the training 15% of the tokens were selected for prediction, with 80% of them being replaced with a special token `[MASK]`, 10% were replaced with a random token, and 10% were left unchanged. The model was trained to predict the original token based on the context from the left and right sides. An additional pre-training task was the next sentence prediction, where BERT was trained to predict if the two presented sentences appear one after another in the training corpus.

BERT for a few years has been widely used in NLP research and application for textual embedding learning and fine-tuning to various downstream tasks (Rogers et al., 2020). Its popularity gradually fades with the introduction of larger models.

RoBERTa Robustly optimized BERT approach (RoBERTa) (Liu et al., 2019b) builds upon BERT, keeping the same architecture but changing the training procedure. Mainly, it removed the next sentence prediction loss and used larger training corpora and batch sizes.

XLM and Multilingual Pack Cross-lingual Language Model (XLM) (Conneau and Lample, 2019a) is a type of training and a family of models oriented on the implementation of language models supporting multiple languages. In addition to the regular masked language model task, XLM considers a translation language modeling task, where the model predicts the masked tokens in a language based on providing translated context in another language as a reference.

XLM design has inspired the design of the multilingual version of RoBERTa: XLM-RoBERTa (XLM-R) (Conneau et al., 2020a).

2.3.3 Decoder-only Models

The last type are **decoder** (or **autoregressive**) models. Similarly to encoders, they are composed of a single stack of layers. However, its purpose is different: to predict the next token in the sequence based only on the previous tokens. The main application of decoder models is generative language modeling. Such models can be also adapted to solving other tasks that can be achieved through instruction tuning. In this method, the model is given tasks in the form of textual prompts that guide it to produce the desired output.

GPT Generative Pre-trained Transformer (GPT) (Radford and Narasimhan, 2018) was the one of first models to introduce autoregressive training with the Transformer architecture. In the generative models, the left-hand context is masked in the Att. layer, and the model is trained just to predict the next tokens. GPT was released in multiple versions, with gradually increasing parameter sizes and training data: GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), GPT-4 (OpenAI, 2023). Unfortunately, the weights and training details of the latest models have not been released to the public.

Some GPT-based implementations have been open sourced. A distinct example is Eluther's GPT-Neox (Black et al., 2022), whose distinguishing architecture feature is the parallel connection of attention and feed-forward layers.

BLOOM BLOOM (Workshop et al., 2023) deserves special mention as a multilingual autoregressive model. The developers have closely followed the regular Transformer architecture, the most significant difference is the usage of ALiBi Positional Embeddings (Press et al., 2022). ALiBi allows training for unlimited sequence lengths and, as shown in the experiments, it smoothes the training loss and improves BLOOM's downstream performance. The author of the dissertation was involved in BLOOM development in the Evaluation and Interpretability group.

Chinchilla Chinchilla models (Hoffmann et al., 2022) focus on finding the optimal relationship between the model size and the training data size (under set computational constraints). The authors conduct a series of experiments of models of different parameter counts and training sizes, showing that many previous GPT models could benefit from increasing training data size. They introduce a new Chinchilla scaling rule, which suggests to use of around 20 training tokens per model parameter. It is important to note that the Chinchilla rule does not fit all scenarios, and the actual ratio can be affected by the quality of training data, language, and specific architectural choices.

LLaMAs Large Language Model Meta AI (LLaMA) (Touvron et al., 2023) gained popularity thanks to its high performance and (semi-)openly available weights. The model was trained on a large, mostly English corpus, which based on the Chinchilla rule, should enable better utilization of the models' parameters. The main architectural distinction of LLaMAs is the usage of rotary position embeddings (Su et al., 2024). In this approach, the position is encoded by multiplying token embeddings by a rotation matrix, instead of adding fixed positional embeddings (as described in Section 2.2.4).

3 What Do Language Models Learn?

In this chapter, we present an overview of the features that are central to our analysis of information encoded in language models. The content of this chapter overlaps in content with our previous survey of syntactic signals encoded in neural networks (Limisiewicz and Marecek, 2020). In the thesis, we add the analysis to cover the broader range of linguistic features: lexical, semantic, and social biases. We also extend the overview to cover the latest developments in the field.

In the first Section 3.1, we describe basic evaluation metrics that are used across multiple linguistic benchmarks. The subsequent sections are dedicated to specific linguistic features: morphology and syntax (Section 3.3), lexicon (Section 3.4), semantics (Section 3.5), coreference (Section 3.6), sentence- and document-level understanding (Section 3.7), and social biases (Section 3.8).

3.1 Introduction: Basic Metrics

We begin the overview by introducing or reminding the basic metrics that are widely used in the definition of most of NLP benchmarks.

3.1.1 Accuracy

Accuracy is a common measure used for the correctness of categorical prediction. In NLP predicted categories are usually tags (classes) $t \in \mathcal{T}$ of tokens (words) in a sequence (e.g. a sentence or a document) $\mathcal{S}$.

$$Acc = \frac{\#correct_t}{|\mathcal{S}|} \tag{3.1}$$

3.1.2 F1 Score: Classification

High accuracy might be relatively simple to achieve. For instance, in the case of a low number of unique tags and low variability, it is possible to always predict the most frequent tag. Therefore, accuracy is prone to saturation (i.e. always returning high values) preventing reliable comparison across models. More faithful evaluation is based on computing recall (R_t) and precision (P_t) of prediction of each tag t in the tag set $\mathcal{T}$. They are defined as follows:

$$\begin{aligned} P_t &= \frac{\#correct_t}{\#predicted_t}, \\ R_t &= \frac{\#correct_t}{\#gold_t} \end{aligned} \tag{3.2}$$

Then for each tag t, we compute the $F1$ score, which is a harmonic mean of precision and recall.

$$F1_t = \frac{2 \cdot P_t \cdot R_t}{P_t + R_t} \tag{3.3}$$

Tagging is a multiclass classification problem, and the $F1$ can be computed either as macro-average, i.e.

$$F1_{macro} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} F1_t \tag{3.4}$$

An alternative aggregation method is micro-average, where precision and recall are first computed globally for all classes together:

$$\begin{aligned} P_{micro} &= \frac{\sum_{t \in \mathcal{T}} \#correct_t}{\sum_{t \in \mathcal{T}} \#predicted_t}, \\ R_{micro} &= \frac{\sum_{t \in \mathcal{T}} \#correct_t}{\sum_{t \in \mathcal{T}} \#gold_t}, \\ F1_{micro} &= \frac{2 \cdot P_{micro} \cdot R_{micro}}{P_{micro} + R_{micro}} \end{aligned} \tag{3.5}$$

The main difference between the two is that the micro-averages are more sensitive to the scores for the more frequent classes, while the macro-average is an unweighted average treating all classes equally.

3.1.3 F1 Score: Retrieval

In some tasks, we do not have a sequence of tags covering all tokens, but a set of entities that should be retrieved from the sequence. We can define the precision, recall, and F1 for a set of retrieved entities E_r and the gold set of entities that are present in a sequence E_g:

$$\begin{aligned} P &= \frac{|E_r \cap E_g|}{|E_r|}, \\ R &= \frac{|E_r \cap E_g|}{|E_g|}, \\ F1 &= \frac{2 \cdot P \cdot R}{P + R} \end{aligned} \tag{3.6}$$

Similarly to the classification task, $F1$ can be averaged in two ways: micro and macro. Yet the definition in retrieval task is different: macro-average is computed as an average of per-sentence $F1$ scores, while micro-average involves global sets of entities: E_g and E_r obtained for the whole dataset.

3.2 Intrinsic Evaluation

The intrinsic metrics evaluate language models on the original pre-training task, i.e. token prediction. The simplest metric is computing the accuracy of predicting the token the same as in classification tasks. Other popular metrics are:

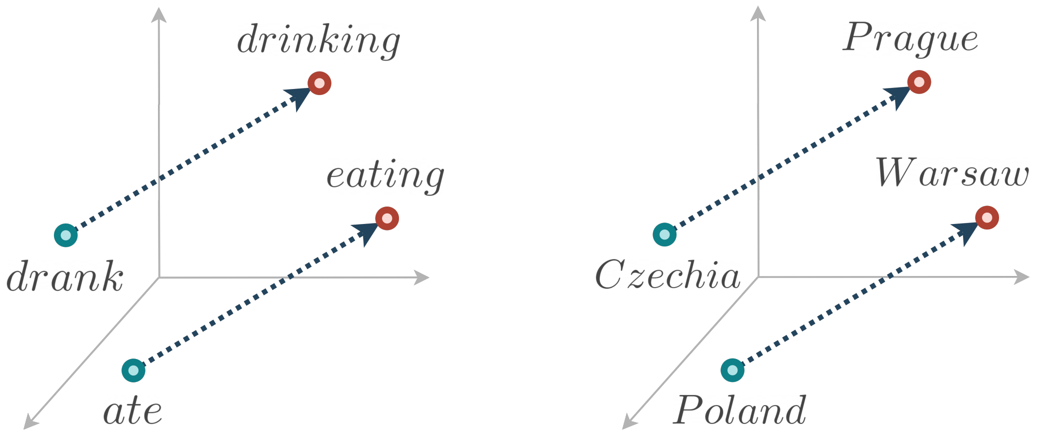
Mean reciprocal rank (MRR) is sensitive not only to the prediction with the highest probability but also considers the case when the correct token is among the top predictions.

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank}(x_i, \hat{P}(\cdot | X \setminus x_i))} \tag{3.7}$$

Perplexity is a measure of uncertainty in predicting the next token. Unlike MRR and accuracy, it is computed in the exponential space.

$$\text{Perplexity} = \exp\left(-\frac{1}{N} \sum_{i=1}^{N} \log \hat{P}(x_i | X \setminus x_i) \right) \tag{3.8}$$

where $\hat{P}(\cdot | X \setminus x_i)$ is the probability over the vocabulary of predicting token x_i by the model given its context: $X \setminus x_i$. In both metrics, the lower scores indicate better performance.



(a) Syntactic analogies

(b) Semantic analogies

Figure 3.1: Visualization of assumed representation of syntactic (present – past participle) and semantic (capital – country) analogies in word embeddings. (Inspiration for visualization: Ashutosh Singh).

Intrinsic metrics are often used due to the simplicity of computation and dropping the need for annotated data. However, such metrics do not provide a comprehensive picture of language model' capabilities and cannot replace the extrinsic evaluation on linguistic or language understanding tasks.

3.3 Morphology and Syntax

This section summarizes the methods of inquiring models for syntactic information.

3.3.1 Morphosyntactic Analogies

In the early works on word embeddings (Mikolov et al., 2013; Pennington et al., 2014), a strong focus was put on discovering spatial shifts in vector spaces. Such shifts correspond to syntactic or grammatical features of words. The idea behind this method is that the shift vectors between the pairs of words that differ in one feature are close to parallel. For example, we could identify the shift between the embeddings of English present and past participle, as shown in Figure 3.1a. Such analogies in vector space was observed for the pairs of words in different types of morphosyntactic relations: adjective – adverb; singular – plural; adjective – comparative – superlative; verb – present participle – past participle. Syntactic analogies of this type were signs that the representations learned by deep models capture linguistic features. The evaluation set of such relations was introduced by (Mikolov et al., 2013) as Google Analogy Test Set (GATS). Another Bigger Analogy Test Set (BATS) with 99,200 analogy questions was introduced by Drozd et al. (2016). [1]

An evaluation example consists of word pairs represented by the embeddings: $(v_1, v_2), (u_1, u_2)$. We compute the analogy shift vector as the difference between embeddings of the first pair $s = v_2 - v_1$. The result is positive if the nearest word embedding (in terms of cosine similarity) to the vector $u_1 + s$ is u_2.

$$WA = \frac{|\{(v_1, v_2, u_1, u_2) : u_2 \approx u_1 + v_2 - v_1\}|}{|\{(v_1, v_2, u_1, u_2)\}|} \tag{3.9}$$

3.3.2 Morphosyntactic Tagging

The morphosyntactic signal can be probed through the prediction of morphological features (e.g., adjective gradation) or coarse part of speech tags. POS tags are the categories of words sharing similar grammatical roles in the sentence, e.g.: noun, verb, adjective, adverb, pronoun etc. The evaluation of morphosyntactic tagging is usually done using token-level accuracy of sequence tagging, as shown in Equation 3.1. Alternatively, for an undiversified set of tags, the F1 score is a preferable metric to avoid benchmark saturation.

Morphosyntactic tags were part of one of the first annotated corpora, such as Brown Corpus (Francis, 1965) and commonly used Penn Treebank (Marcus et al., 1993). POS datasets were also introduced for multiple other languages than English, e.g. French (Abeillé et al., 2019), Czech (Hajič et al., 2020), or German (Brants et al., 2004). Significant effort has been put into unifying the annotation schemes for multiple languages (Zeman, 2008; Petrov et al., 2012).

3.3.3 Supertagging or Almost Parsing

Supertagging (Bangalore and Joshi, 1999) is an extension of morphosyntactic tagset to categories capturing multiple levels of syntactic structure. In addition to the POS tags, the model is tasked to predict the labels (supertags) for longer chunks of tokens: constituency phrases (e.g., noun phrases, verb phrases). The phrases are nested on multiple levels, which increases the complexity of the classification task. Supertagging is also called “almost parsing” because it can be seen as a slightly simplified version of the syntactic parsing task, which is described in the following section.

[1] The test set is called syntactic by authors; nevertheless, it mostly focuses on morphological features.

3.3.4 Syntactic Structure

The unsupervised or semi-supervised inference of multi-word linguistic structures is the more challenging task of testing the syntactic signal encoded in the models.

Dependency trees are evaluated using unlabeled attachment score (UAS) or its simpler-to-predict undirected variant (UUAS):

$$UAS = \frac{\#correctly\_attached\_words}{\#all\_words} \tag{3.10}$$

To increase the level of sophistication, we can consider also specific types of relations in the tree, i.e., labels. The equation for the labeled attachment score (LAS) is the same, but it requires predicting correct dependency labels for each edge.

For evaluating constituency trees, we compute the retrieval $F1$ score, comparing predicted phrases (or constituents) with the annotated ones. Denoted as E_r and E_g respectively, in the formulation of Equation 3.3.

Currently, the most popular datasets for dependency structures are annotated under the universal dependencies (UD) framework (de Marneffe et al., 2021). The UD project is based on community contributions and contains over 200 treebanks in more than a hundred languages. Constituency parses are sometimes available in treebanks mentioned in POS tagging Section 3.3.2, e.g. Penn Treebank (Marcus et al., 1993).

3.4 Lexicon

3.4.1 Lexical Analogies

These types of associations are similar to morphosyntactic analogies but focus on the meaning of the words rather than their grammatical form. The aforementioned GATS (Mikolov et al., 2013) contains the following types of semantic relations: capital – country; currency – country; city in the US – state; male – female persone (e.g. boy – girl), examples of such relations are shown in Figure 3.1b. The same as in morphosyntactic analogies, the correctness of association is measured by the Word Analogy score (WA) defined in Equation 3.9.

3.4.2 Hypernymy Structure

A lexical analogy of syntactic structures hypernymy trees, which are annotated on lexicon (a set of lexemes, i.e. unique words). An example of such a lexicon is WordNet (Miller, 1992). WordNet contains annotations of semantic relations between lexemes: hypernymy, meronymy, and synonymy. The definition is similar to syntactic parsing, but the edges represent one of the relations between lexemes.

In addition to data in WordNet (Miller, 1992), the task is also defined for multilingual corpora collected into Open Multilingual WordNets (Bond and Foster, 2013).

3.4.3 Derivational Tree

Another type of structure involving lexemes is the derivational tree (Vidra et al., 2019). Similarly to WorDnet, the dataset is an annotated lexicon organized into a tree structure. The root of each of the trees is a morphological lemma connected to its derivational forms. The number of edges between lexemes corresponds to the number of derivational steps. The evaluation of parses is defined in the same way as in WordNet or dependency parsing.

3.5 Semantics

3.5.1 Semantic Tagging

Semantic features can be also evaluated with the classification task. A popular example is (Palmer et al., 2005) involving tags, such as: *agent*, *location*, *purpose*, *direction* There are various sets of tags belonging to this category differing in the granularity of categories (Teichert et al., 2017; Rudinger et al., 2018b). Bjerva et al. (2016) proposes semantic sequence tagging, which is a semantic extension of part of speech (POS) tagging, for instance, distinguishing determiners into categories such as *proximal* ("this") and *distal* ("that").

3.5.2 Named Entity Recognition

A particularly useful set of semantic roles are named entities, such as: *person*, *organization*, *geographical entity*. This type of benchmark is called named entity recognition (NER) and is used in various NLP applications. The popular dataset for named entity recognition are MUC-7 (Chinchor, 1998), CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003),and OntoNotes (Weischedel et al., 2017). There are also recent efforts to unify annotation schemes across distinct languages into one standard: Universal NER (Mayhew et al., 2023).

3.5.3 Word Seneses

Distinction of word senses is another task involving semantic reasoning. It precludes distinguishing between multiple senses of a single polysemous word (such as “train” in “I came on a train.” versus “I train young children.”). Performing well in this benchmark requires sensitivity to context in learned representation, which is an advantage of neural networks over static word embeddings. An example of the dataset is SemCor 3.0 (Miller et al., 1993), a corpus with annotations of word senses.

3.6 Coreference

In the coreference resolution task (e.g. OntoNotes Weischedel et al. (2017)), we consider the phenomenon of several entities (typically pronouns or noun phrases) referring to the same real-world entity. For example, “John told Mary he liked her, and she let him kiss her on her cheek.”, where “John”, “he” and “him” refer to one entity (John) and “Mary”, “her” and “she” to another entity (Mary).

Winograd Scheme Winograd Schema Challenge (WSC) (Levesque, 2011) is a type of coreference resolution benchmark that requires from the model a higher level of language understanding and common sense reasoning The Winograd scheme focuses on cases of coreference that are morphologically and syntactically ambiguous, for example: “*Characters* entertain *audiences* because **they** want people to be happy.” where determinaing based on semantic cues whether “they” could refer to “characters” or “audiences” is necessary to correctly solve the example.

The coreference resolution is evaluated by F1 for retrieval of coreference links between words or phrases, defined in Equation 3.6.

3.7 Sentence and Document Level Understanding

In this section, we describe higher-level tasks that require comprehension of both semantic, and syntactic cues, and sometimes also common world knowledge facts. Such benchmarks are currently the most widely used due to their complexity connected with a lower risk of saturation.

3.7.1 Natural Language Inference

Natural language inference (NLI) is a sentence-level benchmark where the model is tasked with determining the logical relationship between two sentences: hypothesis and premise. The task is realized as a three-way classification problem, with the labels: *entailment*, *neutral*, and *contradiction*. If the premise can be logically inferred from the hypothesis, the example should be labeled as *entailment*. In case the premise contradicts the information provided in the hypothesis, the label should be *contradiction*. For the remaining cases, the hypothesis and premise are logically unrelated and labeled as *neutral*.

The models are evaluated using the accuracy (see Equation 3.1). Noticeable datasets for NLI are Stanford Natural Language Inference (SNLI) (Bowman et al., 2015) and Multi-Genre NLI (MNLI) (Williams et al., 2018). Conneau et al. (2018b) introduced a cross-lingual natural language inference (XNLI) version covering 15 languages.

3.7.2 Question Answering: Open Book

In the task of question answering (QA), the NLP system is provided a reference text, e.g. Wikipedia paragraph, and a question about the information contained in the text. In a popular dataset OpenBookQA the task is evaluated as a retrieval task, where the system should identify the specific span of input text containing the answer or indicate that the answer is not given.

The evaluation is done with retrieval-style F1 metric (see Equation 3.6). In other benchmarks, examples include multiple-choice options and standard accuracy of correct answers is reported, an example of such a dataset is:

Natural Questions (NQ) (Kwiatkowski et al., 2019) that includes 307,373 anonymized questions from the Google search engine and adequate Wikipedia snippets.

OpenBookQA (OBQA) (Mihaylov et al., 2018) that contains 5,957 multiple-choice questions aimed at combining science facts with common knowledge.

3.7.3 Question Answering: Closed Book

In a more challenging *closed book* setting, the model is queried for answers without providing the reference text. Due to a lack of reference text, the evaluation is done via the accuracy of choosing the correct answer from the multiple-choice question. Examples of *closed book* benchmarks involve:

AI2 Reasoning Challenge (ARC) (Clark et al., 2018) that contains natural science questions authored for use on standardized tests. It is partitioned into a Challenge Set (1,172 test questions) and an Easy Set (2,376 test questions).

Measuring Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2021) contains 14,042 questions on 57 topics, including math, law, or social sciences. The common practice is providing five correctly solved questions as a reference in each test example

3.8 Bias and Fairness

So far, we described the datasets measuring the models' competency in understanding and processing language. Another aim of evaluation is to identify the societal cues that the models use to make predictions. It has been shown (Bolukbasi et al., 2016) that the models often rely on stereotypes and harmful biases learned from the training data.

The bias evaluation checks to what extent the model's prediction is influenced by stereotypes related to, e.g., gender, religion, ethnicity. Another aspect of bias is checking the model's uneven performance for test cases related to specific social groups. Such test cases are collected into *challange sets* (Nadeem et al., 2021; Nangia et al., 2020a; Zhao et al., 2018; Rudinger et al., 2018a; Stanovsky et al., 2019), for which the models often perform worse than for the general population.

In this Section, we first present the important distinction between factual and stereotypical gender signals (Subsection 3.8.1) as an example of a problematic issue connected to bias evaluation, then we overview the popular metrics for measuring different manifestations of bias: stereotypical associations (Subsection 3.8.2), unequal performance, and bias in generation (Subsection 3.8.4).

3.8.1 Factual and Stereotypical Gender Signal

We consider two types of gender information encoded in text:

- **Factual gender** or **definitional gender** is the grammatical (pronouns "he", "she", "her", etc.) or semantic ("boy", "girl", "king", "queen", etc.) feature of specific word. It can also be indicated by a coreference link. We will call words with factual gender as *gendered* in contrast to *gender-neutral* words.

- **Sterotypical gender** or **Gender bias** is the connection between a word and a specific gender with which it is usually associated, regardless of the factual premise. The association is solely based makon stereotypical cues.[2] We will refer to words with gender bias as *biased* in contrast to *non-biased*.

Please note that those definitions do not preclude the existence of biased and at the same time gender-neutral words. In that case, we consider bias stereotypical and aim to mitigate it in our method. On the other hand, we want to preserve the factual gender signal in gendered words.

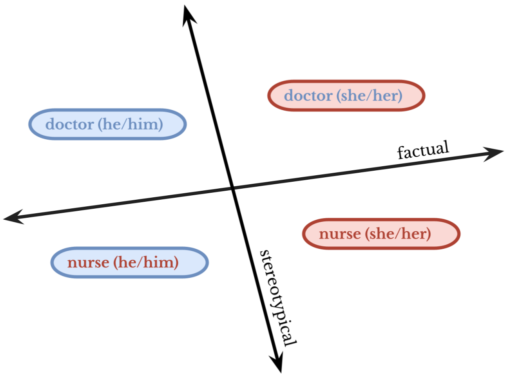


Figure 3.2: A schema presents the distinction between gender bias (stereotypical gender) of nouns and factual (i.e., grammatical) gender in pronouns.

3.8.2 Association Bias

The first set of presented benchmarks tests the stereotyping and negative associations about specific social groups that are encoded in LMs representations. We present metrics for assessing bias based on both single words and whole sentences.

Word Embedding Association Test (WEAT) In Section 3.3.1, we described the Google Analogy Test Set (Mikolov et al., 2013) that tests the association between word embeddings to indicate the language and world knowledge encoded in the vector representations. Further works (Caliskan et al., 2017; Bolukbasi et al., 2016) observed that such association may not be based on facts but rather on stereotypes. Caliskan et al. (2017) introduces the Word Embedding Association Test (WEAT) that measures the associations between embeddings of words connected to social groups and independent attributes. For instance, they compare the association of typically European-American and African-American names with pleasant and unpleasant words, and male and female names with career and family-related words.

Similarly to the analogy test, they use the cosine similarity between the embeddings as a similarity measure $s(u,v) = \cos(u,v)$. For two sets of target words U, V (e.g. male and female names), and two sets of attribute words A, B (e.g. career and family-related words), the WEAT score is defined as follows:

[2] For instance, the words “nurse”, “housekeeper” are associated with women, and words “doctor”, “mechanic” with men. None of those words has a grammatical gender marking in English.

$$\mathrm{WEAT}(U,V,A,B) = \sum_{u\in U}\left(\sum_{a\in A}\frac{s(u,a)}{|A|} - \sum_{b\in B}\frac{s(u,b)}{|B|}\right) - \sum_{v\in V}\left(\sum_{a\in A}\frac{s(v,a)}{|A|} - \sum_{b\in B}\frac{s(v,b)}{|B|}\right) \tag{3.11}$$

WEAT scores closer to zero indicate lower bias.

StereoSet (StereoS) Nadeem et al. (2021) propose a dataset for measuring stereotypical biases of sentences on four domains: gender, profession, race, and religion. The dataset contains 16,995 context sentences for which the models should predict the most likely completion from the set of three options: stereotypical, anti-stereotypical, and meaningless. For instance, a test example for the gender domain includes the following context: *"Girls tend to be more ____ than boys."* and three possible completions: *"soft"* (sterotypical), *"determined"* (anti-stereotypical), and *"fish"* (unrelated).

The evaluation metrics are computed similarly to classification tasks. Instead of accuracy authors propose three aggregated scores:

1. Language Modeling Score (LMS) – the percentage of sentences where the model prefers the meaningful (stereotypical or anti-stereotypical) over the meaningless completion;

2. Sterotype Score (SS) – the percentage of sentences where the model prefers the stereotypical over the anti-stereotypical completion;

3. Idealized Context Association Test (ICAT) score that combines the previous two: $icat = lms \cdot \min(ss, 100 - ss)/50$.

Please note that typically lower ss scores refer to less biased models since they are closer to 50.

CrowS pairs In s similar attempt to StereoS, Nangia et al. (2020a) proposed another association dataset for language models, for more stereotype types.[3] Unlike StereoS, the evaluation is not based on multiple-choice completion, but on estimating the likelihood under the model of the sentences completed with stereotypical and anti-stereotypical candidates. The difference in likelihood indicates the bias of the model.

[3] They consider the following aspects: race, gender, sexual orientation, religion, age, nationality, disability, physical appearance, occupation

3.8.3 Performance Bias

Another manifestation of bias is the difference in system performance that affects specific social groups unequally. This type of bias often stems from the uneven frequency of training examples that consider particular groups. Past works introduced a *challenge sets* that focuses on quality evaluation for data considering underrepresented groups to assess the prevalence of performance bias.

WinoBias (WB) Zhao et al. (2018) proposes using a dataset containing a Winograd Schema Challenge (WSC) examples (Levesque, 2011). Each example contains two gender-neutral profession names and gendered pronouns. The task is to identify the coreference link between the pronouns and the correct professional. The dataset consists of two parts: pro-stereotypical, where coreference links to a profession name with stereotypical gender matching the gender of the pronoun; in anti-stereotypical examples, the profession's stereotypically assumed gender is different from the gender of the pronouns. The stereotype of the profession is based on the gender statistics of the profession provided by the US Department of Labor `https://www.bls.gov/cps/cpsaat11.htm`.

An example from the anti-stereotypical subset: *"The nurse notified the patient that **his** shift would be ending in an hour."* In this example, the coreferential link relies on semantics, while in other instances, coreference can be resolved solely through syntax.

The evaluation is computed by the F1 score of correctly predicting the coreference link between the pronoun and the profession name. The scores are computed separately for pro-stereotypical vs. anti-stereotypical and male vs. female examples and compared against each other. The difference in scores is an indicator of bias presence in the model.

WinoGender (WG) (Rudinger et al., 2018a) proposed a similar dataset based on the WSC for coreference resolution with gendered pronouns. The main difference is that they also considered examples with neutral gender, denoted by the singular *they/them* pronouns.

Gedner Bias in Machine Translation: WinoMT Based on the WinoBias and WinoGender evaluation examples Stanovsky et al. (2019) proposed a dataset WinoMT (WM) for evaluating bias in translation from English into eight languages with morphological marking of gender (e.g., German, Spanish, Russian, Hebrew).

In WinoMT, the correctness of the translation is computed by the F1 score of correctly generating gender inflection of profession words in the target language. The authors propose two scores measuring the system's preference for stereotypical vs. anti-stereotypical and male vs. female translations.

Specifically, ΔG measures the difference in gender translation correctness (F1) between masculine and feminine entities:

$$\Delta G = \mathrm{F1}_{m.trans.} - \mathrm{F1}_{f.trans.} \tag{3.12}$$

Similarly, ΔS measures the difference in F1 between pro-stereotypical and anti-stereotypical instances of gender role assignments.

$$\Delta S = \mathrm{F1}_{pro.trans.} - \mathrm{F1}_{anti.trans.} \tag{3.13}$$

The author of this dissertation has also contributed to extending the MT Gender dataset to Polish and Czech (Kocmi et al., 2020).

3.8.4 Bias in Natural Language Generation

To better understand gender bias in language generation, we construct our dataset of prompts and an interpretable diagnostic measure.

We use the set of professions chosen and annotated by Bolukbasi et al. (2016).[4] Each profession was assigned two scores: *factual* score x_f (originally called *definitionality*) and *stereotypical* score x_s. By convention, scores range from -1 for female-associated words to 1 for male ones.[5] We fill the proposed profession words in the prompts of a structure:

"The **lifeguard** laughed because ..."

where the **lifeguard** is by definition a gender-neutral word ($x_f = 0$) and associated with the male gender by a stereotypical cue ($x_s = 0.6$). We measure the probabilities for gendered prediction for a given prompt $P_M(o|X)$. For that purpose, we use pronouns $o_+ =$ "he" and $o_- =$ "she", as they are probable continuations for given prompts. Subsequently for each prompt, we can compute *emprical* score $y_e = P_M(o_+|X) - P_M(o_-|X)$. To estimate the relationship between the observed score and annotated ones x_s and x_f, we construct a linear model:

[4] The data is available at: `https://github.com/tolga-b/debiaswe/blob/master/data/professions.json`

[5] We use positive values for male gender following the original paper. This is only an arbitrary choice, and switching polarities wouldn't affect this analysis. Importantly, we do not intend to ascribe negative valuations to any of the genders.

$$y_e = a_s \cdot x_s + a_f \cdot x_f + b_0 \tag{3.14}$$

The linear fit coefficients have the following interpretations: a_s is an impact of stereotypical signal on the model's predictions; a_f is an impact of the factual (semantic) gender of the word. Noticeably, y_e, x_s, and x_f take the values in the same range. The slope coefficients (a_s and a_f) tell how the change in annotated scores across professions impacts the difference in predicted probabilities of male and female pronouns. The intercept b_0 measures how much more probable overall are the male than the female pronouns, i.e., when we marginalize the subject.

4 Latent Representation

In this Chapter, we describe works focused on identifying the information learned by latent representations. Our motivation is to find out what kind of information is encoded in latent embeddings h and what connects or differentiates the encoding of distinct kinds of linguistic information throughout languages. In particular, we summarize previous literature on probing which is a lightweight method for such evaluations in Section 4.1. Subsequently, we describe the strengths of our novel *orthogonal probe* in Section 4.2, such as enabling the separation of the representations of specific linguistic features, and higher robustness to memorization of randomly generated structures. Further in Section 4.4, we extend the *orthogonal probe* to a multilingual setting, testing the hypothesis of uniformity and orthogonality of representation across diverse languages. Finally, Section 4.5 outlines the application of *orthogonal probe* to filter unwanted information from the latent space.

4.1 Probing: From Tags to Structure

4.1.1 Probing For Tags

Probes are classification layers (see Section 2.2.5) trained on top of the model with frozen parameters (i.e., not updated in training). A standard probe takes a vector representation as an input and predicts the downstream task-specific labels, e.g. NER, or POS tags. Due to weight freezing, the performance of the probe is an indicator where the task-related classification is encoded in the model (Belinkov, 2022a). Sequence tagging is a common task used in probing due to its simplicity and context dependence.

The probes are often applied to the latent representation of the model, i.e. the output of the intermediate instead of the last layer. Probing the intermediate layers allows for identifying the parts of the models that encode the information in question. Such experiments revealed that specific layers tend to "specialize" in encoding different types of linguistic signals (Tenney et al., 2019).

Researchers have previously used various types of tagging to determine whether the model encodes specific information. For instance, POS and super-tags were probed for by (Liu et al., 2019a; Blevins et al., 2018) to evaluate the morphosyntactic signal, SRL or NER to evaluate the semantic information (Conneau et al., 2018a), and coreference was probed for by (Tenney et al., 2019).

4.1.2 Structural Probes

Probing for structure, typically syntactic trees is more sophisticated than tagging. Structual task complexity makes it harder to memorize the labels in the added classification layer, and thus discern the models that actually learn the relevant signal in pre-training (Hewitt and Liang, 2019). Moreover, structural tasks check the model's ability to encode information highly dependent on the context.

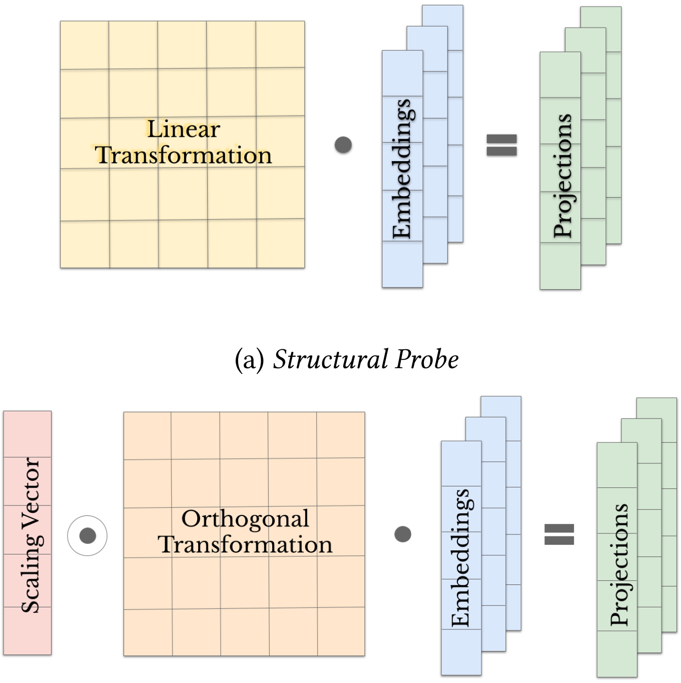
The work Hewitt and Manning (2019) introduced *structural probe* to examine the encoding of syntactic dependency structure in latent embeddings of LM. They introduced a linear transformation on top of the contextual word representations from a pre-trained neural model (e.g. BERT Devlin et al. (2019), ELMo Peters et al. (2018)). The transformation was gradient optimized to approximate the distance in a syntactic tree[1] by squared L2 norm of the differences between transformed word vectors.

$$d_B(h_i, h_j)^2 = (B(h_i - h_j))^T (B(h_i - h_j)), \tag{4.1}$$

where B is the *linear transformation* matrix and h_i, h_j are the vector representations of words at positions i and j. The probe is optimized to approximate the tree distance (d_T) by gradient descent objective:

$$\min_B \frac{1}{s^2} \sum_{i,j} \left| d_T(w_i, w_j) - d_B(h_i, h_j)^2 \right|, \tag{4.2}$$

[1] syntactic tree distance is defined as the number of edges on a path between two considered words.



(a) *Structural Probe*

(b) *Orthogonal Probes* (ours)

Figure 4.1: Comparison of the *Structural Probe* of Hewitt and Manning (2019) and the *Orthogonal Probes* proposed by us.

where s is the length of a sentence. Moreover, the same work introduced depth probes, where vectors were linearly transformed so that the squared L2 length of the mapping approximates the token's depth in a dependency tree, which is equivalent to the distance from the root of the tree:

$$||h_i||_B^2 = (Bh_i)^T(Bh_i)d \tag{4.3}$$

Gradient descent objective is analogical:

$$\min_B \frac{1}{s}\sum_i \left| \|w_i\|_T - \|h_i\|_B^2 \right| \tag{4.4}$$

4.2 Introducing Orthogonal Probing

In this section, we introduce our contribution to the field of probing: *orthogonal probe*, originally introduced in Limisiewicz and Mareček (2021b). The motivation of this method is to identify the specific components of the latent space that encode specific signals. It also enables us to map the interaction between different kinds of encoded signals (e.g. syntactic and lexical) and filter unwanted ones (e.g. gender bias).

4.2.1 Methodology

We introduce orthogonality to the probes. For that purpose, we perform singular value decomposition (SVD) of the matrix B

$$B = U \cdot D \cdot V^T, \tag{4.5}$$

where the U and V are orthogonal matrices, and D is a diagonal matrix. Notably, when we substitute B with $U \cdot D \cdot V^T$ in Equation 4.1, the matrix U cancels out. It can be easily shown by rearranging the variables in the equation:[2]

$$\begin{aligned} d_B(h_i, h_j)^2 &= (UDV^T(h_i - h_j))^T(UDV^T(h_i - h_j)) \\ &= (h_i - h_j)^T VD^TU^TUDV^T(h_i - h_j) \\ &= (h_i - h_j)^T VD^TDV^T(h_i - h_j) \\ &= (DV^T(h_i - h_j))^T(DV^T(h_i - h_j)) \end{aligned} \tag{4.6}$$

We can replace the diagonal matrix D with a vector $\bar{d}$ and use an element-wise product (we will call $\bar{d}$ the *scaling vector*). Finally, we get the following equation for *distance orthogonal probe*:

$$d_{\bar{d}V^T}(h_i, h_j)^2 = (\bar{d} \odot V^T(h_i - h_j))^T(\bar{d} \odot V^T(h_i - h_j)) \tag{4.7}$$

The same reasoning can be applied to Equation 4.3 to obtain *depth orthogonal probe*:

$$||h_i||^2_{\bar{d}V^T} = (\bar{d} \odot V^T h_i)^T(\bar{d} \odot V^T h_i) \tag{4.8}$$

Thus, we showed that *orthogonal probe* is mathematically equivalent to *structural probe*.

[2] A complete derivation can be found in the appendix.

4.2.2 Multitask Training

Orthogonal probe can be easily adapted to multitask probing for a set of objectives $\mathcal{O}$. We use one shared orthogonal transformation and different *scaling vector*s for each task. In one batch, we compute a loss for a specific objective. For each batch (with objective $o \in \mathcal{O}$), a forward pass consists of multiplication by a shared orthogonal matrix V^T and product-wise multiplication by a vector $\bar{d}_o$ designated for a specific task o. All the batches are shuffled together in a training epoch.

4.2.3 Orthogonality Regularization

We use *Double Soft Orthogonality Regularization* (DSO) proposed by Bansal et al. (2018) to coerce orthogonality of the matrix V during training:

$$\lambda_O DSO(V) = \lambda_O(||V^T V - \mathbb{I}||_F^2 + ||VV^T - \mathbb{I}||_F^2) \tag{4.9}$$

where $|| \cdot ||_F$ stands for the Frobenius norm of a matrix.

4.2.4 Sparsity Regularization

In further experiments, we investigate the effects of sparsity in *scaling vector*s. For that purpose, we compute the L1 norm and add it to the training loss. A similar regularization term, *Lasso*, was proved effective for coefficient selection in linear models (Tibshirani, 1996).[3]

$$\lambda_S \|\bar{d}\|_1 \tag{4.10}$$

4.2.5 Training Objective

Altogether, the loss equation of *distance orthogonal probe* for objective $o \in \mathcal{O}$ is the following:

$$L_{o,dist.} = \frac{1}{s^2} \sum_{i,j} \Big| d_T(w_i, w_j) - d_{\bar{d}_o V^T}(h_i, h_j)^2 \Big| + \lambda_O DSO(V) + \lambda_S \|\bar{d}_o\|_1 \tag{4.11}$$

And in *depth orthogonal probe*:

$$L_{o,depth} = \frac{1}{s} \sum_{i} \Big| \|w_i\|_T - \|h_i\|^2_{\bar{d}_o V^T} \Big| + \lambda_O DSO(V) + \lambda_S \|\bar{d}_o\|_1 \tag{4.12}$$

The loss is normalized by the number of predictions in a sentence and averaged across a batch.

[3] Sparsity regularization was used only in one experiment, see Table 4.2

4.2.6 Orthogonal Filters

Orthogonal probe can be optimized for multiple objectives and identify the specific dimension in the latent space to encode the signals. This property inspired us to apply *orthogonal probe* to filtering information specific to one objective from the latent space while preserving information related to another. In the algorithm, we aim to filter out the latent vector's dimensions that encode one kind of information (denoted s) while keeping the dimensions encoding another signal (denoted f). [4].

Thanks to *orthogonal probe*, we can diminish the information by masking the dimensions with a corresponding *scaling vector* coefficient larger than small ϵ. The orthogonal s-filter is defined as:

$$F_{-s} = \overrightarrow{\mathbb{1}}[\epsilon > abs\ (\bar{d}_s))], \tag{4.13}$$

where $abs(\cdot)$ is element-wise absolute value and $\overrightarrow{\mathbb{1}}$ is element-wise indicator. We apply this vector to the representations of hidden layers:

$$\hat{h} = V \cdot (F_{-s} \odot (V^T \cdot h)) \tag{4.14}$$

To preserve signal f we train a separate probe for it, i.e., we share the same orthogonal matrix V and use a different *scaling vector* $\bar{d}_f$. The latent dimension is kept when its importance (measured by the absolute value of the *scaling vector* coefficient) is higher in probing for f than in probing for bias. We define f preserving Orthogonal s-filter as:

$$F_{-s,+f} = F_{-f} + \overrightarrow{\mathbb{1}}[\epsilon \leq abs(\bar{d}_s) < abs(\bar{d}_f)] \tag{4.15}$$

4.3 Multitask Orthogonal Probes

We train probes on top of each of the 24 layers of the English BERT large-cased model (Devlin et al., 2019) hosted at HuggingFace (Wolf et al., 2020). We optimize for the approximation of depth and distance in four types of structures: syntactic dependency, lexical hypernymy, absolute position in a sentence, and randomly generated trees. In the following paragraphs, we describe them in more detail:

[4] The letters s and f are used to denote the *sterotypical* and *factual* signal, respectively. We use this notation because, in our experiments, we will focus on filtering gender bias

Dependency Syntax We probe for syntactic structure in universal dependencies parse trees (de Marneffe et al., 2021). Specifically, we use the trees from the English Web Treebank (Silveira et al., 2014a). We focus on distances between words in dependency trees and their depths, i.e., distances from the syntactic root.

Lexical Hypernymy We introduce probing for lexical information. We optimize probes to approximate the distance between pairs of words in the hypernymy tree and the depth for each word. For that purpose, we use the tree from WordNet (Miller, 1992). We consider lexical distances between pairs of nouns and pairs of verbs in sentences and lexical depth for each noun and verb. We use gold POS information and disambiguate the meaning of a lexeme for sets sharing the same orthographic form (sysets).

Position in a Sentence Probing for the sentence index of a word and positional difference between pairs of words.

Random Structures We probe for randomly generated trees. This control task allows us to determine the extent to which our probes memorize the structures and thus overfit to the training data.

4.3.1 Experiments

We assess Spearman's rank correlation between gold and predicted values. We report the average correlations for the sentences with lengths from 5 to 50 in the same way as Hewitt and Manning (2019).

Our *orthogonal probe*s are trained jointly for multiple objectives. We evaluate the effect of multitasking by checking different configurations: **A)** separate probing for each objective; **B)** joint probing for distance and depth in the same structure type; **C)** joint probing for distance in all structures; **D)** joint probing for depths in all structures; **E)** probing for all objectives together. We compare the results with two baselines: **I)** optimizing only *scaling vector*[5]; **II)** *structural probe*s.

Dimensionality of Scaling Vector We hypothesize that the orthogonality regularization allows us to find embedding subspace capable of representing a particular linguistic structure. We examine the performance of lower-rank projections and ask whether further restrictions of dimensionality affect the results. Subsequently, we analyze interactions between subspaces related to a particular objective in a joint probing setting.

[5] Optimizing only *scaling vector* equates to fixing U and V from Equation 4.5 as identity matrices.

4.3.2 Results

We compare Spearman's correlations of predicted values with gold tree depths and distances in Table 4.1. The correlations obtained from *orthogonal probe*s are high for linguistic structures: from 0.803 for lexical distance to 0.882 for lexical depth. Predicted positional depths and distances nearly match gold values.

***Orthogonal probe*s are selective** Correlation on training data for random structures is very weak, hinting that the probes do not memorize structures during training but extract them from the model's representations. The correlation for distances is higher than for depth. We hypothesize it is because the probes learn some basic tree properties.[6]

Finding 1

Using orthogonal regularization in probing mitigates the risk of memorizing the examples revealed in supervised training.

***Orthogonal probe*s capture linguistic signals** The results obtained by *orthogonal probe*s are close to those of *structural probe*s. For dependency distance, the difference is not statistically significant. Notably, correlations, computed on the training set, for randomly generated trees decreased. It suggests that *orthogonal probe*s are less vulnerable to memorization. In multitask probing, correlation evenly decreases across all tasks, while selectivity (the difference between average correlation for dependency, lexical, and positional objectives and random objectives) increases from 0.673 to 0.726. Optimizing only a *scaling vector* gives distinctly lower correlations. These results emphasize the necessity of changing the coordinate system to identify the dimensions that correspond to linguistic information more than underlying neurons.

In Figure 4.2 (upper), we observe that the performance varies throughout the layers, confirming previous observations by Hewitt and Manning (2019) and Tenney et al. (2019). The mid-upper layers tend to be more syntactic, and the mid-lower ones are more lexical. Predicting word position is more accurate in the lower layers, dropping significantly toward the last layers. This is because, in BERT, positional embeddings are added before the first layer. Random structure probes maintain steady results across all the layers.

[6] For instance, when the distances between nodes X and Y, and Y and Z are both 1, then the distance between X and Z needs to be 2

	I	II	A	B	C / D	E
				multitask orthogonal probing		
	Scaling Vector only	Structural Probe	Orthogonal Structural Probe	distance + depth	all distances or all depths	all tasks
DEP Depth	.459 ±.001	.856 ±.001	**.858** ±.001	.855 ±.001	.850 ±.002	.852 ±.001
Layer	17	18	17	16	16	16
DEP Dist.	.513 ±.001	**.843** ±.001	**.842** ±.001	.838 ±.001	.833 ±.001	.832 ±.002
Layer	18	17	17	17	17	16
LEX Depth	.572 ±.001	**.892** ±.002	.882 ±.002	.869 ±.005	.885 ±.004	.873 ±.005
Layer	13	8	8	8	6	9
LEX Dist.	.560 ±.001	**.816** ±.008	.803 ±.005	.789 ±.004	.792 ±.010	.792 ±.005
Layer	13	6	6	7	6	6
POS Depth	.232 ±.013	**.989** ±.001	.983 ±.001	.986 ±.001	.976 ±.004	.982 ±0.001
Layer	5	1	6	1	2	3
POS Dist.	.441 ±0.001	**.980** ±.001	.979 ±.001	.977 ±.001	.978 ±.001	.976 ±0.001
Layer	1	4	4	4	5	4
RAND Depth	.008 ±.002	.206 ±.010	.136 ±.007	.129 ±.010	.163 ±.023	**.107** ±.019
Layer	6	17	18	18	18	19
RAND Dist.	.149 ±.001	.242 ±.005	.220 ±.006	**.206** ±.004	**.209** ±.005	**.208** ±.007
Layer	17	19	18	17	19	15
AVG. DEP, LEX, POS	.463	.896	.891	.886	.886	.883
ABOVE - AVG. RAND	.385	.673	.713	.718	.699	.726

Table 4.1: The highest Spearman's correlations (across layers) between predicted values and gold annotations on a held out test set (for random structures computed on a train set to test memorization). Each column represents another variant of training. The standard deviation was calculated for six runs. Each row's optimal result is underlined (except baseline **I**); **results within 95% confidence interval** based on Student's t-test Student (1908) are marked in bold.

Finding 2

Probing reveals that in BERT the most accurate representation of lexical structures is found in the mid-lower layers, while syntactic structures in the mid-upper layers.

Signal is encoded in a low-rank subspace of the embedding space We observe that orthogonality constraint is quite effective in restricting the probe's rank. In most of our experiments, the majority of *scaling vector* parameters converged to zero. It allows the selection of sparse subspaces encoding particular linguistic features. We want to answer whether such subspace has enough capacity for each probing task. For that purpose, we zero out the dimensions with corresponding *scaling vector*

	Subspace		Share of Dropped			Sparsity Regularization					
			Dimensions			$\lambda_S = 0.005$		$\lambda_S = 0.05$		$\lambda_S = 0.1$	
	Dims	Corr	25%	33%	50%	Dims	Corr	Dims	Corr	Dims	Corr
DEP Depth	137	.858	.783	.758	.700	26	.856	2	.832	1	.822
DEP Dist.	189	.842	.800	.781	.741	76	.835	21	.784	14	.746
LEX Depth	19	.884	.841	.822	.784	19	.875	11	.852	10	.836
LEX Dist.	263	.805	.768	.755	.722	92	.792	60	.756	52	.737
POS Depth	20	.983	.760	.686	.526	11	.982	6	.981	3	.981
POS Dist.	98	.979	.890	.859	.627	38	.978	14	.975	11	.970
RAND Depth	259	.128	.108	.101	.091	6	.037	1	.011	1	.010
RAND Dist.	399	.222	.215	.213	.208	116	.208	20	.163	13	.155

Table 4.2: The highest Spearman's correlations (across layers) between predicted values and gold annotations on a held-out test set (for random structures computed on a train set). In columns 2-3, results, when only dimensions with corresponding *scaling vector* values closer to zero than $\epsilon = 10^{-4}$ are kept. In columns 4-6, a random portion of the selected dimensions is masked. In columns 7-12, sparsity regularization with different λ_S is applied. In each scenario, we probed for a single objective.

		DEP		LEX		POS		RAND	
		Depth	Dist.	Depth	Dist.	Depth	Dist.	Depth	Dist.
DEP	Depth	62	48	0	0	10	19	23	21
	Dist.		126	0	0	9	23	25	30
LEX	Depth			20	18	0	4	1	5
	Dist.				131	0	7	5	19
POS	Depth					14	10	13	10
	Dist.						70	33	50
RAND	Depth							131	95
	Dist.								262

Table 4.3: The number of shared dimensions selected by *scaling vector*s after the joint training of probe on top of the 16th layer.

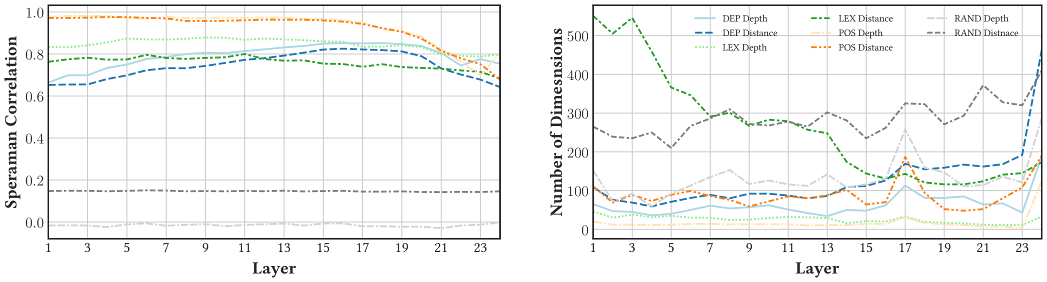


Figure 4.2: Spearman's correlations and number of non-zero *Scaling Vector*'s dimensions across layers for joint training.

weights closer to zero than $\epsilon = 10^{-4}$. We observed that dimension selection is not sensitive to the choice of low $10^{-30} < \epsilon < 10^{-3}$. Their elimination does not affect the results; correlations in Table 4.2 and Table 4.1 column **A** are practically equal. The dimensionality reduction is the strongest for lexical and positional depth probes, where subspaces with the rank of 19 and 20 respectively encode the structures as well as the whole embedding space with 1024 dimensions (Figure 4.2, lower). The number of selected dimensions is the highest in probing for random structures because a large capacity is required for memorization.

Identified dimensions should not be further pruned to preserve the signal Another question we pose is whether it would be adequate to shrink the subspace even further. For each objective, we choose and drop a random portion of parameters to examine how it would affect the predictions. We conduct a procedure similar to cross-validation, i.e., we repeatedly drop disjoint and exhaustive sets of dimensions and average results for each set at the end.[7] Table 4.2 shows that dimension dropping had the largest impact on positional probes: -0.458 for depth; the decrease is low for lexical distance, only -0.083. It suggests that the information necessary for predicting depth is less dispersed than for predicting distance.

Finding 3

Linguistic structures are encoded in sparse subspaces of latent embedding space.

[7] E.g. when we drop 25% of dimensions, we randomly choose four sets. Each dimension is exactly in one set, we average scores for four experiments.

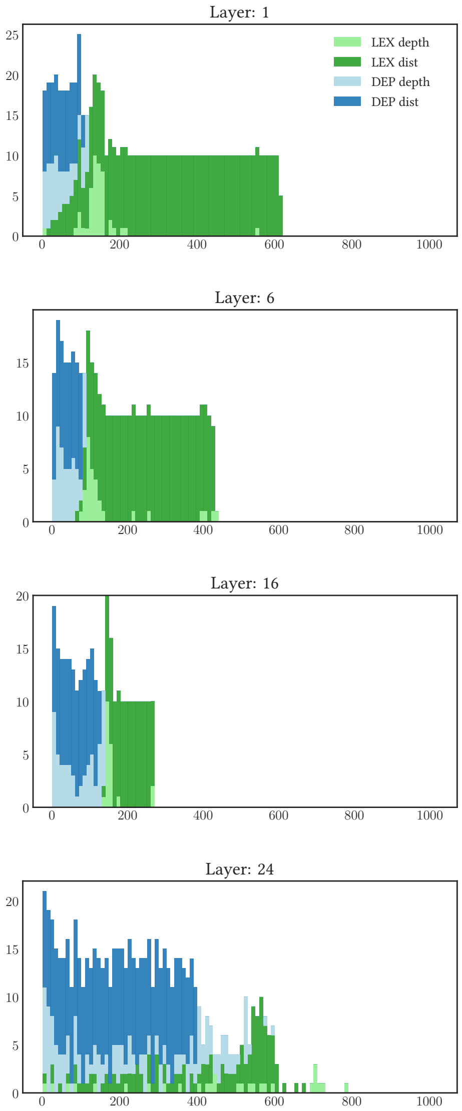


Figure 4.3: Histograms of dimensions selected by dependency and lexical *Scaling Vector* after joint training.

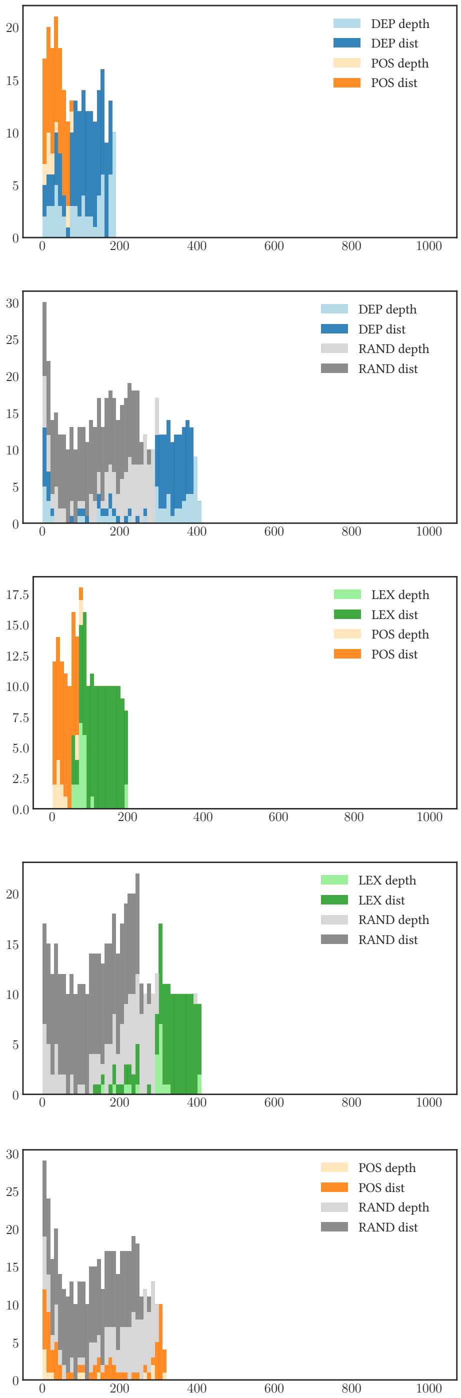


Figure 4.4: Histograms of dimensions selected by *Scaling Vector* after the joint training of probe on top of the 16th layer.

Semantic and lexical signals are encoded in separable subspeces Another outcome of joint training was the ability to examine relationships between subspaces for each of the objectives. Figure 4.3 shows histograms of the dimensions selected in lexical and dependency probes. Each bin of the histogram corresponds to 10 coordinates. The height of a bar (in one color) represents how many were selected for a specific task. The dimensions on the x-axis are ordered by the weighted absolute values of *scaling vector*s. [8]

We found that in layers 6 and 16 (which achieve the highest correlation in lexical and dependency, respectively), the histograms are disjoint, indicating that the layers' representations of dependency syntax and lexical hypernymy are orthogonal to each other in the embedding space. The orthogonality of task subspaces is less visible in the first layer and disappears almost entirely in the top one. In most layers, the depth subspace is included in the distance subspace for the same structural type. This behavior was expected as distance probing is more complex and therefore requires more capacity.

Finding 4

Lexical and syntactic structures are encoded in distinct, mutually orthogonal subspaces of the latent embedding space.

Finding 5

A space that encodes a simpler task (tree depth) is a subspace of space encoding a more complex one (tree distance).

In Figure 4.4 we present histograms for additional tasks at the model's 16th layer. The positional subspace has a sizable intersection with the syntactic one, yet only a few common dimensions with the lexical subspace. The connection can be attributed to the fact that dependency edges can often be inferred from words' relative positions. Probing for random structures is interlinked with other objectives. The sizes of shared subspaces for each pair can be found in Table 4.3.

[8] We weigh the values before sorting to keep together non-zero dimensions of each *scaling vector*, i.e., dependency depth values are multiplied by 1000, dependency distance 100, and lexical depth by 10. The weighting is performed only for visualization; the separation of linguistic information can be observed independently in Table 4.3.

4.4 Multilingual Orthogonal Probes

We utilize the new joint optimization capability to analyze how the encoding of linguistic phenomena is expressed across different languages in multilingual mBERT representations. Specifically, we ask whether linguistic information is uniformly encoded in the representations of various languages. And if this assumption does not hold: Is it possible to learn *orthogonal transformation* to align the embeddings?

4.4.1 Experiments

We evaluate three settings of multilingual *orthogonal probe* training. The approaches are sorted by expressiveness; the most expressive one makes the weakest assumption about the likeness of representations across languages:

In-Lang no assumption We train a separate instance of *orthogonal probe* for each language. Neither *scaling vector* nor *orthogonal transformation* is shared between languages.

MappedLangs isomorphity assumption We train a shared *scaling vector* for each probing task and a separate *orthogonal transformation* per language. If the embedding subspaces are orthogonal across languages, the orthogonal mapping will be learned during probe training, and the setting will achieve similar results as the previous one.

AllLangs: uniformity assumption Both the *scaling vector* and *orthogonal transformation* are shared across languages. If the same embedding subspace encodes the probed information across languages, the results of this setting will be on par with the first approach.

The first and the last approaches were proposed and analyzed for *structural probe*s by Chi et al. (2020). MappedLangs setting is possible thanks to our formulation of *orthogonal probe*s. For evaluation, we compute Spearman's correlations between predicted and gold depths and distances. Furthermore, we analyze the impact of two language-specific features on the results: a) size of the mBERT training corpus in a given language; b) typological similarity to English. The former is expressed in the number of tokens in Wikipedia. The latter is a Hamming similarity between features in WALS (Dryer and Haspelmath, 2013). [9]

[9] In this work, we consider all the features in the areas: Nominal Categories, Verb Categories, and Lexicon for computing a lexical typological similarity, and features in the areas: Nominal Syntax, Word Order, Simple Clauses, and Complex Sentences as a syntactic typological similarity.

Probed structures Similarly to the monolingual setting, we probe for semantic and lexical structures. For the former, we use dependency trees from Universal Dependencies available for multiple languages (Nivre et al., 2020). For lexical structure, we use a multilingual collection of WordNet (Miller, 1992) released as Open Multilingual WordNet (Bond and Foster, 2013). In both cases, we jointly optimize probes for depth and distance.

Choice of Layers We probe the representations of the 7th layer for dependency information and representations of the 5th layer for lexical information. These layers achieve the highest performance for the respective features.

4.4.2 Results

	EN	ES	SL	ID	ZH	FI	AR	FR	EU
Dependency Distance Spearman's Correlation									
In-Lang	.813	.859	.857	.856	.829	.791	.839	.856	.770
Chi et al. (2020)	.817	.859	-	.807	.777	.812	.822	.864	-
△ MappedLangs	-.001	-.002	.000	-.017	.001	-.001	-.002	-.003	.001
△ AllLangs	-.001	-.009	-.007	-.029	-.040	-.002	-.027	-.006	-.032
Chi et al. (2020)	-.011	-.011	-	-.018	-.060	-.010	-.037	-.011	-
Dependency Depth Spearman's Correlation									
In-Lang	.844	.869	.869	.856	.843	.824	.868	.875	.796
△ MappedLangs	-.004	-.002	-.003	-.002	.003	-.003	-.002	-.002	.000
△ AllLangs	-.008	-.010	-.010	-.011	-.037	-.006	-.033	-.006	-.029
Lexical Distance Spearman's Correlation									
In-Lang	.756	.840	.644	.722	.793	.646	.752	.791	.676
△ MappedLangs	.000	.002	-.025	-.003	.018	.022	.025	.000	-.001
△ AllLangs	-.036	-.021	-.045	-.061	-.003	-.022	.004	-.012	-.062
Lexical Depth Spearman's Correlation									
In-Lang	.846	.883	.779	.859	.868	.778	.922	.854	.848
△ MappedLangs	.011	-.014	.010	-.014	.013	.027	-.011	-.001	.017
△ AllLangs	-.017	-.044	-.030	-.118	-.064	.002	-.311	-.031	-.017

Table 4.4: Spearman's correlation between gold and predicted depths and distances. We probe the representations of the 7th layer for dependency information and representations of the 5th layer for lexical information. Correlations for dependency distance are compared with *structural probe* reported by Chi et al. (2020).

Using In-Lang probes for each language gives high Spearman's correlations across the languages. The MappedLangs approach brings only a slight difference for most of the configurations while imposing uniformity constraint (AllLangs) deteriorates the results for some of the languages, as shown in Table 4.4.

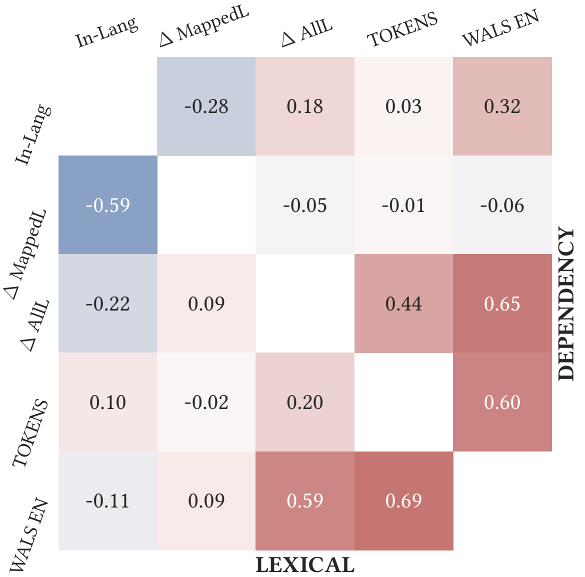


Figure 4.5: Pearson's correlation between results from Table 4.4 for each language and two language-specific features: typological similarity to English and number of tokens in Wikipedia. Correlations for dependency probes are in the upper-right triangle and for lexical probes in the lower-left triangle.

In Figure 4.5, we present Pearson's correlations between results from Table 4.4 and two language-specific features. The key observation is that topological similarity to English is strongly correlated with ΔAllLangs. Hence, a shared probe achieves a relatively good correlation for English, Spanish, and French. It shows that lexical and dependency information is uniformly distributed in the embedding space for those languages.

Finding 6

In multilingual BERT, representation of the same type of linguistic structure is shared across similar languages.

Notably, European languages are over-represented in the mBERT's pre-training corpus. Nevertheless, the size of pre-training corpora is correlated to a lesser extent with ΔAllLangs than WALS similarity. There is no significant correlation between ΔMappedLangs and typological similarity; the embeddings of diverse languages can be similarly well mapped into a shared space as these linguistically similar to English. Notably, we observe that some languages with lower performance of In-Lang probes can benefit from mapping (e.g., Chinese, Finnish, and Arabic when lexical distance is considered). We interpret it as a benefit of cross-lingual transfer from more resourceful languages.

	N	ZH	EU	SL	FI	AR
Lauscher+*	0	51.41	50.31	-	65.66	44.46
Wang et al.		-	-	67.86	65.45	-
+CLBT**		-	-	69.04	67.96	-
+FT* **		-	-	69.16	**69.16**	-
MappedL		34.44	39.10	35.44	37.33	40.95
AllL		**52.92**	**58.77**	**70.76**	64.60	**57.47**
Lauscher+*	10	57.73	57.23	-	65.13	71.00
MappedL		37.01	39.63	35.77	40.15	36.81
AllL		53.12	58.51	70.85	64.98	68.59
Lauscher+*	50	66.78	66.73	-	69.26	75.84
MappedL		45.07	50.02	55.09	49.32	57.77
AllLangs		53.63	59.07	70.43	65.02	68.81
Lauscher+*	100	69.91	65.70	-	70.25	78.50
MappedL		50.27	56.07	60.00	52.86	62.36
AllL		53.71	60.23	70.54	64.83	68.71
Lauscher+*	1000	80.12	74.75	-	78.00	83.85
MappedL		60.57	65.98	72.81	63.80	68.85
AllL		57.17	63.49	72.35	66.05	69.57

Table 4.5: UAS of extracted dependency trees. N is the number of in-language examples used for fine-tuning or probe optimization. Our two approaches are compared to the previous works that use a biaffine parser (Lauscher et al., 2020; Wang et al., 2019). We probed the representations of the 7th layer. *): fine-tuning of mBERT is used. **): A multilingual dictionary is used to align the embeddings.

Finding 7

The shared representation of linguistic structure is observed for languages typologically similar to English. Representations for typologically distinct languages are encoded in mutually isomorphic subspaces, which can be aligned by orthogonal transformation (i.e., rotation).

4.4.3 Application to Zero- and Few-shot Parsing

Our observation of considerable latent embeddings' similarity across languages inspired us to apply *orthogonal probes* to parsing in zero- and few-shot scenarios. In these settings, the probe is trained on a source language and then applied for parsing of the target language with no (zero-shot) or minimal number of in-language annotated examples (few-shot).

We examine cross-lingual transfer for parsing sentences in Chinese, Basque, Slovene, Finnish, and Arabic. For each of them, we train the probe in the remaining eight languages. In a few-shot setting, we also optimize 10 to 1000 examples from the target language. To get valid dependency parses, we first use the Maximum Spanning Tree algorithm on the predicted distances to obtain the tree structure. Then we apply the extension of the algorithm proposed by Kulmizev et al. (2020) to assign the direction to edges in the tree based on the predicted depths. We evaluate the correctness of trees using the unlabeled attachment score (UAS) described in Section 3.3.4.

Parsing Effectivness For all languages (except Finnish) in zero-shot configuration, our AllLangs approach is better than other works that utilize a biaffine parser (Dozat and Manning, 2017) on top of mBERT representations, as reported by Lauscher et al. (2020); Wang et al. (2019) (see Table 4.5). Without any supervision, our MappedLangs approach performs poorly because mapping cannot be learned effectively. When some annotated data is added to the training, the difference between AllLangs and MappedLangs decreases. We observe that between 100 and 1000 training samples are needed to learn the *orthogonal transformation*. Also, with higher supervision, we observe that the results reported by Lauscher et al. (2020) notably outperform our approach. The outcome was anticipated because they fine-tuned mBERT with a biaffine layer, which has a larger expressiveness than a probe. Therefore in this approach, the introduction of supervision is more advantageous than in probing.

Innovation 1

We propose a competitive method for zero- and few-shot parsing based on *orthogonal probes.*

4.5 Orthogonal Filters for Mitigating Gender Bias

We outline a method for disentangling the factual gender information and gender bias encoded in the representations. Following the formulation in Section 3.2, we aim to construct a filter that would preserve the factual gender information while diminishing gender bias. For that purpose, we leverage *orthogonal probe* training for factual and stereotypical gender to identify the distinct dimensions encoding each of the signals. This method was introduced in our paper: Limisiewicz and Mareček (2022), we outline it and present some of the key observations below.

4.5.1 Methodology

We hypothesize that semantic gender information (from pronouns) is encoded in the network distinctly from the stereotypical bias of gender-neutral words (Figure 3.2). We focus on interactions of gender bias and factual gender information in coreference cues of the following form:

[NOUN] examined the farmer for injuries because [PRONOUN] was caring.

In English, we can expect to obtain the factual gender from the pronoun. Revealing one of the words in the coreference link should impact the prediction of the other. Therefore, we can name two causal associations:

$$C_I\text{: } \text{bias}_{\text{noun}} \rightarrow \text{f. gender}_{\text{pronoun}}$$

$$C_{II}\text{: } \text{f. gender}_{\text{pronoun}} \rightarrow \text{bias}_{\text{noun}}$$

In our method, we will primarily focus on two ways bias and factual gender interact. For gender-neutral nouns (in association with type C_I), the effect on predicting masked pronouns would be primarily correlated with their gender bias. At the same time, the second association is desirable, as it reveals factual gender information and can improve the masked token prediction of a gendered word. We define two conditional probability distributions corresponding to these causal associations:

$$\begin{gathered} P_I(y_{\text{pronoun}}|X, s) \\ P_{II}(y_{\text{noun}}|X, f) \end{gathered} \tag{4.16}$$

Where y is a token predicted in the position of pronoun and noun, respectively; X is the context for masked language modeling. Variables s and f are bias and factual gender factors, respectively. We model the bias factor by using a gender-neutral biased noun. Below we present examples for introducing female and male bias: [10]

Example 1:

b_f **The nurse** examined the farmer for injuries because [PRONOUN] was caring.

b_m **The doctor** examined the farmer for injuries because [PRONOUN] was caring

Similarly, the factual gender factor is modeled by introducing a pronoun with a specific gender in the sentence:

Example 2:

[10] We use [NOUN] and [PRONOUN] tokens for a better explanation, in practice, they both are masked by the same mask token, e.g. [MASK] in BERT (Devlin et al., 2019).

f_f [NOUN] examined the farmer for injuries because **she** was caring.

f_m [NOUN] examined the farmer for injuries because **he** was caring.

We aim to diminish the role of bias in the prediction of pronouns of a specific gender. On the other hand, the gender indicated in pronouns can be useful in the prediction of a gendered noun. Mathematically speaking, we want to drop the conditionality on the bias factor in P_I from Equation 4.16, while keeping the conditionality on the gender factor in P_{II}.

$$\begin{aligned} P_I(y_{\text{pronoun}}|X, b) &\rightarrow P_I(y_{\text{pronoun}}|X) \\ P_{II}(y_{\text{noun}}|X, f) &\not\rightarrow P_{II}(y_{\text{noun}}|X) \end{aligned} \tag{4.17}$$

To decrease the effect of gender signals from words other than pronoun and noun, we introduce a baseline, where both pronoun and noun tokens are masked:

Example 3:

$\varnothing$ [NOUN] examined the farmer for injuries because [PRONOUN] was caring.

4.5.2 Filtering Gender Bias

To mitigate the influence of bias on the predictions Equation 4.17, we focus on the latent representations of the language model. We aim to inspect contextual representations of words and identify their parts that encode the causal associations C_I and C_{II}.

We want to approximate gender information introduced by a gendered pronoun f (factual) and a gender-neutral noun s (bias). The variable f takes the values -1 for female pronouns, 1 for male ones, and 0 for gender-neutral “they”. The variable s is the stereotype value associated with each of the words, for definition see Section 2.1 in Limisiewicz and Mareček (2022). We denote the element of *orthogonal probe*:

- V: *orthogonal transformation*, mapping representation to new coordinate system.
- d: *scaling vector*, element-wise scaling of the dimensions in a new coordinate system. We assume that dimensions that store probed information are associated with large *scaling vector*.

The probing losses are the following:

$$\begin{aligned} L_I &= \Big| ||\bar{d}_s \odot (V \cdot (h_{s,P} - h_{\varnothing,P}))||_d - s \Big| \\ L_{II} &= \Big| ||\bar{d}_f \odot (V \cdot (h_{f,N} - h_{\varnothing,N}))||_d - f \Big|, \end{aligned} \tag{4.18}$$

where, $h_{s,P}$ is the vector representation of the masked pronoun in example 1; $h_{f,N}$ is the vector representation of the masked noun in example 2; vectors $h_{\varnothing,P}$ and $h_{\varnothing,N}$ are the representations of masked pronoun and noun respectively in baseline example 3. To account for negative values of target factors (s and f) in Equation 4.18, we generalize distance metric to negative values in the following way:

$$||\overrightarrow{v}||_d = ||\max(\overrightarrow{0}, \overrightarrow{v})||_2 - ||\min(\overrightarrow{0}, \overrightarrow{v})||_2 \tag{4.19}$$

We jointly probe for both objectives (orthogonal transformation is shared). Subsequently, we use filters described in Section 4.2.6 to marginalize the stereotypical signal identified by d_I while keeping gender information corresponding to d_{II}.

4.5.3 Experiments

We construct *orthogonal filter*s on top of from one up to four top layers of large BERT. We examine two setting: filtering stereotypical gender F_{-s} (Equation 4.13), filtering stereotypical gender while preserving factual signal $F_{-s,+f}$ (Equation 4.15). As a training set, we use sentences from WinoMT (Stanovsky et al., 2019). They denote the position of the pronoun and noun in the sentence, making it straightforward to use masking according to the pattern presented in Examples 1-3. To evaluate the effect of filtering on model performance and gender signal, we evaluate the accuracy of masked language model prediction in two settings: general and gendered.

General MLM We compute prediction accuracy for the masked tokens in the test set from English Web Treebank UD Silveira et al. (2014b) consisting of 2077 sentences.

Gendered MLM We evaluate the capability of the model to infer the personal pronoun based on the context. For that purpose, we use the GAP Coreference Dataset Webster et al. (2019) with 8908 paragraphs. In each test case, we mask a pronoun referring to a person usually mentioned by their name, while professional mentions are the source of stereotype in the prediction.

4.5.4 Results

The results in Table 4.6 show that filtering out bias dimensions moderately decreases MLM accuracy up to 0.037. Using factual gender-preserving filtering decreases the drop in results.

Setting	FL	Gendered MLM			General MLM
		Overall	Male	Female	
original	-	0.799	0.**816**	0.781	**0.516**
-s	1	0.690	0.757	0.624	0.515
	2	0.774	0.804	0.744	0.504
	4	0.747	0.770	0.724	0.479
-s +f	1	0.754	0.782	0.726	0.515
	2	0.785	0.801	0.769	0.510
	4	**0.801**	0.807	**0.794**	0.489

Table 4.6: Top-1 accuracy for general domain MLM in EWT UD Silveira et al. (2014b) and for gendered pronoun prediction in GAP dataset Webster et al. (2019). FT is the number of the model's top layers for which filtering was applied.

In gendered predictions, we observe a more significant drop in results when applying the F_{-s} filter. The deterioration can be alleviated by omitting factual gender dimensions in the filter. This setting can even bring improvement over the original model. Our explanation of this phenomenon is that filtering can decrease the confounding information from stereotypically biased words that affect the prediction of correct gender.

Innovation 2

Filters based on *orthogonal probes* can disentangle factual and stereotypical gender signals allowing to preserve the first while mitigating the former. We obtain a less biased representation with a small change in the model's general performance.

4.6 Implementation Details

This section describes the implementation details of *orthogonal probe* that were shared across all the settings described in this Chapter. Optimization is conducted with Adam (Kingma and Ba, 2015) with an initial learning rate of 0.02, we use batches of size 12 We use learning rate decay and early-stopping mechanism: if validation loss does not achieve a new minimum after an epoch, the learning rate is divided by 10. After three consecutive learning rate updates not resulting in a new minimum, the training is stopped.

Orthogonality Regularization In our experiments, we took λ_O equal to 0.05.[11] The regularization converged early during the gradient optimization. Hence we can assume that matrix V is orthogonal.

Sparsity Regularization By default, we set $\lambda_S = 0$. Only for the experiments summarized in Table 4.2, we add sparsity regularization by setting λ_S to a positive value (0.005, 0.05, or 0.1) when DSO drops below 1.5 during the training. This mechanism prevents weakening orthogonality constraints in early epochs.

4.7 Conclusions

The line of research on *orthogonal probe*s provides new findings regarding the nature of the information encoded in the hidden embeddings of the model. Firstly, we show that many types of information can be disentangled from each other by using *orthogonal transformation* in probing. In our results, we specifically focused on the case of syntactic and lexical structures.

Although specific task-related dimensions are contained in orthogonal subspaces, the representations are considerably similarly distributed across languages. Our experiments with multilingual models show that the distribution is shared across languages with similar typological features.

We show two applications of *orthogonal probe*s based on the presented findings. The first is zero-shot cross-lingual parsing which is realized by constructing trees based on the predictions of dependency and depth probes. The relatively high performance of our approach is the result of a similar distribution of latent embeddings across languages. The second application is filtering unwanted gender bias. For this purpose, we introduce *orthogonal filter*s that allow us first to disentangle important gender-related signals from unwanted stereotypical bias. Subsequently, we construct a filter based on *scaling vector*s coefficients to remove the stereotypical gender signal while preserving the factual one (as defined in Section 4.5).

[11] We experimentally checked that ten times smaller and ten times larger values of λ_O do not affect the orthogonality of matrix V and lead to the same results.

5 Feed Forward

In this Chapter, we further pursue decreasing gender bias in language models while preserving their high performance in language understanding tasks. We turn our attention to larger causal language models from LLaMA family (Touvron et al., 2023) with 7 to 65 billion parameters. Due to recent advances in model scaling, feed-forward overtook the embedding layers as the component with the largest share of parameters in Transformer models (Geva et al., 2021). Therefore, in work Limisiewicz et al. (2023b) we posit the question of whether the parameter size of the feed-forwards is reflected by their capability to store information learned during pre-training.

In particular, we study the encoding of the gender bias in the language models, and we perform the analysis with the use of *causal tracing* (Vig et al., 2020), a method inspired by *causal mediation analysis* which was previously established in causality literature (Pearl, 2009). To explain the title of this chapter, we upfront reveal that *causal tracing* identified mid-upper feed-forward layers as the most prone to encode gender bias in the language models.

5.1 Causal Tracing

The *causal tracing* was inspired by the causality literature, in particular, *causal mediation analysis* (Pearl, 2001; VanderWeele, 2015). The work of Vig et al. (2020) builds upon *causal mediation analysis* to identify the way the information flows from the input to the output of a language model. They particularly focus on identifying the model components crucial for passing the information through the model, so-called *mediators.* They propose to use *causal mediation analysis* for that purpose, which is the method of measuring the change in the model's predictions upon intervention in the *mediator* (i.e., specific component). In *causal tracing*, *direct effect* of the mediator h measures the change in output y resulting from intervening in the input x while holding a mediator h fixed, while its *indirect effect* quantifies the change caused by setting h to a value it would have if we performed intervention in the input x. The graph of the assumed causal relation between the input, mediator, and output is presented in the causal graph in Figure 5.1.

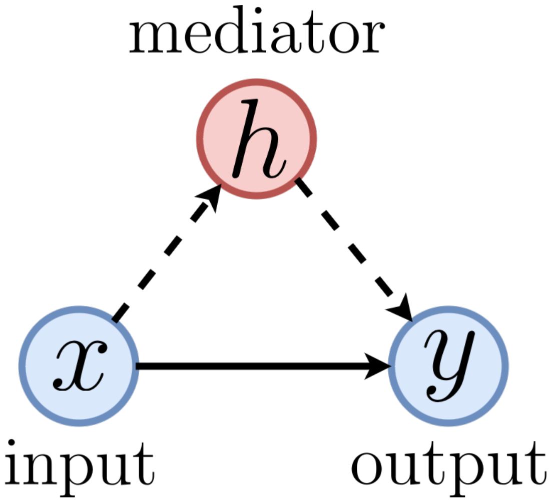


Figure 5.1: A causal structure assumed in *causal mediation analysis* applied to LM. In our experiments, we will measure the *indirect effect* on the model's output y by setting latent representation h (mediator) to the value it would take under intervention in input x, while keeping the input x unchanged.

We will specifically consider the *causal tracing* formulation used by Meng et al. (2022), who analyze the mediator's effect on the LM output (i.e. next token prediction) given an input prompt stimulating specific behavior of the model, e.g. biased prediction. In causal tracing, we need to identify the token stimulating the model to produce the expected output. For instance, we can consider the token connected with world-knowledge query (e.g. "*Vienna*" in prompt "*Vienna is the capital of*") and its impact on predicting factual continuation ("*Austria*"). For tracing, we need to perform a corruption that would erase the signal from the token, by adding Gaussian noise to the corresponding input embedding. The method is performed in the following steps:

1. Perform a *clean run* (forward-pass without any corruption) and collect all the activations at all layers and tokens.

2. Perform a *corrupted run* by adding noise to the tokens corresponding to a specific signal (e.g. common sense fact, or stereotype).

3. Perform *corrupted runs* with restoration: at each step, we restore the activations from the *clean run* of each module (e.g. *attention head*, MLP) at one particular layer and temporal position (h_t). And check the effect on predictions.

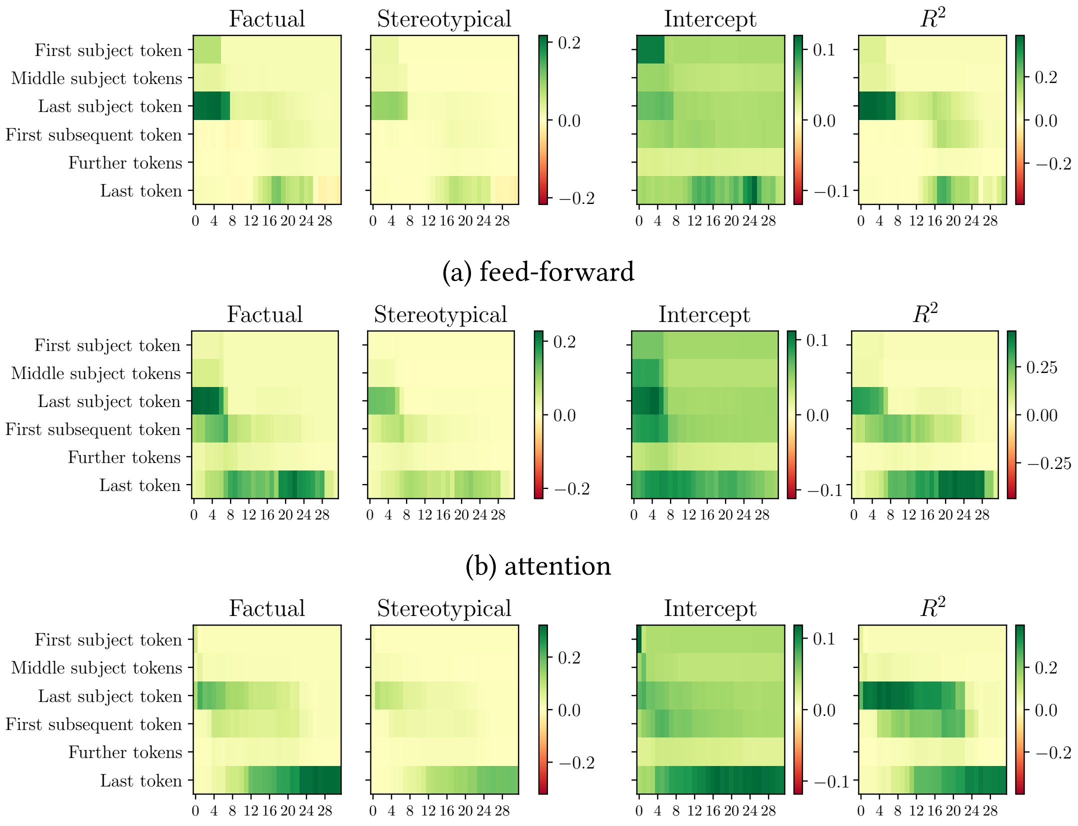
In each step, we compute the effect of adding the signal in question (e.g. input with clean representation of word "*Vienna*") to one module. Therefore we examine modules' *inddirect effect*. Alternatively, we can examine the impact of noising the representation of the specific module representation, which would be its *direct effect*. The overall change between *clean* and *corrupted* runs is called *total effect*. Please note that in this formulation, *clean run* corresponds to performing intervention in model input as it reveals information, e.g., from the word "*Vienna*". Past results indicated that investigating *indirect effects* is more informative in the search of mediators in Transformer models (Meng et al., 2022).

5.1.1 Causal Tracing for Bias Location

To identify the components storing gendered associations, we perform *causal tracing* for gender bias in text generation. As a stimulus of gendered prediction, we use the prompts X consising the profession words introduced in Section 3.8.4, see an example in Figure 5.3a. For each prompt, we compute the empirical gender score, as $y_e(X) = P_M(o_+|X) - P_M(o_-|X)$, where o_+ and o_- are the probabilities of the model predicting male and female pronouns, respectively. We then fit a linear model (Equation 5.1) across all prompts X to obtain coefficients a_s and a_f measuring the impact of the stereotypical and factual gender cues on the model's output.

$$y_e = a_s \cdot x_s + a_f \cdot x_f + b_0 \tag{5.1}$$

Then we conduct the causal tracing using a_s and a_f coefficients to identify the components passing gender information (stereotypical and factual) to the model's output. Specifically in step 2 of the procedure described in Section 5.1, we add noise to the embeddings corresponding to the profession words. Then for step 3, we compute the *indirect effects* by reintroducing the model's activations from the *clean run* to each module at temporal position h_t. Following Meng et al. (2022), we aggregate token positions into six groups shared across the whole dataset: first, middle; and last subject token; the first token following the subject; all the tokens following the subject; and the last token.



(a) feed-forward

(b) attention

(c) Layer

Figure 5.2: Causal tracing of modules' *indirect effect* of *factual* and *stereotypical* signals in LLaMA 7B. Effects are measured by the linear coefficient a_s and a_f introduced in the Equation 5.1. The *indirect effect* is calculated by reintroducing *clean representation* to the output of specific components (FF or Att. or whole layer) and token position.

5.1.2 Results

We show the coefficient of the linear model in Table 5.1. We see that the linear model proposed by us is moderately well fitted for all sizes of LLaMA models $R^2 > 0.35$. For all sizes, the factual coefficient is higher than the stereotypical one hinting that the models are more influenced by semantical than stereotypical cues ($a_f > a_s$). Also, we observe a positive intercept b in all cases, showing that LLaMA models are more likely to predict male than female pronouns. Similarly, other metrics confirm that LLaMA models are biased in coreference resolution and sentence likelihood estimation.

Finding 8

LLaMA models' predictions show the presence of gender bias.

In Figure 5.2a, we observe the indirect effect of multi-layer perceptrons (MLPs) (in feed-forwards) in each layer and token position of the 7B model. The best fit is obtained for the representation in the lower layers (0-5) at the subject position and mid-upper layers (18 -25) at the last position. In the search for stereotypically biased components, we direct our attention to the mid-upper layers because they appear to convey less signal about factual gender. We also expect that the information stored in those FF layers is more likely to generalize to unseen subjects. Interestingly, the last layers manifest weak negative slope coefficients, suggesting that these FFs tend to counter the bias of the models.

We further see in the bottom part of Figure 5.2 the results of *causal tracing* for attention and the whole layer. For those components, the high indirect effects are distributed more extensively across both token positions and layers, indicating that they primarily reflect bias from the FFs. For larger models, we observe analogous patterns shifted according to the total layer count. The results are presented in Figures 5.5, 5.6, and 5.7.

Finding 9

The strongest encoding of gender-related information is observed in the mid-upper feed-forward layers of LLaMA models.

5.2 Debiasing the Feed-Forward Layers with DAMA

We introduce the algorithm that decreases bias in language models by directly editing the model weights. This section describes our method based on projection-based intervention in selected layers: Debiasing through Model Adapatation (DAMA). Further, we provide theoretical and empirical backing for the method's effectiveness.

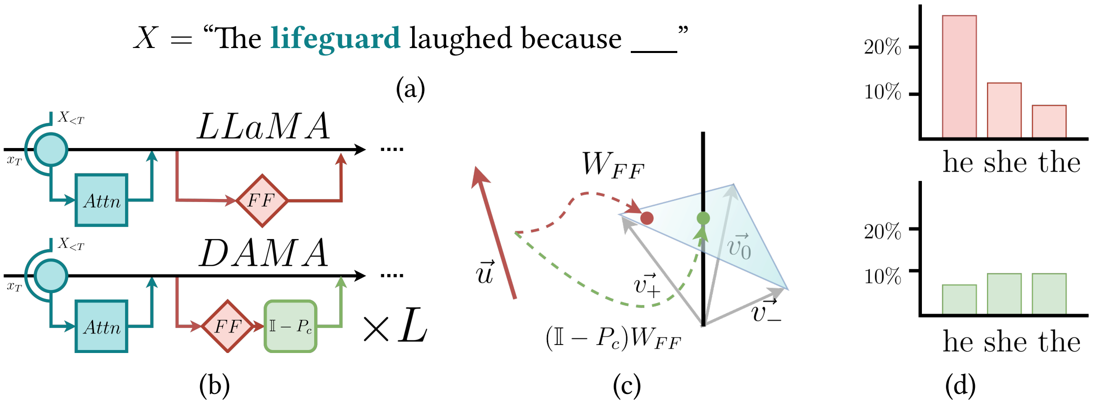


Figure 5.3: Schema (b) shows DAMA intervention in LLaMA layer. Even though $\mathbb{I} - P_c$ is depicted as a separate module, in practice, it is multiplied with the output matrix of a feed-forward layer (W_{FF}). Therefore, DAMA is neutral to the model's parameter count and architecture. (a) We show the behavior of the model when presented with a stereotypical prompt. Specifically, (c) shows the projections of the FF output latent vector ($\vec{u}$) onto the output space. With DAMA (lower arrow), we nullify the gender component of the representation. It results in balanced probabilities of gendered tokens in the model's output, as shown in (d).

5.2.1 Obtaining Stereotype Keys and Gendered Values

Following the convention from Geva et al. (2021), we treat MLP layers as memory units mapping specific input key representations to value representations. Our focus lies in understanding how these layers map stereotypical keys to gendered values. As our choice of keys, we take prompts introduced in Section 3.8.4, which carry stereotypical signals. The values are the output vectors corresponding to one of the personal pronouns (male, female, or neutral).

To compute the stereotypical key at lth layer, we feed the stereotypical prompt X up to l layer's feed-forward MLP (FF_l) to obtain its vector representation. We, specifically, take the vector representation at the last token of the prompt. We denote stereotypical keys as $u \in \mathbb{R}^{d_{FF}}$ following the convention from Figure 5.3c.[1]

5.2.2 Obtaining Projection on Stereotype Subspace with PLS

To identify the stereotype subspace, we concatenate value vectors for each pronoun (male, neutral, and female) across all prompts to obtain gendered value matrices V_+, V_0, and V_-. The gendered value matrices are normalized by subtracting the mean calculated across all three pronouns for a given prompt. We also concatenate key

[1] Notably, for clearer distinction of MLP input and output vectors, we use u and v, instead of h and h' used throughout the thesis.

vectors for all prompts into one matrix U. Then, we multiply it by the feedforward's output matrix denoted $W_{FF,out,l}$:

$$W_{FF,out,l} \cdot U \rightarrow \hat{U} \tag{5.2}$$

We concatenate V_{+}, V_0, and V_{-} together and concatenate $\hat{U}$ three times. We use the partial least squares (PLS) algorithm to identify the linear mapping B_1 maximizing correlation between stereotypical keys $[\hat{U}, \hat{U}, \hat{U}]$ and gendered values $[V_{+}, V_0, V_{-}]$:

$$[V_{+}, V_0, V_{-}] \approx_{\text{PLS}} B_1 \cdot [\hat{U}, \hat{U}, \hat{U}] + B_0 \tag{5.3}$$

By definition of PLS, B_1 identifies the stereotypical directions most correlated with gendered values.[2] Therefore, we compute the matrix projecting representation on subspace orthogonal to the one spanned by d_c first columns of B_1 to nullify the stereotypical signal. For brevity, we denote the trimmed matrix as $B_1^{d_c} = B_1[:, :d_c]$. The projection is given by the equation:

$$P = \mathbb{I} - P_c = \mathbb{I} - B_1^{d_c}(B_1^{d_c T} B_1^{d_c})^{-1} B_1^{d_c T} \tag{5.4}$$

Finally, we perform the model editing by multiplying lth MLP feed-forward matrix $W_{FF,out,l}$ by the projection matrix P, see Figure 5.3c. Our algorithm: DAMA is based on iterative computation and applying projections to feed-forwards of multiple subsequent MLP layers. It changes neither the model's architecture nor parameter sizes, as the result of matrix multiplication is of the same dimensionality as the original feed-forward matrix.

5.2.3 Theoretical Perspective

We show theoretical guarantees that multiplying linear feed-forward matrix $W_{FF,out,l}$ by projection matrix P will be the optimal mapping between keys (U) and values (V), fulfilling that $W_{FF,out,l} \cdot U$ is orthogonal to the guarded bias subspace $\mathcal{C}$.

Theorem 1. *Assume that we have a linear subspace $\mathcal{C} \subseteq \mathbb{R}^o$. Given a n-element key matrix $U \in \mathbb{R}^{i \times n}$ a value matrix $V \in \mathbb{R}^{o \times n}$, we search a mapping matrix $W \in \mathbb{R}^{o \times i}$ minimizing the least squares and satisfying $\forall_{i=1}^{n} W u_i \perp \mathcal{C}$. Specifically, we solve:*

$$\hat{W} = \underset{W}{\operatorname{argmin}} ||WU - V||_F^2 \quad \textit{such that} \quad \forall_{i=1}^{n} W u_i \perp \mathcal{C}$$

[2] Matrix B_0 can be used to normalize the value matrix. However, we have noticed that its values become nearly zero due to the earlier normalization of $[V_{+}, V_0, V_{-}]$.

This equation is solved by:

$$\hat{W} = (\mathbb{I} - P_c)VU^T(UU^T)^{-1}$$

Where P_c is a projection matrix on a subspace $\mathcal{C}$.

Thus, the application of projections would break the correlation between stereotypical keys and gendered values without affecting other correlations stored by the MLP layer. To prove the theorem, we present a theorem that will help prove Theorem 1.

Theorem 2 (Ordinary Least Square Problem)**.** *Given a n-element key matrix $U \in \mathbb{R}^i$ and a value matrix $V \in \mathbb{R}^{o \times n}$, we search for a mapping matrix $W \in \mathbb{R}^{o \times i}$ minimizing least squares. Specifically, we solve:*

$$\hat{W} = \text{argmin}\, ||WU - V||_F^2$$

This equation is solved by:

$$\hat{W} = VU^T(UU^T)^{-1}$$

The theorem says that $VU^T(UU^T)^{-1}$ solves the regular mean square error problem of mapping prompt keys to values corresponding to the model's output. Due to gradient optimization in the model's pre-training, we can assume that in general case $W_{FF,out,l} = VU^T(UU^T)^{-1}$. The proof can be found in the statistical literature, e.g., in Goldberger et al. (1964). Equipped with Theorem 2 we can prove Theorem 1:

Proof. Without loss of generality, we consider a case where $n = 1$, i.e., U and V are column vectors. For clarity, we will denote those vectors $u \in \mathbb{R}^i$ and $v \in \mathbb{R}^o$ respectively. Therefore, we aim to solve an equation:

$$\hat{W} = \underset{W}{\text{argmin}}\, ||Wu - v||_F^2 \quad \text{such that} \quad Wu \perp \mathcal{C} \tag{5.5}$$

Note that we can substitute the Frobenius norm with the Euclidean norm and decompose vector v into the sum of two orthogonal vectors.

$$||Wu - v||_F^2 = ||Wu - v||^2 = ||Wu - (\mathbb{I} - P)v - Pv||^2 \tag{5.6}$$

We infer that $Wu - (\mathbb{I} - P)v \perp \mathcal{C}$ from a) $Wu \perp \mathcal{C}$ (5.5); and b) $(\mathbb{I} - P) \perp \mathcal{C}$ as P is projection matrix on $\mathcal{C}$. Moreover, from the properties of linear projection, we have $Pv \in \mathcal{C}$. We note thus that $Wu - (\mathbb{I} - P)v \perp Pv$. From Pythagoras Theorem we can reweite 5.6 as: [3]

$$||Wu - (\mathbb{I} - P)v - Pv||^2 = ||Wu - (\mathbb{I} - P)v||^2 + ||Pv||^2 \tag{5.7}$$

In argmin notation, we can omit the second part of the formula because it doesn't depend on W

$$\hat{W} = \operatorname*{argmin}_W ||Wu - v||^2 = \operatorname*{argmin}_W ||Wu - (\mathbb{I} - P)v||^2 \tag{5.8}$$

Now, we can apply the same steps to all the columns in $U = [u_1, \ldots, u_n]$ and $V = [v_1, \ldots, v_n]$, to obtain:

$$\hat{W} = \operatorname*{argmin}_W ||WU - (\mathbb{I} - P)V||_F^2 \tag{5.9}$$

Based on Theorem 2 it is solved by $\hat{W} = (\mathbb{I} - P)VU^T(UU^T)^{-1}$. We can easily obtain this result by substituting V by $(\mathbb{I} - P)V$ in the theorem.

Lastly, it can be shown that for any vector $x \in \mathbb{R}^i$ we have $\hat{W}x \perp C$ from the fact that applying P projection to $\hat{W}x$ always produces a null vector:

$$P\hat{W}x = P(\mathbb{I} - P)VU^T(UU^T)^{-1} = (P - P)VU^T(UU^T)^{-1} = \vec{0} \tag{5.10}$$

□

Finding 10

Linear layers can be adapted so to mitigate the presence of specific unwanted correlations while keeping other correlations learned in training.

5.2.4 Empirical Perspective

After providing the theoretical guarantees, we proceed to investigate DAMA utility in practice, by conducting a series of experiments on the LLaMA models.

[3] Pythagoras Theorem for vectors states that for a pair of orthogonal vectors: $\vec{a} \perp \vec{b}$, we have $||\vec{a}||^2 + ||\vec{b}||^2 = ||\vec{a} + \vec{b}||^2$. In 5.6, we substitute $\vec{a}$ with $Wu - (\mathbb{I} - P)v$ and $\vec{b}$ with Pv.

Effectivness We apply DAMA to FFs in approximately one-third of the model's upper layers (in LLaMA 7B layers 21 - 29 out of 32 with projection dimensionality $d_c = 256$). In Section 5.1.2, we have shown that those layers are the most prone to stereotypical bias. We check the impact of DAMA on bias coefficients of the linear model and LM perplexity measured on Wikipedia texts (defined in Section 3.2). Furthermore, we evaluate the modified model on a set of diverse downstream tasks described in detail in Chapter 3. In the choice of tasks, we focused both on gender bias (WinoBias, StereoSet) and language understanding evaluation (OBQA, ARC, MMLU).

Baselines We compare the method with a similar model editing method **MEMIT** (Meng et al., 2023) and a parameter-efficient fine-tuning via **LoRA** (Hu et al., 2022). In both baselines, we optimize the model to predict a randomly sampled pronoun when presented with a biased prompt.

Choice of Layers and Dimensionality We analyze how the results vary depending on the number of layers selected for debiasing. Due to the iterative character of intervention, we always start editing at the fixed layer (22 in LLaMA 7B) and gradually add subsequent layers. Further, we check the effect of the number of projection dimensions (d_c) in the power sequence from 32 to 1024.

Scaling Lastly, we examine the algorithm's performance for larger scales of LLaMA model: 13B, 30B, and 65B.

5.2.5 Results

Effectivness DAMA effectively decreases the gender bias of the model while preserving its performance on other tasks, as seen in Table 5.1. Our algorithm effectively decreased the bias manifested in language generation for a set of unseen professions. Moreover, DAMA significantly mitigates bias in StereoSet and WinoBias. In the latter task, general accuracy decreases, presumably due to the weakening of the stereotypical cue contributing to correct predictions in numerous test examples.

Innovation 3

We introduce an efficient method (DAMA) for significantly decreasing the presence of various types of bias in language models.

	Bias in LM				WinoBias			StereoSet gender		
	↓ a_s	↑ a_f	↓ b	↓ R^2	↑ Acc	↓ ΔS	↓ ΔG	↑ lms	↓ ss	↑ ICAT
LLaMA 7B	0.235	**0.320**	0.072	0.494	**59.1%**	40.3%	3.0%	95.2	71.9	53.7
DAMA	**-0.005**	0.038	**-0.006**	**0.208**	57.3%	**31.5%**	2.3%	95.5	**69.3**	**58.5**
± (std)	0.004	0.004	0.004	0.026	0.5%	0.9%	0.7%	0.3	0.8	1.5
MEMIT	0.209	0.282	0.071	0.497	59.3%	40.5%	3.3%	95.6	72.0	53.6
LLaMA 13B	0.270	0.351	0.070	0.541	70.5%	35.7%	-1.5%	95.2	71.4	54.4
DAMA	0.148	0.222	0.059	0.472	66.4%	31.1%	-1.1%	94.4	68.6	59.4
LLaMA 30B	0.265	0.343	0.092	0.499	71.0%	36.0%	-4.0%	94.7	68.4	59.9
DAMA	0.105	0.172	0.059	0.471	63.7%	26.7%	-3.7%	94.8	65.7	65.0
LLaMA 65B	0.249	0.316	0.095	0.490	73.3%	35.7%	1.4%	94.9	69.5	57.9
DAMA	0.185	0.251	0.100	0.414	71.1%	27.2%	0.8%	92.8	67.1	61.1

Table 5.1: Bias evaluation for the LLaMA models and their debiased instances. Significance analysis for the 7B model was performed by running DAMA with five random seeds. We bold the score for the original model or DAMA, whichever is better if there are more than two standard deviations apart. We underline the best value in each column.

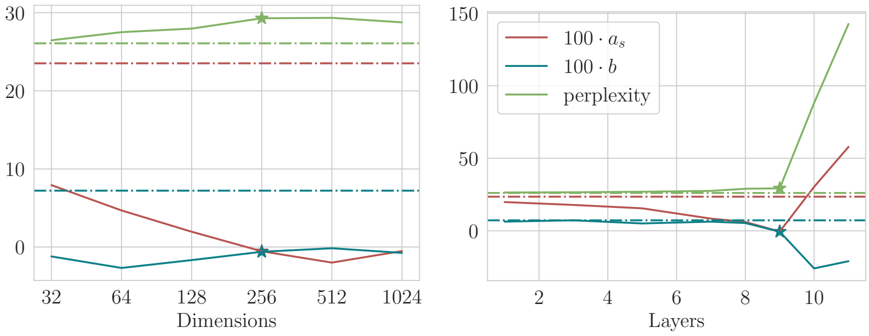
	LM	Downstream			
	↓ PPL	↑ ARC-C	↑ ARC-E	↑ OBQA	↑ MMLU
LLaMA 7B	**26.1**	42.2	**69.1**	57.2	30.3
DAMA	28.9	41.8	68.3	56.2	30.8
± (std)	0.2	0.4	0.2	0.5	0.5
MEMIT	26.1	42.7	68.9	57.0	30.2
LLaMA 13B	19.8	44.9	70.6	55.4	43.3
DAMA	21.0	44.7	70.3	56.2	43.5
LLaMA 30B	20.5	47.4	72.9	59.2	–
DAMA	19.6	45.2	71.6	58.2	–
LLaMA 65B	19.5	44.5	73.9	59.6	–
DAMA	20.1	40.5	67.7	57.2	–

Table 5.2: Performance evaluation for the LLaMA models and their debiased instances. The significance analysis was performed as described in Table 5.1.

Our observations confirm that MLP layers contain stereotypical correlations responsible for multiple manifestations of bias. Furthermore, we observe in Table 5.2 that the algorithm causes a slight deterioration in general language modeling measured by perplexity. It has a minor reflection in performance for downstream tasks. The altered model achieves a slightly lower score, yet differences are statistically significant only for one task (ARC-E). Therefore, we can conclude that DAMA does not impact the model's ability in question-answering tasks.

Innovation 4

DAMA does not harm the models' performance in language understanding tasks.



(a) Number of layers fixed at 9

(b) Dimensionality fixed at 256

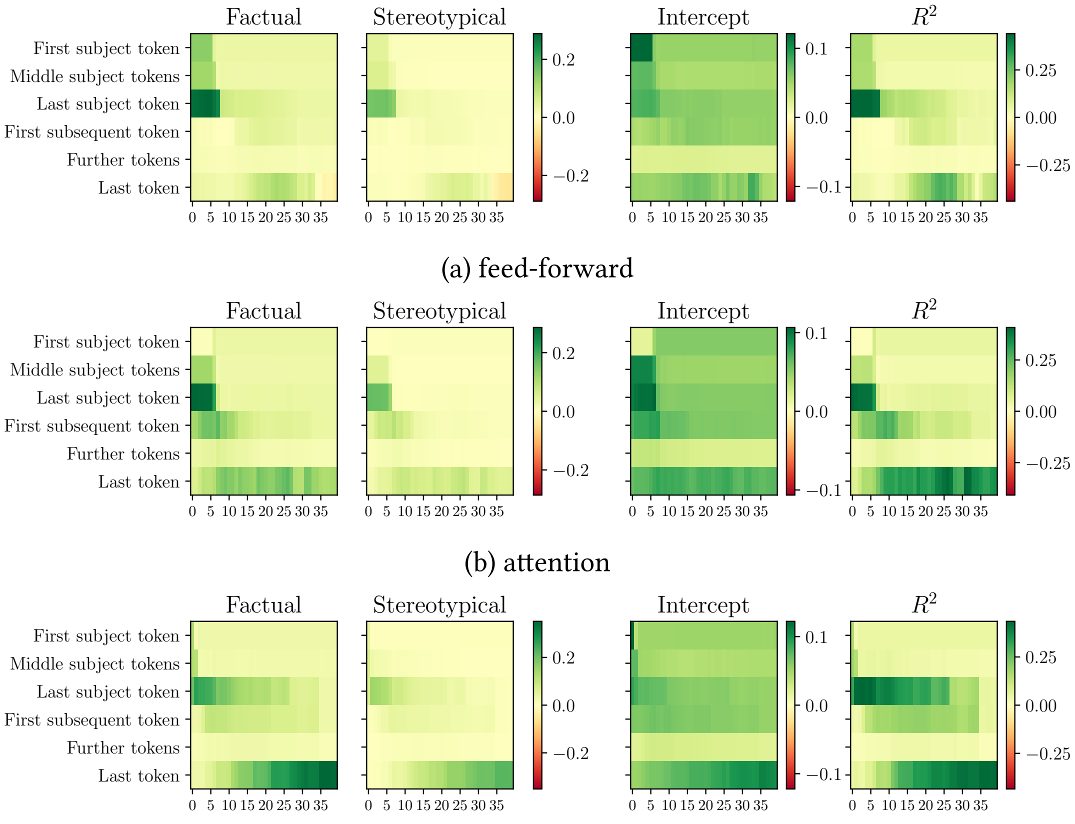
Figure 5.4: The effect of applying DAMA to LLaMA 7B model on performance and bias in language modeling. We measured bias on gendered prompts by coefficients: a_s and b from Equation 3.14, the causal language modeling capabilities are measured by perplexity. Stars mark the performance of the model picked for further evaluation. The dashed line corresponds to the scores of the original LLaMA 7B model.

Baselines In contrast to DAMA, MEMIT has a minor effect on bias benchmarks. We think it is because it is aimed to alter information specific to key-value pairs selected for intervention. Therefore, the intervention performed on the training set of professions does not generalize to unseen professions or other types of gender bias. LoRA manifests stronger debiasing properties, coming close to the results of DAMA in multiple bias metrics, and performs better in StereoSet ss and $ICAT$. Nevertheless, fine-tuning significantly deteriorates perplexity and the performance in language understanding tasks.

Choice of Layers and Dimensionality In Figure 5.4, we observe that the choice of the number of layers for debiasing and the dimensionality of projection affect both parameters. Expanding the depth (number of layers) and width (dimensions) of the intervention increases the intensity of debiasing, i.e., decreases a_s and b coefficients and negatively impacts perplexity. Interestingly, we observe a negative impact on both measured aspects when applying DAMA on the two last layers of the models. As noted in Section 5.1, the MLPs in those layers tend to counter bias in the original model.

Finding 11

The last layers of the models slightly decrease the stereotypical cues in the predictions. Moreover, they should not be adapted to preserve high language modeling performance.



(a) feed-forward

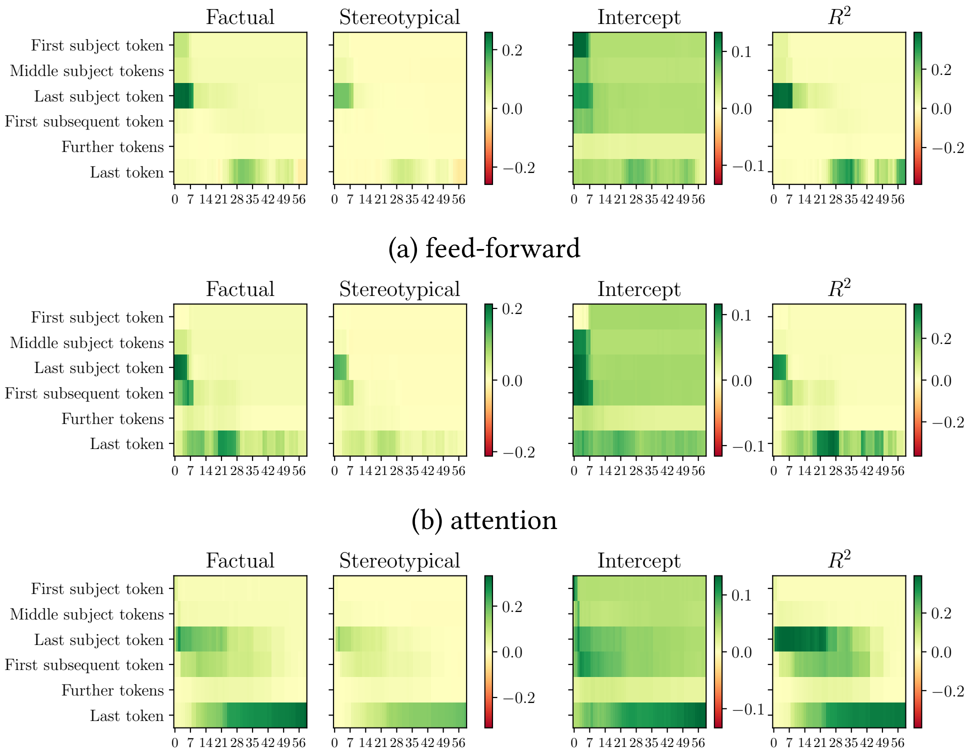
(b) attention

(c) Layer

Figure 5.5: *Causal tracing* analysis of gender signals in LLaMA 13B

Model size	# layers	layers adapted	# dimensions	projected dimensions
Llama 7B	32	21 – 29	4096	256
Llama 13B	40	26 – 36	5120	512
Llama 30B	60	39 – 55	6656	1024
Llama 65B	80	52 – 71	8192	2048

Table 5.3: Number of layers and latent dimensions of LLaMA models compared with the number of DAMA adapted layers and the projected dimension d_c.

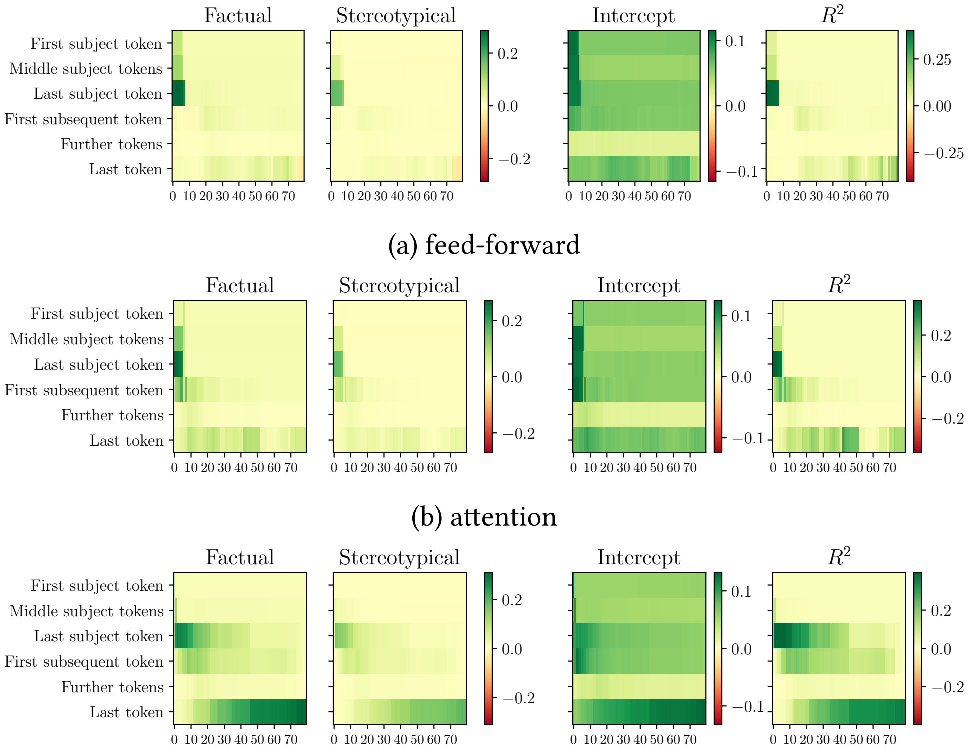


(a) feed-forward

(b) attention

(c) Layer

Figure 5.6: *Causal tracing* analysis of gender signals in LLaMA 30B



(a) feed-forward

(b) attention

(c) Layer

Figure 5.7: *Causal tracing* analysis of gender signals in LLaMA 65B.

Scaling We performed a coarse hyperparameter search for sensitive parameters of DAMA: number of layers and dimensionalities of the projections. Our analysis showed that the algorithm should be applied to the mid-top layers, starting from the 65th percentile to the 93rd percentile of layers ordered from input to output. Also the dimensionality of the projection d_c should be selected based on the dimensionality of the latent representation. We present the best parameters for each of the model sizes in Table 5.3.

Finding 12

Debiasing should be applied to the mid-upper layer of LLaMA models spanning from approximately the 65th to 93rd percentile of layers sorted from input to output.

Finding 13

The dimension of projection used in debiasing adaptation depends on the dimensionality of the latent representation.

We have achieved a notable reduction in bias scores for all models. Noticeably, although we do not observe a common trend for different bias metrics across different model sizes, the improvements brought by DAMA are consistent. Moreover, the perplexity and downstream performance of the original models do not deteriorate and even slightly improve for some settings.

5.3 Conclusions

We have conducted an in-depth causal analysis of the gender signals encoded in the LLaMA models. Causal tracing has revealed that the main culprit of storing bias in the models are the mid-upper feed-forward layers. We also observe that *stereotypical* and *factual* signals are distributed similarly across the model's modules. Interestingly, the last layers tend to counteract the bias impact on the prediction, which was indicated by the low negative value of coefficient a_s in these layers. Further exploration of this phenomenon is an interesting direction for future research.

Based on the results of *causal tracing*, we have introduced a novel method for debiasing language models, DAMA. We show that the method has theoretical guarantees in reducing specific signals encoded in the latent subspace of FF layers. We confirm this observation in practice by applying DAMA to the LLaMA models in the scales from 7B to 65B. The method effectively decreases multiple manifestations of bias (even ones it wasn't explicitly trained on) while preserving the model's high performance on downstream tasks. Furthermore, the method is less computationally demanding than fine-tuning methods. After DAMA optimization, the obtained projection matrix is multiplied together with the feed-forward output matrix. Therefore, the edited model has the same number of parameters and inference complexity as the original LLaMA model. One limitation of our work is that it is data-dependent, and its effectiveness can be affected by the choice of adequate prompts and tokens stimulating the expected model's output (in our case, gender prediction).

As a future work, we plan to apply DAMA to multilingual models to mitigate bias in machine translation (ALMA-R, Xu et al., 2024). It is a crucial and challenging problem because biases as any other societal constructs are heavily dependent on language and culture. Another potential research direction is extending DAMA to other types of bias, e.g., racial or religious, which requires thoughtful design of prompts and potential outputs for model editing.

6

Attention Weights

The motivation for this chapter is to gain a better understanding of syntactic structures emerging in language models without explicit supervision during pre-training. Attention weights were shown to capture linguistic information (Voita et al., 2019; Clark et al., 2019; Vig and Belinkov, 2019). To this end, this chapter (based on publication Limisiewicz et al. (2020)) describes a straightforward algorithm for undercover linguistic signals captured in the attention weight. We show that the attention weights in BERT model (Devlin et al., 2019) convey the representation of syntactic trees even without providing any annotated examples. Furthermore, our method enables mapping syntactic relations whose representation is spread across multiple *attention head*s, heads capturing multiple types of relations, and heads capturing the same type of relation across languages in multilingual BERT.

6.1 Measuring Syntactic Structure in Attention Weights

In Section 3.3.4, we describe parsing as the evaluation of syntactic structures, given a specific annotation style. A 2-dimensional representation for pair of words (as in *attention head*s) enables studying the correspondence between syntactic relations and weights of the matrix. An example of such a method is dependency alignment (DepAl) (Vig and Belinkov, 2019) which sums the attention weights at the positions corresponding to the pairs of tokens forming a dependency edge in the tree.

$$DepAl_A = \frac{\sum_{(i,j)\in\mathcal{E}} A_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{n} A_{i,j}} \tag{6.1}$$

Dependency accuracy (DepAcc) is an alternative metric. For each dependency label it measures how often the governor or dependent key token is the most attended token by the dependent or governor query token (respectively).

$$DepAcc_{l,d,A} = \frac{|\{(i,j) \in \mathcal{E}_{l,d} : j = \arg\max A_{i,\cdot}\}|}{|\mathcal{E}_{l,d}|} \tag{6.2}$$

$\mathcal{E}$ is a set of all dependency tree edges and $\mathcal{E}_{l,d}$ is a subset of the edges with the label l and with direction d, i.e., in dependent-to-governor direction the first element of the tuple i is dependent of the relation and the second element j is the governor; A is a self-attention matrix and $A_{i,\cdot}$ denotes i^{th} row of the matrix; n is the sequence length.

There were several attempts to retrieve the syntactic structure based on the weight of attention layers. For example the works of Clark et al. (2019); Vig and Belinkov (2019) looked into the alignment between the attention weights and the dependency structure, while Kim et al. (2020); Mareček and Rosa (2019) investigated the attention patterns aligned with the constituency phrases.

6.2 Methodology and Experimental Setting

Our analysis aims to uncover the syntactic structure represented in the attention weights of BERT (both English and multilingual models). Following previous works (Voita et al., 2019; Clark et al., 2019; Vig and Belinkov, 2019), we measure the alignment between attention weight matrices and dependency trees. Specifically, we use dependency accuracy (DepAcc) to quantify the alignment.

6.2.1 Dependency Structure and Adaptations

We use dependency annotation for English (Karakanta et al., 2018) in EuroParl multiparallel sentences (Koehn, 2005). In a multilingual setting, we also use EuroParl for other European languages (Czech, French, German, Finnish), Google Universal Dependency Treebank (GSD) for Indonesian, Korean, and Japanese (McDonald et al., 2013); the UD Turkish Treebank (IMST-UD) (Sulubacak et al., 2016).

Since the explicit dependency structure is not used in BERT training, syntactic dependencies captured in latent layers are expected to diverge from the annotation guidelines introduced in universal dependencies. After initial experiments, we have observed that some of the differences are systematic, as shown in Table 6.1. Based on these observations, we modify the universal dependencies annotations in our experiments to better fit the BERT syntax, using UDApi[1] (Popel et al., 2017). All of the modifications are presented with accompanying examples in Table 6.1.

[1] https://udapi.github.io

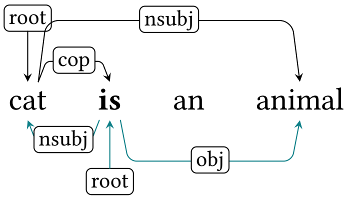
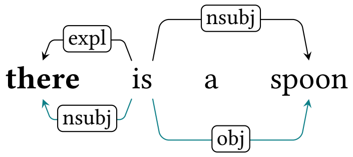
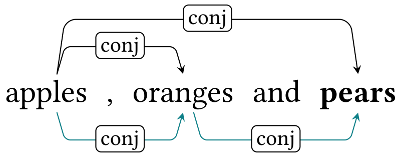
UD	Modified	Example
Copula attaches to a noun	Copula is a root.	 root nsubj cop cat **is** an animal nsubj obj root
Expletive is not a subject	Expletive is treated as a subject	
In multiple coordination, all conjuncts attach to the first conjunct	Conjunct attaches to a previous one	 conj conj apples , oranges and **pears** conj conj

Table 6.1: Comparison of original universal dependencies annotations (**edges above**) and our modification (edges below).

The main motivation of our approach is to get gold-standard trees similar to structures emerging from BERT, which we have observed in qualitative analysis of attention weights. We note that for copulas and coordinations, "BERT's syntax" resembles surface-syntactic universal dependencies (SUD) (Gerdes et al., 2018). Nevertheless, we decided to use our custom modification, since some systematic divergences between SUD and the latent representation occur as well. It is not our intention to compare two annotation guidelines.

Finding 14

We observe that attention weights in BERT align with syntactic structures, yet there are systematic differences between these emergent structures and the gold-standard annotations.

Innovation 5

We propose reversible modifications of gold-standard annotations to make them more similar to patterns emerging in BERT attention weights.

6.2.2 Method: Head Ensembles

We propose a method of ensembling multiple heads of BERT language model (Devlin et al., 2019) by averaging their attention weights. The method contrasts with the previous works that analyzed each head separately. Our objective is to find a set of heads for each directed relation so that their averaged attention weights have a high dependency accuracy. The algorithm can be described in the following steps:

1. We define the maximum number N of heads in the subset;
2. We sort the heads based on their DepAcc on the development set;
3. Starting from the most syntactic one we check whether including the head's attention matrix in the average would increase DepAcc;
4. If the score is improved, the head is added to the ensemble.

When there are already N heads in the ensemble, the newly added head may substitute another added before, so to maximize DepAcc of the averaged attention matrices. In our experiments, we set N to be 4, as allowing larger ensembles does not improve the results significantly.

Innovation 6

We propose a method for selecting and averaging weights of multiple attention heads to increase the alignment with specific syntactic relation types.

6.2.3 Method: Dependency Tree Construction

To extract dependency trees from self-attention weights, we use a method similar to Raganato and Tiedemann (2018), which employs a maximum spanning tree algorithm (Edmonds, 1966). It uses gold information about the root of the syntax tree. We use the following steps to construct a labeled dependency tree:

1. For each non-clausal UD relation label, syntactic heads ensembles are selected as described in the previous method. Attention matrices in the ensembles are averaged. Hence, we obtain two matrices for each label (one for each direction: ”dependent to parent” and ”parent to dependent”).
2. The ”dependent to parent” matrix is transposed and averaged with the ”parent to dependent” matrix. We use a weighted geometric average with weights corresponding to dependency accuracy values for each direction.

3. We compute the final dependency matrix by max-pooling across individual relation-label matrices from step 2. At the same time, we save the syntactic-relation label that was used for each position in the final matrix.

4. In the final matrix, we set the row corresponding to the gold root to zero, to ensure it will be the root in the final tree as well.

5. We use the Chu-Liu-Edmond's algorithm (Edmonds, 1966) to find the maximum spanning tree. For each edge, we assign the label saved in step 3.

It is important to note that the total number of heads used for tree construction can be at most $4 \times 12 \times 2 = 96$, (number of heads per ensemble × number of considered non-clausal labels × two directions). However, the number of used heads is typically much lower (see Table 6.3). That means our method uses at most 96 integer parameters (indices of the selected heads). It is considerably less than projection layers in fine-tuning or probing (described in Section 4.1), which consist of thousands of real number parameters. In the method, we only utilize the ensembles for non-clausal relations, as dependency accuracy for clausal ones was relatively low. Nevertheless, in the evaluation of obtained parses, we consider all the relations.

Innovation 7

We propose an effective algorithm enabling extracting labeled trees from attention weights with minimal supervision.

6.3 Results

We describe the results of our methods in two steps. In Section 6.3.1, we evaluate *head ensembles* based on their dependency accuracy. Subsequently in Section 6.3.2, we investigate the utility of *head ensembles* by evaluating syntactic trees extracted from them.

6.3.1 Head Ensembles

In Table 6.2, we present results for the dependency accuracy of a single-head and four-head ensemble. We compare them with the positional baseline. This baseline looks at the most frequent relative position for each dependency label. Noticeably, a single *attention head* surpasses the baseline for every relation label in at least one direction. The average of 4 heads surpasses the baseline by more than 10% for every relation.

Relation	Base-	1 Head		4 Heads	
	line	d2p	p2d	d2p	p2d
label	line	d2p	p2d	d2p	p2d
amod	78.3	90.6	77.5	**93.8**	79.5
advmod	48.7	53.3	62.0	62.1	**63.6**
aux	69.2	90.9	86.9	**94.5**	88.0
case	36.4	83.0	67.1	**88.4**	68.9
compound	75.8	83.2	75.8	**87.0**	79.1
conjunct	31.7	47.4	41.6	**58.8**	51.3
det	56.5	95.2	62.3	**97.2**	69.4
nmod	25.4	34.3	41.5	49.1	**54.7**
nummod	57.9	75.9	64.6	**79.3**	72.6
mark	53.7	66.2	54.7	**73.5**	65.9
obj	39.2	84.9	68.6	**89.3**	78.5
nsubj	45.8	56.2	62.7	57.8	**76.0**
⇑ AVG. NON-CLAUSAL	52.8	67.8		**74.1**	
acl	27.9	41.5	36.5	**50.5**	43.8
advcl	9.3	26.3	26.7	**40.7**	26.3
csubj	20.0	20.7	**31.0**	24.1	**31.0**
x/ccomp	34.8	60.4	47.9	**66.9**	52.1
parataxis	10.4	17.6	12.1	23.1	**24.2**
⇑ AVG. CLAUSAL	20.5	32.1		**38.3**	
punct	9.4	21.1	40.3	28.4	**44.0**
dep	18.8	21.6	33.1	25.1	**37.0**

Table 6.2: Dependency accuracy for single heads, 4 heads ensembles, and positional baselines. The evaluation was done using the pre-trained model BERT. The positional baseline looks at the most frequent relative position for each dependency label (Voita et al., 2019). The names of the relations are abbreviations used in universal dependencies. We group some of them and present the average score: *obj*: objects also include indirect objects (*iobj*), *x/ccomp*: open clausal complements and clausal complements; *dep*: *dep* relations and all remaining relations not included in this table. The relations are aggregated into two groups: clausal (relations connecting tokens across clauses) and non-clausal (spanning inside a single clause).

Setting	Use labels	Model	Selection sentences	Heads per ensemble	Heads used	UAS	LAS
Left branching baseline	—	—	—	—	—	11.0	—
Right branching baseline	—	—	—	—	—	35.5	—
Raganato+ (paper)	no	NMT	1000*	—	1	38.9	—
Raganato+	no	BERT	1000*	—	1	37.2	—
Our method (ablation)	no	BERT	1000	1	2	36.0	—
	yes	BERT	1000	1	15	37.4	9.5
	yes	BERT	20	4	36	43.6	14.5
	no	BERT	1000	4	8	51.2	—
Our method (best)	yes	BERT	1000	4	48	**52.0**	**21.7**

Table 6.3: Parsing evaluation for different settings of dependency tree extraction. For a fair comparison with previous methods, we consider all types of dependency relation (including clausal) and do not apply our modifications of UD trees presented in Table 6.1. In Raganato and Tiedemann (2018) experiments, the trees were induced from each encoder head, but we report only the results for the head with the highest UAS on test set.

Ensembling brings the most considerable improvement for nominal subjects (p2d: +13.3%) and noun modifiers (p2d: +13.2%). The relative change of accuracy is more evident for clausal relations than non-clausal.[2] Dependent-to-parent direction has higher accuracy for modifiers (except adverbial modifiers), functional relations, and objects. Whereas parent-to-dependent favors other nominal relations (nominal subject and nominal modifiers).

6.3.2 Dependency Trees

In Table 6.3, we report the evaluation results on the English PUD treebank using unlabeled attachment score (UAS) and labeled attachment score (LAS). For comparison, we also include the left- and right-branching baseline with gold root information. Moreover, we compare to the best-performing head found by Raganato and Tiedemann (2018) They used a Transformer model trained for machine translation and extracted whole trees from a single *attention head* and did not average directions. The results show that ensembling multiple *attention head*s for each relation label allows the construction of better trees than the single-head approach.

The number of unique heads used in the process turned out to be two times lower than the maximal possible number (96). This is because many heads appear in multiple ensembles. We examine the multipurpose heads (i.e. shared across multiple ensembles) in Section 6.4.2.

[2] Clausal relations connect tokens across clauses, while non-clausal span inside a single clause.

Lang-	Features	DepAcc		UAS		LAS
uage		b-line	Our	b-line	Our	Our
EN	SVO, AN	52.8	**73.2**	35.5	**51.0**	21.8
DE	—*, AN	42.3	**72.9**	32.9	**45.5**	19.5
FR	SVO, NA	50.6	**72.8**	34.7	**48.3**	18.0
CS	SVO, AN	44.3	**69.7**	34.0	**40.1**	17.1
FI	SVO, AN	55.6	**77.0**	35.5	**45.8**	15.9
ID	SVO, NA	47.0	**64.2**	29.7	**36.9**	14.6
TR	SOV, AN	60.0	**68.0**	**38.8**	29.3	7.9
KO	SOV, AN	**41.8**	32.4	**49.3**	28.8	8.0
JA	SOV, AN	56.9	**69.5**	35.9	**39.0**	14.3
Mean SVO		50.1	**71.4**	33.9	**44.4**	17.5
Mean SOV		52.8	**56.7**	**34.1**	32.4	13.9
Mean AN		50.6	**66.1**	34.3	**39.9**	16.6
Mean NA		48.8	**68.5**	32.2	**42.6**	16.3

Table 6.4: Average dependency accuracy for non-clausal relations (with UD modification) compared with positional baseline. UAS, LAS of constructed trees (without UD modification) compared with UAS of left or right-branching trees with gold root, whichever is higher. mBERT was used for all languages. *: German does not have a dominant order.

Furthermore, to the best of our knowledge, we are the first to produce labeled trees from *attention head*s and report both UAS and LAS. For reference, the unsupervised parser of Han et al. (2019) obtains 61.4% UAS. However, the results are not fully comparable since their parser uses information about gold POS tags, and the results were measured on different evaluation data (WSJ Treebank).

6.3.3 Ablation

We analyze how much the particular steps of our tree extraction method influenced the quality of constructed trees. We also repeat the experimental setting proposed by Raganato and Tiedemann (2018) (i.e. extracting trees from a single head) on the enBERT model to see whether a language model is better suited to capture syntax than a translation system. Additionally, we alter the procedure described in Section 6.2.3 to analyze which decision influenced our results the most, i.e., we change:

1. Size of *head ensemble*s.

2. Number of sentences used for head selection.

3. Use the same *head ensemble* for all relation labels in each direction. Hence we do not use synatactic labels and skip max-pooling described in Subection 6.2.3, point 3.

In Table 6.3, we see that analyzing a single head in BERT (as Raganato and Tiedemann (2018)) produces slightly worse trees than the same method applied to neural machine translation. If we do not use ensembles and only one head per each relation label and direction is used, our pipeline offers only a 0.2% rise in UAS and poor LAS.

The introduction of *head ensemble*s of size four has brought the most significant improvement in our method of tree construction, which is roughly +15% for both variants (with and without labels). Together with the findings of head ensembling, this supports our claim that syntactic information is spread across many heads. Interestingly, max-pooling over attention weight matrices improves UAS only by 0.8%. Nevertheless, this step is necessary to construct labeled trees. The performance is competitive, even with as little as 20 sentences used for head selection.

6.3.4 Multilingual Model

In table 6.4 we present the results of our methods applied to mBERT and evaluated on Parallel Universal Dependencies in nine languages. Comparison of the results for English with table 6.3 shows that the dependency accuracy and UAS decreased only slightly by changing the model from enBERT to mBERT, while LAS saw a 0.1% increase. The model captures syntax comparably well in German, French, and Finnish.

We observe that results for languages following subject-object-verb (SOV) order (Turkish, Korean, Japanese) are significantly lower than for subject-verb-object (SVO) languages (English, French, Czech, Finnish, Indonesian) in both dependency accuracy (14.7%) and the UAS (10.5%). Our methods outperform the baselines in the latter group by 17.2% to 25.4% for dependency accuracy and from 6.1% to 15.5% for UAS. The influence of Adjective and Noun order is less apparent. On average, the NounAdj order languages results are higher than for the AdjNoun languages by 2.4% in dependency accuracy and 2.7% in UAS.

The disparity in the results for SVO and SOV languages was previously observed by Pires et al. (2019), who fine-tuned mBERT for part of speech tagging and evaluated zero-shot accuracy across typologically diverse languages. We hypothesize that worse performance for SOV languages may be due to their lower prevalence in mBERT's pre-training corpus. In the following section, we provide further analysis of emerging multilingual patterns in specific heads.

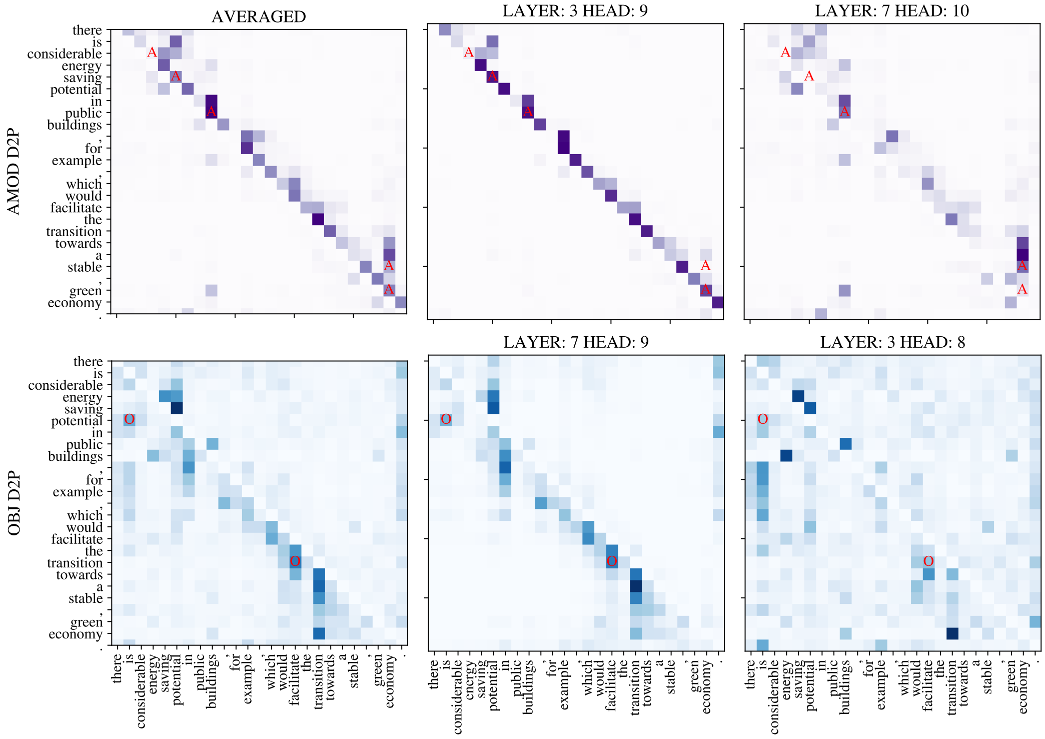


Figure 6.1: Examples of two enBERT's attention heads covering the same relation label and their average. Gold relations are marked by red letters.

6.4 Analysis of UD in BERT: Both More Specific and More General

In this section, we analyze the correspondence between *attention heads* and relation types in the dependency structures. We present the most interesting finding of our work, specifically that patterns emerging in attention can be both more granular (check Section 6.4.1) or more general (see Section 6.4.2) than syntactic relation types in UD. Additionally in Section 6.4.3, we show that the heads of multilingual BERT can capture similar relation types across languages.

6.4.1 One Relation in Many Heads

We observe that a single head often captures only a specific aspect or subtype of one UD relation type. Therefore in our approach, we averaged multiple heads to cover different manifestations of one dependency relation type.

In Figure 6.1, we show examples of heads capturing the same type of syntactic relation in English. The first column shows the average weights of the heads which offer noticeably better alignment with the relation than single heads. In the top row (purple), both heads identify the parent noun for an adjectival modifier: Head 9 in

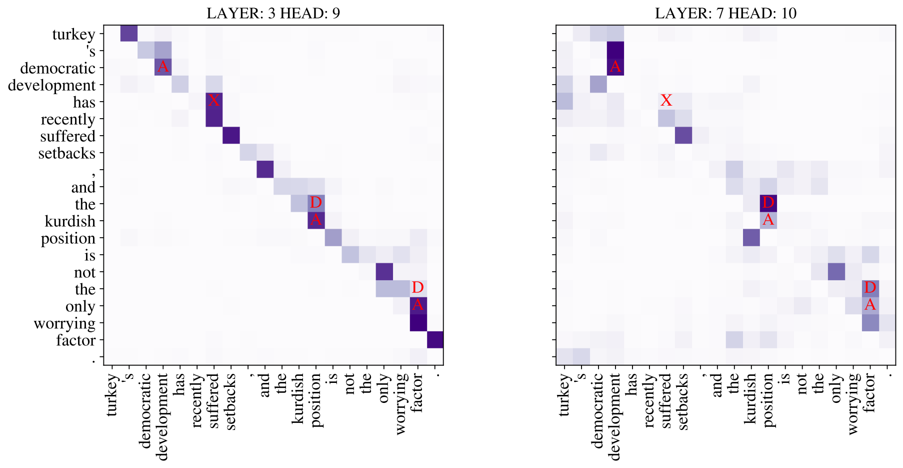


Figure 6.2: Syntactic BERT heads retrieving the parent for three relation labels: **A**djective modifiers, Au**X**iliaries, **D**eterminers. UD relations are marked by A, X, and D respectively.

Layer 3 if their distance is two positions or less, Head 10 in Layer 7 if they are further away (as in "a stable, green economy"). Similarly, for an object to predicate relation (blue bottom row), Head 9 in Layer 7 and Head 8 in Layer 3 capture pairs with shorter and longer positional distances, respectively.

Finding 15

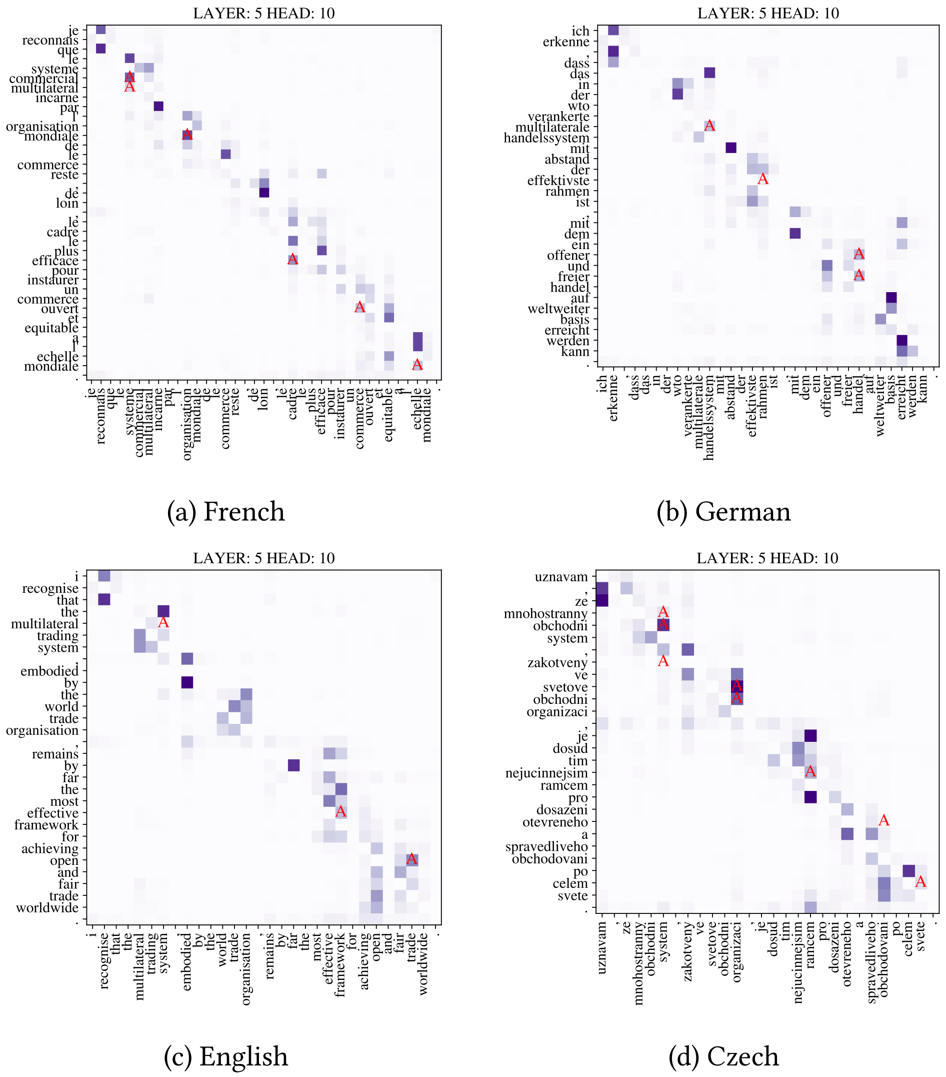
Syntactic relations are typically encoded by multiple heads, to increase the alignment the weights of those heads should be averaged.

6.4.2 Many Relations in One Head

In Figure 6.2, we visualize a few examples of heads whose weights align with multiple types of syntactic relations. We show that those shared relations typically span within one type of constituency phrase. For instance, specific heads (e.g., 9th in layer 3 and 10th in layer 7) find both article-to-noun and adjective-to-noun relations. This observation supports the previous findings that the *attention heads* contextualize information inside the constituency phrases (Mareček and Rosa, 2019).

Finding 16

Some attention heads encode multiple different syntactic relation types, that have similar linguistic purposes.



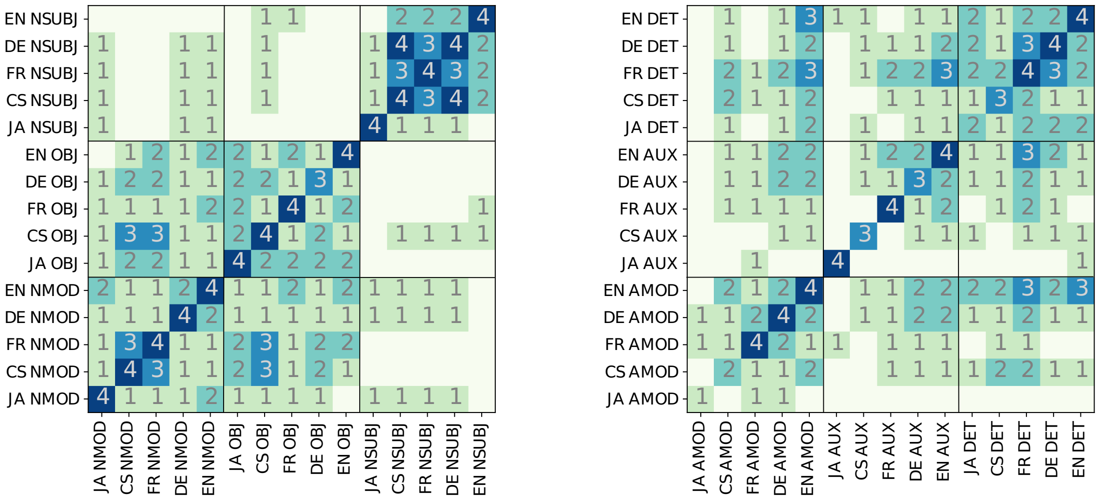
(a) French

(b) German

(c) English

(d) Czech

Figure 6.3: A single mBERT head which identifies noun heads of French adjective modifiers. It also partially captures the relation in German, English, and Czech, although these languages, unlike French, follow “Adjective Noun” order.



(a) Nominal relations P2D

(b) Modifiers D2P

Figure 6.4: Number of mBERT's heads shared across *head ensembles*, both within and across languages.

6.4.3 Multilingual Heads

In Figure 6.3, we show that specific heads can capture the same type of syntactic relation also across languages. Figure 6.4 presents the sizes of intersections between *head ensembles* for different languages and dependency labels. Except for Japanese, we observe an overlap of the heads pointing to the governor of adjective modifiers, auxiliaries, and determiners. Shared heads tend to find the root of the syntactic phrase. Interestingly, common heads occur even for relations typically spanning within verb and noun phrases, such as auxiliaries and adjective modifiers. Nevertheless, we have not noticed that these heads would focus attention on any particular part of speech tokens. Similarly, objects and noun modifiers share at least one head for all languages. They have a similar function in a sentence, with a distinction that objects are dependents of predicates, while noun modifiers are dependents of nouns.

Finding 17

In multilingual BERT, weights in some attention heads align with similar syntactic relation types in multiple languages.

6.5 Conclusions

In this chapter, we have confirmed and extended the observations of correspondence between attention weights and syntactic relations. We found out that the emerging patterns correlate with the UD relations, yet there are also systematic differences. Notably, we observe that there is no one-to-one correspondence between relations and heads: some heads capture multiple types of relations, while some relations are captured by multiple heads. Moreover, in a multilingual setting, we identify heads capturing the same type of relations across diverse languages.

Finally, we show the practical implications of this work. One contribution is the new method of *head ensemble* identification, which enables the construction of dependency trees from *attention head*s, selected based on minimal supervision. Furthermore, we note that the structures emerging in *attention head*s may be an inspiration for modifying existing annotation guidelines.

Admittedly, finding a syntactic head requires small supervision for better performance. Therefore, the method is still dependent on data, albeit to a lesser extent than previously described *probing*. The extracted trees exhibit lower quality than the ones obtained with the supervised parser, although the latter requires more annotated examples.

7 Input Embeddings

In this chapter, we turn our focus on the input embedding layers of the language models. The input embeddings were often overlooked in the analyses of the language models, yet in many models, they make up a large portion of the model's parameters, e.g. 192M out of 270M parameters in XLM-RoBERTa$_{\text{Base}}$(Conneau et al., 2020a) (70%).

The majority of recently deployed NLP models use subword tokenization as the method of representing input and output sequences in a numerical way as embeddings (Mielke et al., 2021). Therefore to study the properties of the input embeddings, we need to understand how they are allocated to input and output sequences by subword tokenizers, and especially how the subword vocabulary is constructed. This chapter presents the results of the work Limisiewicz et al. (2023a) and unlike previous chapters, it is solely devoted to the multilingual setting.

7.1 Subword Vocabulary and Tokenization

Text segmentation is an important step in the Transformer, as it maps discrete units: word, character, or even bytes to continuous representation, i.e. word embedddings. Recently the most dominant approach has been to split text into subword tokens. Tokens are selected based on their frequency, enabling representing frequent words as single tokens, while rare words are split into more subwords, which in the edge case can be single characters or bytes. The sets of subwords are learned from the training data with self-supervised vocabulary construction algorithms, e.g. Byte Pair Encoding (BPE, Sennrich et al., 2016) or Unigram (Kudo, 2018). During model training and inference, texts are tokenized into subwords, i.e. encoded left-to-right as subwords (Devlin et al., 2019; Song et al., 2021) or by choosing the most probable subword segmentation for each word as in SentencePiece (Kudo and Richardson, 2018). While subword tokenization has significant benefits over traditional rule-based tokenization, e.g., by better treating of Out-of-vocabulary (OOV) words. It has been observed that the choice of subword vocabulary has a significant impact on the performance of the models.

We will discuss what subword vocabularies are adequate to process multilingual texts and allow to encoding of necessary lexical information in the embedding representation. We focus on the characteristics of subword tokenization methods in a multilingual setting. To this length, we introduce methods for measuring whether tokenizers effectively represent meaningful language-specific tokens in the vocabulary (*vocabulary allocation*) and whether the units they learn are shared across languages (*vocabulary overlap*). Our approach aims to answer: how do sub-word tokenizers differ in *vocabulary overlap* and *vocabulary allocation* of learned vocabularies? and which properties of multilingual tokenizers affect the LM's representation quality?

7.2 Methodology and Experimental Setting

We train a set of multilingual RoBERTa-like models with different methods for vocabulary construction and subword tokenization. We train four tokenizers for a set of diverse 6 languages (English, Spanish, Turkish, Greek, Chinese, Arabic) using existing methods: Unigram, BPE, and our methods for monolingual tokenizer merging: NoOverlap, TokMix (described in the subsequent section). We always set the size of the vocabulary to $V = 120,000$ tokens. Using these tokenizers, we then train four models following the settings of XLM-R (Conneau et al., 2020a) which we then use for the probing experiments.

Subsequently, we repeat the analysis for the broader set of 20 diverse languages (including six mentioned earlier and: Hebrew, Georgian, Urdu, Hindi, Marathi, Thai, Tamil, Telugu, Bulgarian, Russian, Swahili, Vietnamese, French, German) with three tokenization methods used in three pre-trained models. In this setting, we do not use NoOverlap tokenizer, which cannot be trained effectively given the chosen size of the vocabulary.

7.2.1 Merging of Monolingual Tokenizers

Due to a significant imbalance of the data sizes for different languages in multilingual corpora, the multilingual tokenizers tend to allocate a vast majority of vocabulary units to the most frequent languages hindering models' performance on low-resource languages (Rust et al., 2021). To alleviate this issue, we suggest utilizing monolingual tokenizers for multilingual tokenization. First, the Unigram LM tokenizers are trained on separate monolingual corpora. The tokenizers are then combined to create a tokenizer suitable for multilingual data. We propose two methods for combining monolingual tokenizers:

Language-specific Tokenization NoOverlap: We train Unigram tokenizers for each of L considered languages with the same vocabulary size for each of the languages $\frac{V}{L}$. In multilingual tokenization, we apply the tokenizer for a specific language separately and produce a token with language identification.[1] The vocabulary consists of L segments of total size V. Naturally, the tokenized texts in different languages will consist of tokens from distinct vocabulary segments. Noticeably, the same character sequence in different languages can be assigned different token identities.

Language-Mixed Tokenization TokMix: We train Unigram LM tokenizers for each of L languages. Subsequently, we averaged vocabulary unit probabilities across tokenizers, sorted them, and trimmed the vocabulary to the pre-set vocabulary size V keeping the units with the highest probability.[2]

$$\hat{\theta} = \sum_{i=1}^{L} w_i \theta_i \tag{7.1}$$

w_i are weights assigned to each language. By default, we set the weights to be uniform and equal to $\frac{1}{L}$. Unlike NoOverlap, the same vocabulary units coming from distinct monolingual tokenizers are merged into one unit with averaged probability.

7.2.2 Tokenizer and Model Training Setting

We download 10% of CommonCrwal corpus available atv `https://data.statmt.org/cc-100/`. Following the methodology Conneau and Lample (2019b), we subsample each language's data to ensure that the training corpus is well-balanced across languages. An equation defines the sample size c_l for language l:

$$c_{l,\alpha} = c_{\min} \cdot \left(\frac{|C_l|}{c_{\min}} \right)^{\alpha} \tag{7.2}$$

[1] Only the special tokens are shared across languages, e.g., "<s>" – the beginning of a sentence token.

[2] To account for possible overlaps between language-specific vocabularies, we set their sizes above $\frac{V}{L}$. It assures that joint vocabulary will have at least V tokens.

Where $c_{\min}$ is the minimal sample size (defined by the smallest language), and C_l is all data available for a language, α is the so-called *balancing parameter*. In our experiments, we set $c_{\min}$ to 10 M characters, C_l is, e.g., 8.8 B characters for English. We set α to 0.25, which corresponds to a balancing factor picked in XLM-R. The training data for the tokenizer and the model are the same. The vocabulary size V was set to 120,000.

7.2.3 Quantifying Tokenizer Properties

First, we introduce an analytical approach to evaluate different aspects of multilingual tokenization. The measures are non-parametric and describe the key properties of multilingual tokenizers: quality of vocabulary representation for particular languages and lexical overlap across languages.

We base our analysis on the empirical probability distribution of vocabulary units $v \in \mathcal{V}$ computed on training corpus for each language l:

$$d_{l,\mathcal{V}}(v) = \frac{f(v, C_l)}{\sum_{v \in \mathcal{V}} f(v, C_l)} \tag{7.3}$$

Function $f(v, C_l)$ is the number of occurrences of a vocabulary unit v in monolingual training corpus C_l.

We aim to quantify how well multilingual vocabulary represents meaningful lexical units of particular languages. Our intuition is that a good lexical representation is obtained when: 1. It uses a vast portion of multilingual vocabulary, and thus a larger part of the embedding layer is devoted to the language; 2. The text in the language is split into longer and potentially more meaningful tokens.

***Vocabulary allocation*: Average Rank** To measure the number of vocabulary units available for modeling specific languages, we propose an estimation of the average rank (AR) of vocabulary units in distribution over a monolingual corpus.[3] This measure denotes how many tokens are typically considered by a language model that has access to language identity information but no context (probabilistic unigram model).

$$\mathrm{AR}_{l,\mathcal{V}} = \sum_{v \in \mathcal{V}} \mathrm{rank}(v, d_{l,\mathcal{V}}) d_{l,\mathcal{V}}(v) \tag{7.4}$$

[3] In this context, rank is the position of unit v in the vocabulary $\mathcal{V}$ sorted in descending order by the probability distribution $d_{l,\mathcal{V}}$

Our intuition is that the model will have better information about the language's lexicon when vocabulary is distributed over a larger number of tokens as more parameters of the input embedding layer would be allocated to represent language-specific features. Moreover, larger vocabularies tend to cover longer and more meaningful units.

***Vocabulary overlap*: Character per Token** In line with previous intuition, longer tokens should have a more meaningful representation. Therefore, we measure text fragmentation by computing characters per token (CPT) the average number of characters for a vocabulary unit in monolingual corpus C_l.:

$$\mathrm{CPT}_{l,\mathcal{V}} = \frac{|C_l|}{|T_{\mathcal{V}}(C_l)|} \tag{7.5}$$

$T_{\mathcal{V}}(C_l)$ is the tokenization of the corpus with vocabulary $\mathcal{V}$; $|C_l|$ is the size of the corpus measured as the number of characters. We choose the number of characters as a base unit because it's not susceptible to cross-lingual differences regarding word boundaries and the average length of words. Still, the amount of information conveyed by a single character varies largely with the writing systems, e.g., texts written in logographic scripts (e.g., Chinese, Japanese) tend to be shorter in the number of characters than similarly informative texts in the alphabetic script (e.g., Latin, Perfetti and Liu, 2005).

Vocabulary Overlap Another important property of multilingual vocabulary is sharing lexical units across languages. Previous works claimed that vocabulary overlap improves cross-lingual transfer for learning downstream tasks (Pires et al., 2019; Wu and Dredze, 2019). We measure overlap as the divergence between corpora distributions d_l (defined in equation 7.2.3). We use the Jensen-Shanon divergence (JSD).[4], because it is symmetric and applicable for distribution with different supports. The latter is often the case when distributions are estimated for languages with distinct writing systems.

$$\mathrm{JSD}(d_{l1,\mathcal{V}}||d_{l2,\mathcal{V}}) = \frac{1}{2}\sum_{v\in\mathcal{V}} d_{l1,\mathcal{V}}(v)\log_2\frac{d_{l1,\mathcal{V}}(v)}{m_{l1,l2,\mathcal{V}}(v)} + \frac{1}{2}\sum_{v\in\mathcal{V}} d_{l2,\mathcal{V}}(v)\log_2\frac{d_{l2,\mathcal{V}}(v)}{m_{l1,l2,\mathcal{V}}(v)} \tag{7.6}$$

where:

$$m_{l1,l2,\mathcal{V}} = \frac{1}{2}d_{l1,\mathcal{V}} + \frac{1}{2}d_{l2,\mathcal{V}} \tag{7.7}$$

[4] In NLP literature, JSD is also known as "information radius" (Manning and Schütze, 2001).

JSD is bounded in the range 0 to 1. The lower the value, the larger the overlap across corpora. Another possibility to quantify overlap is to count unique vocabulary units appearing in tokenized texts across languages. The advantage of divergence is that it reflects the frequency of shared tokens across corpora. It is also less affected by the choice of the data size used for estimating empirical probability distributions (d_l).

Innovation 8

We introduce benchmarks for evaluating tokenizer properties: *vocabulary allocation* and *vocabulary overlap*. These benchmarks are easy to compute and do not require costly model training.

7.2.4 Impact on Language Modeling

In this section, we present the tasks and measures for the evaluation of multilingual language models trained with different tokenizers.

Intristic Evaluation We evaluate the masked language modeling performance with mean reciprocal rank (MRR) (defined in Section 3.2) on the test set from CommonCrawl corpus.

Probing for End-tasks To measure the models' ability to encode linguistic information, we probe them for a set of downstream classification tasks (for an extended description of probing see Section 4.1). We group them based on the granularity of examined representations: word-level and sentence-level tasks.

Word-level Tasks We test syntactic tasks: part of speech and dependency labeling on universal dependencies de Marneffe et al. (2021) and named entity recognition on Wikiann dataset (Pan et al., 2017).

Sentence-level Tasks In this set of tasks, we examine whether the model learns sentence-level representations that capture its semantics and can be transferred across languages. To obtain this sentence embedding, we average the model's output representation across all the tokens in the sentence. We evaluate cross-lingual natural language inference dataset (Conneau et al., 2018b) and cross-lingual sentence retrieval on Tatoeba bitext corpus (Artetxe and Schwenk, 2019). Unlike previous tasks, sentence retrieval is solved by an unsupervised algorithm matching sentences based on their cosine similarity.

Testing In-language vs. Cross-lingual Transfer For all the downstream tasks, except sentence retrieval, we compute in-language performance by training the probe and evaluating it on held-out test data in the same language. We quantify cross-lingual transfer by training a probe on one language (source) and evaluating it on the test set for another language (target).

7.3 Results

We present the results of the experiments in two sections: the evaluation of tokenizers' properties and their impact on the performance of language models.

7.3.1 Evaluation of Tokenizers' Properties

***Vocabulary allocation* largely varies throughout languages and tokenization methods.** Table 7.1 shows that the average rank noticeably differs across languages. The highest AR is observed for Chinese, which is explained by the usage of logographic scripts, which require an extensive vocabulary capacity to encode all characters.

		AR	TR	ZH	EL	ES	EN
AR	Unigram	2129	2719	**5919**	2070	1439	1513
	BPE	2972	3226	4294	**2907**	**2220**	**2143**
	NoOverlap	2537	2653	2090	2065	1661	1597
	TokMix	**3485**	**4167**	3961	2639	1999	1898
CPT	Unigram	3.16	4.01	1.84	3.5	3.88	3.91
	BPE	**3.7**	4.19	**2.03**	**3.97**	**4.34**	**4.22**
	NoOverlap	3.53	4.19	1.56	3.81	4.15	4.15
	TokMix	**3.7**	**4.45**	1.73	3.9	4.24	4.18

Table 7.1: Values of *vocabulary allocation* measures for 4 tokenizers trained on the small language set. The highest values for each language are bolded.

Multilingual *vocabulary allocation* is highly dependent on the tokenization method used. Vocabulary learned with Unigram underperforms BPE and TokMix in both average rank and characters per token. This trend holds throughout languages except for Chinese. The observation suggests that our vanilla Unigram is a suboptimal multilingual vocabulary learner.

It is important to note that NoOverlap scores lower than Unigram in the *vocabulary allocation* measures due to the limited vocabulary size for each language caused by prohibiting overlap. However, as shown in the next section, LM trained with this tokenizer can achieve good results on some tasks.

The choice of tokenization method affects *vocabulary overlap*. Figure 7.1 shows Jensen-Shanon divergence values between the vocabularies of six languages. We observe that the highest cross-lingual overlaps appear in the vocabulary obtained by Unigram, followed by TokMix, and BPE. Expectedly, we do not observe overlaps

Metric	Tokenizer	Different script	Same script	All transfers
Overlap	Unigram	**0.77**	**0.62**	**0.74**
(JSD)	BPE	0.83	0.68	0.8
	NoOverlap	1.0	1.0	1.0
	TokMix	0.8	0.65	0.77
NER	Unigram	31.3 ±0.4	55.4 ±0.2	36.1 ±0.4
(F1)	BPE	**33.5** ±0.5	**59.9**±0.2	**38.7** ±0.4
	NoOverlap	32.0 ±0.5	48.6 ±0.4	35.3 ±0.5
	TokMix	31.8 ±0.4	58.0 ±0.3	37.0 ±0.4
POS	Unigram	18.1 ±0.4	38.3 ±0.4	22.2 ±0.4
(F1)	BPE	**25.8** ±0.5	**40.8** ±0.4	**28.8** ±0.5
	NoOverlap	20.1 ±0.5	**41.9** ±0.5	24.5 ±0.5
	TokMix	21.9 ±0.4	40.4 ±0.3	25.6 ±0.4
Dep. labeling	Unigram	11.1 ±0.3	25.5 ±0.3	14.0 ±0.3
(F1)	BPE	**15.9** ±0.4	**27.0** ±0.4	**18.1** ±0.4
	NoOverlap	12.8 ±0.4	**27.8** ±0.5	15.8 ±0.4
	TokMix	12.6 ±0.5	26.1 ±0.3	15.3 ±0.5
NLI	Unigram	**42.2** ±0.7	**43.7** ±0.7	**42.5** ±0.7
(Acc)	BPE	**42.4** ±0.7	**45.2** ±0.8	**43.0**±0.7
	NoOverlap	37.3 ±0.6	37.1 ±0.5	37.2 ±0.6
	TokMix	**41.2** ±0.7	**42.7** ±0.5	**41.5** ±0.7
Retrieval	Unigram	21.0	**43.9**	25.6
(Acc)	BPE	20.9	40.7	24.9
	NoOverlap	12.3	28.0	15.4
	TokMix	**23.0**	43.4	**27.1**

(a) 6 languages

Tokenizer	Different script	Same script	All transf
Unigram	**0.75**	**0.58**	**0.73**
BPE	0.83	0.67	0.81
TokMix	0.8	0.64	0.78
Unigram	33.2 ±0.5	50.7 ±0.6	35.4 ±0.5
BPE	**36.6** ±0.6	**54.3** ±0.3	**38.8** ±0.5
TokMix	**36.5** ±0.6	**53.7** ±0.5	**38.7** ±0.6
Unigram	23.4 ±0.5	32.9 ±0.3	24.6 ±0.5
BPE	**30.5** ±0.6	**40.7** ±0.4	**31.8** ±0.6
TokMix	**29.2** ±0.5	**40.4** ±0.3	**30.7** ±0.5
Unigram	13.0 ±0.6	15.6 ±0.5	13.4 ±0.6
BPE	**16.5** ±0.6	**19.2** ±0.5	**16.9** ±0.5
TokMix	**16.0** ±0.5	**19.4** ±0.4	**16.5** ±0.5
Unigram	**37.3** ±0.5	37.5 ±0.4	37.4 ±0.5
BPE	36.2 ±0.5	**38.7** ±0.5	**36.7** ±0.5
TokMix	**37.8** ±0.5	**39.2** ±0.5	**38.1** ±0.5
Unigram	**44.1**	44.4	44.2
BPE	**44.1**	**49.1**	**45.1**
TokMix	42.8	46.9	43.6

(b) 20 languages

Table 7.2: Averaged results of the evaluation for cross-language overlaps and transfers. Each probing result is an average of 5 random seeds (for 6 languages) and 3 random seeds (for 20 languages). The best value in each metric is underlined, and bolded results are closer than the sum of standard deviations from the optimal value.

for NoOverlap's setting (JSD $= 1$). Jensen-Shanon divergence divergence is a good predictor of whether the languages share the script. For all tokenization methods, the divergence is significantly smaller in the bottom-right square grouping of the languages using Latin script.

Finding 18

Popular tokenization methods produce vocabularies with significantly different properties: Unigram is characterized by higher overlap and lower allocation than BPE.

Innovation 9

Our novel tokenization method (TokMix) offers an increase in vocabulary allocation while keeping high overlap across languages in comparison to a similar Unigram tokenizer.

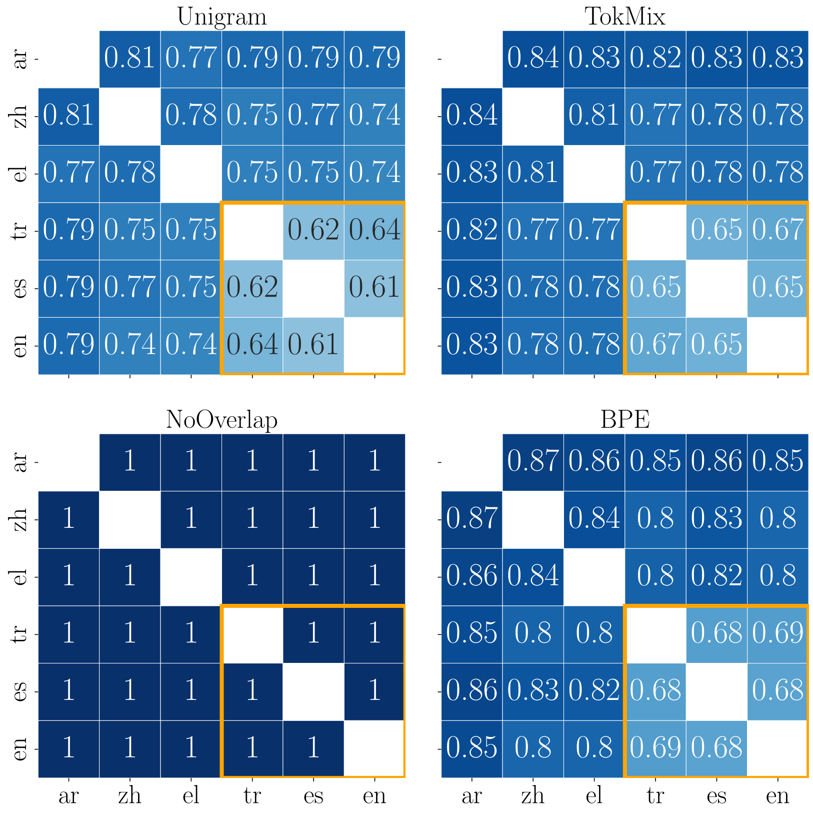
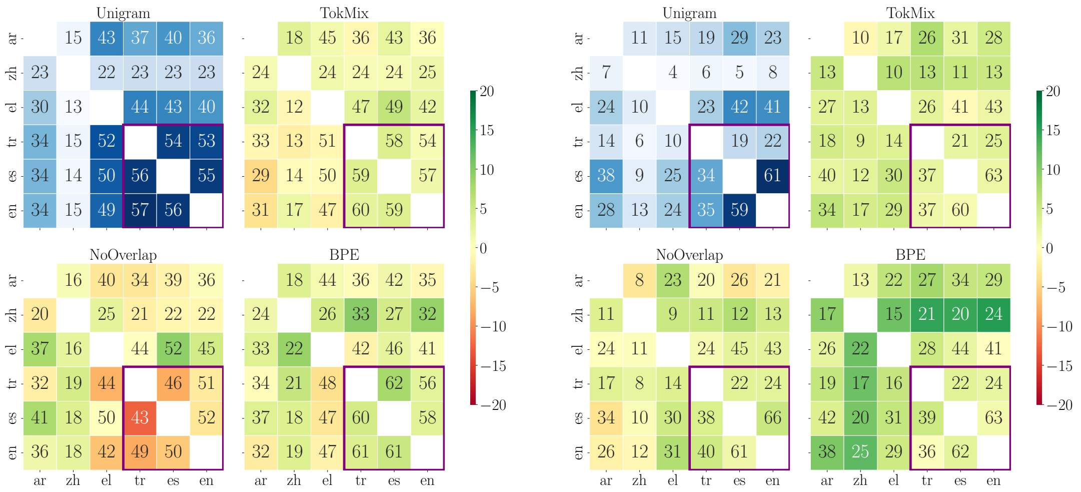


Figure 7.1: *Vocabulary overlap* measure: Jensen-Shanon divergence for four tokenization methods. The orange square in the bottom right groups the languages with the same script (Latin).

7.3.2 Tokenizer Properties Impact LM Performance

High *vocabulary allocation* improves downstream results for word-level tasks. In Table 7.3a, we observe that the choice of the tokenization method significantly impacts the results for POS, dependency labeling, and NER. We presume it results from learning good lexical representations throughout languages, e.g., by BPE and TokMix. The higher *vocabulary allocation* is especially beneficial for word-level tasks, whereas the influence on the sentence-level task NLI is minimal.

Notably, the model instance with NoOverlap tokenizer achieves the best F1 in POS and dependency labeling despite underperforming in *vocabulary allocation*. It is the result of learning language-specific representation for each token, which is especially useful for syntactic tasks.



(a) NER (F1)

(b) POS (F1)

Figure 7.2: Cross-lingual transfer for POS and NER tasks, the probes are trained on data for languages on the y-axis and evaluated for languages on the x-axis. The absolute values are presented for the Unigram tokenizer. For other tokenization methods, the color scheme shows a difference from the Unigram algorithm. In the case of named entity recognition, we observe a drop in cross-lingual transfer for NoOverlap tokenization, especially for the same script pairs It suggests that lexical overlap is an important aspect contributing to cross-lingual transfer for NER. We don't see similar drop in the case of part of speech tagging.

	V. Allocation		**MLM**	**NER**	**POS**	**Dep. labeling**	**NLI**
	(AR)	(CPT)	(MRR)	(F1)	(F1)	(F1)	(Acc)
Unigram	2042	3.17	42.0	62.8 ±0.1	57.1 ±0.2	48.1 ±0.4	**53.4** ±0.5
BPE	2193	**4.47**	35.6	**70.4** ±0.1	**68.9** ±0.2	**58.7** ±0.4	**53.3** ±0.3
NoOverlap	1829	3.16	**42.7**	69.4 ±0.1	**69.2** ±0.2	**58.8** ±0.3	**53.0** ±0.4
TokMix	**2198**	3.34	38.7	**70.2** ±0.1	67.3 ±0.1	57.3 ±0.4	**53.3** ±0.4

(a) 6 languages

	V. Allocation		**MLM**	**NER**	**POS**	**Dep. labeling**	**NLI**
	(AR)	(CPT)	(MRR)	(F1)	(F1)	(F1)	(Acc)
Unigram	623	2.89	**52.6**	58.9 ±0.2	54.0 ±0.4	43.7 ±0.4	**53.2** ±0.3
BPE	**809**	**3.43**	40.5	**66.3** ±0.2	**67.3** ±0.4	**54.5** ±0.5	**53.5** ±0.3
TokMix	689	3.23	44.8	65.4 ±0.3	**66.5** ±0.4	**53.9** ±0.5	**52.3** ±0.3

(b) 20 languages

Table 7.3: Avearged results of evaluation for in-language properties and tasks. Each probing result is an average of 5 random seeds (for 6 languages) and 3 random seeds (for 20 languages). The best value in each metric is underlined, and bolded results are closer than the sum of standard deviations from the optimal value.

Better Masked Language Modeling performance doesn't bring improvement to downstream tasks. In Table 7.3a, we observe that the models performing better on masked token prediction (MRR) tend to be worse on downstream tasks (POS and NER). Average rank provides a possible explanation for this phenomenon. The higher it is, the more vocabulary units a language model needs to consider for masked token filling, making masked word prediction harder. At the same time, a high average rank means that the vocabulary is broader and contains lexical units important for downstream tasks. Again, this trend does not hold for the results for NoOverlap setting, in which the search space for the masked-word problem is limited to the language-specific tokens leading to the best performance in MLM and syntactic tasks (POS and dependency label prediction).

Finding 19

Higher *vocabulary allocation* correlates with lower results in intrinsic language modeling evaluation (MRR) while benefiting performance downstream evaluation. The longer tokens are harder to predict in the masked language modeling task, but they carry more information that is useful for end tasks.

Impact of *vocabulary overlap* on cross-lingual transfer varies across tasks. We observed that NoOverlap approach obtains competitive results for POS tagging. Surprisingly prohibiting sharing vocanulary units also improves cross-lingual transfer in the task among languages with Latin script (shown in Figure 7.2b). We think that the reason behind the strength of NoOverlap approach is that particular tokens have different meanings across languages. For instance, the word "a" is an indefinite article in English and a preposition in Spanish.

Nevertheless, vocabulary overlap is crucial to cross-lingual transfer in some tasks. Especially NER within the same script languages (Figure 7.2a) and sentence-level tasks. For these tasks, NoOverlap significantly underperforms other tokenization methods. The drop within Latin script languages is in the range: $6.8 - 11.3\%$ for NER and $12.7 - 15.9\%$ for sentence retrieval. In these cases, usage of the same tokens can indicate that texts refer to the same entities across languages, e.g., names are usually the same or similar strings in the languages sharing writing system.

	V. Overlap	V. Allocation SRC		V. Allocation TGT	
	(JSD)	(AR)	(CPT)	(AR)	(CPT)
NER	-0.111	**0.249**	**0.33**	0.209	**0.28**
POS	**0.395**	**0.365**	**0.547**	**0.489**	**0.653**
Dep l.	**0.463**	0.19	**0.425**	**0.249**	**0.44**
NLI	**-0.516**	**0.421**	0.203	**0.297**	0.103
Retrieval	**-0.648**	**0.235**	0.082	**0.238**	0.085

Table 7.5: Spearman's correlations between cross-lingual transfer results and tokenization measures. *vocabulary overlap* is measured by JSD, we also measure the correlation with *vocabulary allocation* of source and target language of the transfer directions. Statistically significant correlations ($p < 0.01$) are bolded. Computed for six languages.

Finding 20

High vocabulary overlap is helpful for tasks that benefit from similar orthographic forms across languages, e.g. named entities, and bilingual sentence retrieval. Nevertheless, it can be detrimental in syntactic tasks (POS, UD) where the same orthographic forms tend to have different morphosyntactic functions across languages.

Results generalize to the larger set of languages. The key observation for six language sets holds in the model trained for twenty languages. Table 7.3b shows that BPE and TokMix obtain better *vocabulary allocation* than Unigram leading to improved results for word-level downstream tasks (NER, POS, Dependency labeling). Due to the smaller vocab size to the language ratio ($\frac{V}{L}$), average ranks decrease for all methods in comparison to the six language setting. We observe in Table 7.2b that the cross-language vocabulary overlap is the highest for Unigram and lowest for BPE, similar to the six languages settings. However, the association between *vocabulary allocation* and the cross-lingual transfers is less pronounced.

7.3.3 Statistical Analysis

In this analysis, we check the statistical significance of the observed correlations between the tokenizers' properties and the intrinsic and end-task evaluation.

In Table 7.4, we show that the strong relationship between *vocabulary allocation* (AR and CPT) and language model performance (MRR) is statistically supported. The length of token units has a strong positive influence on POS, dependency labeling, and NER results ($r > 0.65$) and a negative influence on MRR ($r < -0.9$), while it

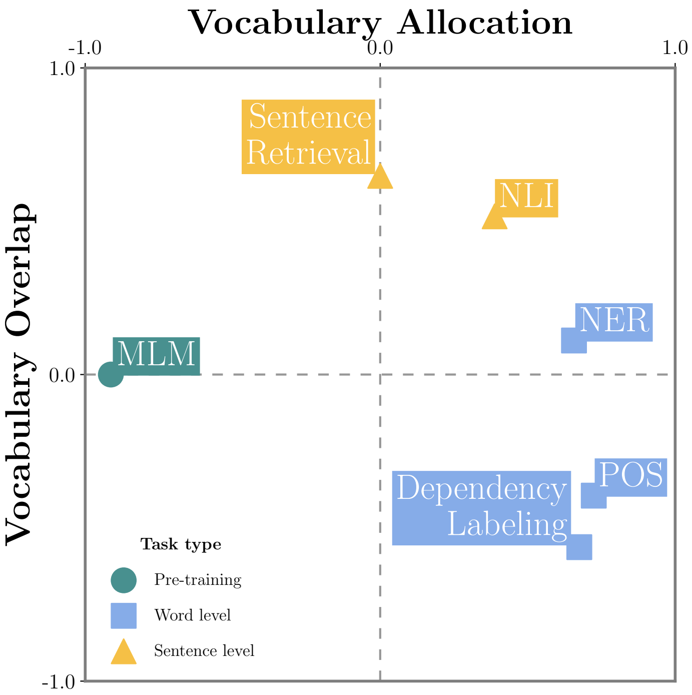


Figure 7.3: Mapping the impact of *vocabulary allocation* and *vocabulary overlap* on language model performance. The location of points corresponds to Spearmnan's correlation between vocabulary measures and the task score (see the details in Tables 7.4 and 7.5). High *vocabulary overlap* benefits NER and sentence-level tasks (NLI, sentence retrieval) and hinders POS and dependency labeling performance. High *vocabulary allocation* improves word-level tasks but leads to a decrease in masked language modeling scores. For exact values of correlation refer to Tables 7.4 and 7.5. Masked language modeling is measured only in language. Thus it's unaffected by *vocabulary overlap*. Analogically, sentence retrieval is solely cross-lingual and unaffected by *vocabulary allocation*.

does not significantly affect NLI results. The correlation between the average rank and MRR, NER scores is weaker but still significant. Moreover, it is significantly correlated with XNLI accuracy with a medium coefficient $r = 0.56$, even though the changes in XNLI are low across tokenizers.

Table 7.5 presents the correlations for cross-lingual transfer scores with JSD measuring *vocabulary overlap*. The coefficient supports our previous observation that lower overlap (thus higher JSD) improves transfer for POS tagging and dependency labeling and deteriorates it for other tasks. However, the correlation for NER is not significant. The *vocabulary allocation* of source and target languages significantly influence the cross-lingual transfers. Similarly to the in-language correlations, the influence of characters per token is more substantial on word-level tasks, while average rank affects sentence-level tasks to a larger extent. This observation underscores the importance of allocating a sufficient portion of vocabulary for low-resource for better cross-lingual transfer.

To summarize the results of correlation analysis, we plot the coefficients in Figure 7.3, showing to what extent *vocabulary allocation* impacts the downstream performance and *vocabulary overlap* affects cross-lingual transfer.

	V. Allocation		**MLM**
	(AR)	(CPT)	(MRR)
CPT	**0.790**	-	-
MRR	**-0.723**	**-0.913**	-
NER	**0.394**	**0.657**	**-0.745**
POS	0.320	**0.724**	**-0.754**
Dep l.	0.266	**0.675**	**-0.695**
NLI	**0.56**	0.388	**-0.437**

Table 7.4: Spearman's correlations between task coefficients for in-language results and tokenizer measures. Statistically significant correlations ($p < 0.01$) are bolded. Computed for 20 languages.

7.4 Conclusions

In this chapter, we introduced a new framework for the evaluation of multilingual subword tokenizers. We made notable observations regarding the impact of vocabulary choice and the model's performance and cross-lingual transfer across diverse languages and end tasks:

1. Including longer and more diverse vocabulary units (higher *vocabulary allocation*) improves in-language results and cross-lingual transfers for word-level tasks.

2. *Vocabulary overlap* is beneficial for cross-lingual transfer in sentence-level tasks.

3. Among languages with the same script, *vocabulary overlap* improves transfer for NER and deteriorates it for POS and dependency labeling.

8

Discusion and Related-Work

The section summarizes the findings of this thesis and discusses the results in the broader context.

8.1 Distribution of Linguistic Information across Layers

The work of Tenney et al. (2019) showed the different layers of BERT specialize in encoding specific types of information. They drew an analogy between the information flow in the Transformer and a sequential NLP pipeline, consisting of multiple intermediate components responsible for different levels of linguistic comprehension, i.e., lexical, syntactic, and semantic.

Figure 8.1 (adapted from our work Limisiewicz and Marecek, 2020) summarizes the evaluation of syntactic information across layers for different approaches. In language models: BERT, mBERT, and GPT-2, the middle layers are the most syntactic. In neural machine translation models, the top layers of the encoder are the most syntactic. However, it is important to note that the MT Transformer encoder is only the first half of the whole translation architecture, and therefore the most syntactic layers are, in fact, in the middle of the process.

Analogously Figure 8.2 shows the distribution of semantic and lexical information across layers. Contrasting to syntax we observe, that the lexical signal is better captured by the lower layers, it is especially stark in the analysis methods that do not involve probing (B and C). In both figures, we observe that the information distribution across layers in probing (Hewitt and Manning, 2019; Chi et al., 2020)

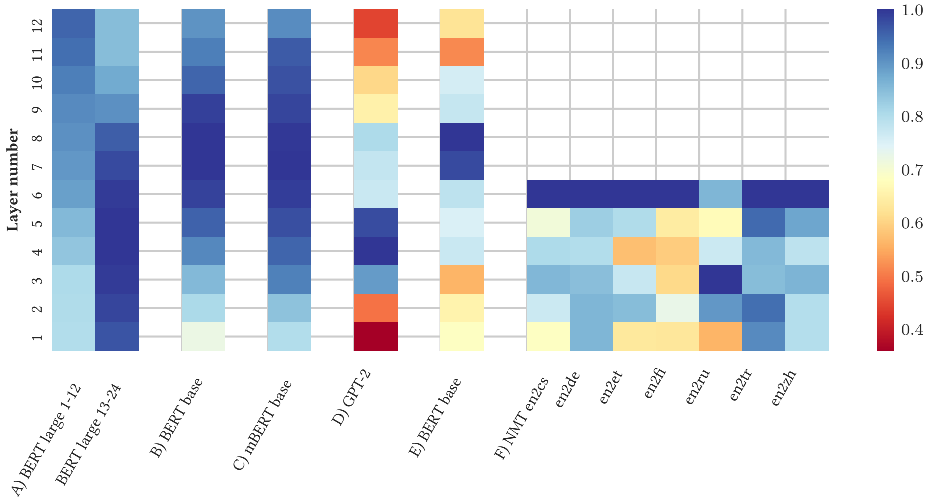


Figure 8.1: Distribution of syntactic information across layers in different Transformer models. The values are normalized so that the best layer for each method is assigned 1.0. The columns A), B), C), and G) show undirected UAS trees extracted by probing the n-th layer Limisiewicz and Mareček (2021b); Hewitt and Manning (2019); Chi et al. (2020). Column D) shows the dependency alignment averaged across all heads in each layer Vig and Belinkov (2019). The columns E) and F) show UAS of trees induced from attention heads by the maximum spanning tree algorithm Raganato and Tiedemann (2018); Limisiewicz et al. (2020). The results for the best layer (corresponding to value 1.0 in the plot) are: A) 84.2; B) 79.8; C) 80.1; D) 22.3; E) 24.3; F) en2cs: 23.9, en2de: 20.9, en2et: 22.1, en2fi: 24.0, en2ru: 22.4, en2tr: 17.5, en2zh: 21.6; G) 77.0

is smoother than in unsupervised methods (Limisiewicz et al., 2020; Raganato and Tiedemann, 2018). This observation hints that probing may be prone to producing a false indication of the linguistic information encoded in the model, due to training on supervised data.

Finding 21

Our survey of previous studies investigating linguistic features encoding in language models shows that syntactic information tends to be encoded in mid-upper layers while lexical is more prevalent in mid-lower ones.

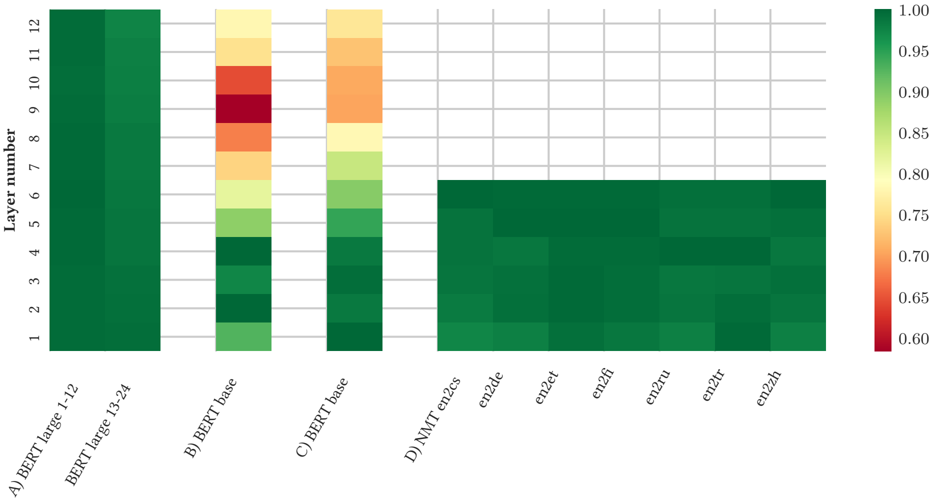


Figure 8.2: Distribution of semantic lexical information across layers in different Transformer models. The values are normalized so that the best layer for each method is assigned 1.0. Column A) shows results of probing in hypernymy structure (Limisiewicz and Mareček, 2021b). Column B) presents accuracy on word analogy tasks (BATS) and C) correlation with lexical annotations (LSIM) both reported in Vig and Belinkov (2019). Columns D) show the results of probing for semantic tags from Raganato and Tiedemann (2018). The results for the best layer (corresponding to value 1.0 in the plot) are: A) 90.5; B) 29.3; C) 51.3; D) en2cs: 86.3, en2de: 85.9, en2et: 82.4, en2fi: 83.0, en2ru: 84.6, en2tr: 78.7, en2zh: 86.0

8.2 Distribution of Linguistic Information across Components

We can study the distribution of captured signals in the model not only across layers but also in different modules, as described throughout this thesis: attention, feed-forward, and latent embeddings.

Input and Output Embeddings In the first wave of popularity of Transformer models, input and output embeddings were crucial to encode lexical information (Musil, 2019). In encoder models, the parameters of the embedding layers constituted often the majority of the model's parameters: in BERT approximately 50% (Devlin et al., 2019), XLM-R: 70% (Conneau et al., 2020a). For XLM-V(Liang et al., 2023), which stands out with 1 million tokens vocabulary, the share of parameters in the embedding layer reaches 90%. Chapter 7 describes how vocabulary units, and by extension the parameters of embedding layers, are allocated to specific languages in the multilingual models. *Vocabulary allocation* has a strong impact on the model's end-task performance in particular languages.

Attention The attention mechanism in Transformers is crucial for efficient contextualization of the latent representations, i.e. enabling information flow between positions of the input sequence. Analytical studies have shown that the patterns obtained in attention weight matrices are often aligned with syntactic (Voita et al., 2019; Clark et al., 2019; Vig and Belinkov, 2019) or semantic structures (Wu et al., 2020) as annotated in linguistic corpora. Our results in Chapter 6 confirm this similarity for dependency parses, but we also observe that there are systematic differences between the linguistic theory and the emergent structures. In line with these findings, (Kulmizev et al., 2020) shows that some linguistic annotations (e.g., surface dependencies Gerdes et al. (2018)) are easier to be predicted than from the latent vectors.

Feed-Forward In the LLMs, unlike their predecessor, the largest chunk of the model's parameters is stored in the feed-forward layers (Geva et al., 2021). This component is also the one that is the most benefited in the scaling of the models (Kaplan et al., 2020). Multiple recent works have focused on feed-forwards in the pursuit of explaining the models' functions. These studies identified the feed-forwards as a crucial component for the model's comprehension of the world knowledge and information (Meng et al., 2022; Dai et al., 2022; Merullo et al., 2023), e.g. as determined by *causal tracing* (Pearl, 2009). Following this line of research in Chapter 5, we used causal tracing to single out mid-upper feed-forward layers as responsible for encoding gender bias. We also showed, similarly to (Meng et al., 2023), that oriented editing of parameters in these modules can mitigate the specific signal from the model.

Based on this observation, we conclude that the distribution of the information across modules varies across sizes and architectural choices. The recent tendency of the rising model's scales prioritizes the feed-forward parameters over previously dominating embedding layers. This change has been reflected in the modules' capacity to encode information. It has been also observed that feed-forwards are responsible for models' learning capabilities (Pires et al., 2023).

8.3 Modules Crucial for Multilinguality

The most important direction of this work is the interpretability of the multilingual models.

Previous work studied the impact of architectural and training choices on the multilinguality of the model, or in other words, how well the model is capable of sharing knowledge across languages. The issue is especially interesting because the models demonstrate the capability to transfer knowledge across languages, even with minimal in-language signal. To this end, Conneau et al. (2020b) studied the importance of sharing different components of the models trained on data in various languages. They show the importance of sharing the parameters of top Transformer layers and demonstrate the lower impact of shared vocabularies. Dufter and Schütze (2020) performed an in-depth study of a similar phenomenon, showing the most influential factor enabling cross-lingual transfer is the depth of the model. The study of K et al. (2020) does not indicate the necessity of same input embeddings across languages but identifies other aspects of the input representation as crucial: positional embeddings and shared special tokens.

Our results in Chapter 7 show similar tendencies: the shared vocabulary is not always necessary for the model to transfer knowledge. However, for some tasks, its influence is significant, e.g. in named entities, where we expect similarities across languages, this trend was also observed by Patil et al. (2022).

8.4 Sharing Representation Across Languages

We also investigate the shared representation of diverse languages in the latent space. The previously mentioned observation of the benefit of shared dense layers across languages in multilingual models (Conneau et al., 2020b) suggests that they learn similar representations for different languages, which is beneficial for cross-lingual transfer. Indeed, the empirical results of past research show that the per-language distributions of embeddings are isomorphic given that they were obtained based on data of similar quality and size (Vulić et al., 2020). Libovický et al. (2020) observed that the embeddings of the multilingual masked language model (mBERT) are to a large extent language-neutral. Furthermore, Wu and Dredze (2020) and Chi et al. (2020) showed that syntactic information is encoded uniformly across languages and retrievable with shared parser or probe (respectively). To this end, in Section 4.4, we show the results confirming that linguistic signals are distributed uniformly in latent space across typologically similar languages, while the representation of diverse languages can be aligned with an orthogonal rotation based on minimal supervision. Aligning benefits cross-lingual transfer in syntactic parsing. The effectiveness of orthogonal rotation for aligning similar concepts across languages, known also as *bilingual induction*, has been a widely researched topic in multilingual NLP (Søgaard et al., 2018; Artetxe et al., 2018; Vulić et al., 2020; Wang et al., 2019; Marchisio et al., 2021).

9 Conclusion

The main goal of this thesis is to offer a better understanding of language models function. We focus on mapping signals and biases learned from the data in specific models' components and identifying the key factors affecting their mono- and cross-lingual comprehension. To achieve this goal, we have proposed a new methodology and also followed already established methods to study the local behavior of model components. Namely:

1. We have introduced the new concept of *orthogonal probe*s to study the distribution of linguistic signals in the model's embeddings. *Orthogonal probe*s allow us also to disentangle various types of information, e.g. syntactic and semantic. We have shown that the linguistic signals are distributed in embeddings across diverse languages. We have proposed two practical applications of *orthogonal probe*s: zero-shot cross-lingual parsing and filtering out unwanted biases.

2. We have conducted a deeper analysis of the biases encoded in the models with the use of causal tracing. *Causal tracing* identified mid-upper feed-forward layers as the main culprit of models' gender bias. Based on this observation, we have proposed a novel debiasing method, DAMA.

3. We have analyzed the alignment between syntactic structures and patterns emerging in the attention weights, showing its potential for parsing with limited supervision. This method has been shown to effectively reduce various manifestations of gender bias in the LLaMA models while preserving their high performance on downstream tasks.

4. Finally, we have introduced a new framework for the evaluation of the impact of multilingual subword vocabulary on the model's performance in various languages and in cross-lingual transfer. We notably observed the necessity of allocating sufficient capacity of the embedding layer for a language to improve in-language comprehension and the lower role of sharing vocabulary across languages.

Overall, throughout the thesis, we have shown that the interpretability analysis of language models can provide theoretical insights into the opaque neural models. Moreover, such low-level observations of the model's interiors enabled the development of new methods for high-precision control of specific behaviors. Such an approach is crucial in practice, as biases and unwanted signals are not fully accounted for in ever-growing training corpora and often slip through data filters (Navigli et al., 2023). We have shown that effort in models' interpretation is necessary and complementary to the other advances in the field of NLP, i.e., research on scaling, efficiency, and new architectural design.

9.1 Limitation

Some of our results are based on probing experiments (see Section 4.1), in which we kept a model intact and fine-tuned the shallow network (or probe) to estimate how well the model represents specific linguistic phenomena. This approach has been criticized due to its reliance on data and the associated risk of "squinting eyes on the data", which is a metaphor for interpreting the probe's high results as the indicator of the model's comprehension while it might be just the results of memorization in the probe (Rogers et al., 2020; Belinkov, 2022a). The memorization is highly likely when using complex probes and simple tasks (e.g. POS tagging) (Pimentel et al., 2020; Hewitt and Liang, 2019). In Chapter 4, we compare probe results in retrieving linguistic structures with the results for randomly sampled ones to control for memorization and show that particularly *orthogonal probe*s are less prone to this risk.

Another criticism is related to the fact that to estimate linguistic comprehension, we base the evaluation on annotation schemes that are to some extent arbitrary. To this end in Chapter 6, we show that the emergent structures observed in BERT are systematically different from the syntactic annotation in UD. The subjectivity of annotation is especially visible in the benchmarks related to social phenomena, e.g. biases, that are highly affected by their creators' and annotators' social background and personal beliefs. It has been observed that the results of bias estimations are often inconsistent, making it hard to compare the results of different studies and models (Delobelle et al., 2022; van der Wal et al., 2024). However, we still believe that reliance on human annotation is necessary to interpret the model's function, and can reveal consistent trends when we intervene in specific components.

A few recent studies have proposed an alternative formulation of probing, focusing on the interventions in the model's representation e.g. by nullifying linguistic signals (Elazar et al., 2021) or patching latent representation (Vig et al., 2020; Meng et al., 2022; Hendel et al., 2023) to observe the effect on the model's output. Such approaches are more truthful to the base model function, as they do not involve additional complexity as in the case of probing. However, similarly to probing, they need to rely on limited supervision in designing the interventions.

9.2 Future Challenges

The field of NLP is going through a period of rapid advancement and changes, altering the architecture and scale of the models, which makes explaining the function of language models challenging. For instance in new larger models, feed-forward layers tend to be more heavily scaled up than embedding and attention layers. There are also differences in architectural approaches suggesting replacing some of the current approaches with more efficient solutions, to name a few: replacing attention with space-state models (Gu and Dao, 2023) or amending the structure of feed-forward for better scaling (Liu et al., 2024). However, time and further empirical studies will determine if and when this solution will be adopted, and potentially break Transformer domination in NLP.

The fast pace of new models' introduction is also a challenge for developing analysis methods that stand the test of time. Notably, the pursuit of explanation is an ongoing effort and it is unlikely to be solved in the foreseeable future. Nevertheless, some directions are more universal and more likely to generalize to the new architectures. We think that such approaches should be independent of the specific architectural choices, e.g. multi-head attention (as described in Chapter 6) and use methods that consider universal types of representations, e.g. latent vectors (Chapter 4). Alternatively, the robustness of an analytical method is determined by whether it can be applied to different types of modules to estimate their direct contribution of a module or data to the model's output, such as causal tracing (Chapter 5).

We also expect that multilingual and multimodal interpretability analysis can gain more attention in the future. The upcoming studies could further explore the interaction of signals coming from data in different languages and modalities, and investigate the transfer across them.

9.3 Acknowledgment of Co-Authorship

The research is not a one-person effort, and experiments presented in this work were conducted in collaboration with others. In order to maintain comprehension of the text and present all relevant results, we decided to keep this contribution and acknowledge their authors here:

1. In Chapter 5, we acknowledge the work of David Mareček, who analyzed and proposed the prompts and noise coefficients in the *causal tracing*. He also implemented an evaluation on StereoSet. Furthermore, we thank Tomaš Musil for evaluating the original and edited models on downstream tasks: MMLU, ARC, OpenBookQA. Finally, we thank Paul Maouret for implementing and running LoRA finetuning on the models as a valuable baseline for our debiasing method.

2. In Chapeter 6, we recognize the input of Rudolf Rosa, who provided the initial version of the code for visualizing attention head weights, mentored the development of the new ensembling method and consulted the modification we made in universal dependencies.

3. In Chapter 7, we thank Jiří Balhar for discussing the metrics for tokenization. He also implemented the evaluation script for XNLI probes.

Lastly, I acknowledge the help of my advisor, David Mareček, who provided substantive comments for all described experiments. I also thank him, Jindřich Libovický, and Rudolf Rosa for comments and suggestions to earlier versions of this thesis.

Bibliography

Abeillé, A. – Clément, L. – Liégeois, L. Un corpus arboré pour le français : le French Treebank [A parsed corpus for French: the French treebank]. *Traitement Automatique des Langues*. 2019, 60, 2, p. 19–43. Available at: https://aclanthology.org/2019.tal-2.2.

Artetxe, M. – Schwenk, H. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*. 2019, 7, p. 597–610. doi: 10.1162/tacl_a_00288. Available at: https://aclanthology.org/Q19-1038.

Artetxe, M. – Labaka, G. – Agirre, E. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 789–798, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1073. Available at: https://aclanthology.org/P18-1073.

Ba, L. J. – Kiros, J. R. – Hinton, G. E. Layer Normalization. *CoRR*. 2016, abs/1607.06450. Available at: http://arxiv.org/abs/1607.06450.

Bahdanau, D. – Cho, K. – Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Bengio, Y. – LeCun, Y. (Ed.) *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015a. Available at: http://arxiv.org/abs/1409.0473.

Bahdanau, D. – Cho, K. – Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Bengio, Y. – LeCun, Y. (Ed.) *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015b. Available at: http://arxiv.org/abs/1409.0473.

Bangalore, S. – Joshi, A. K. Supertagging: An Approach to Almost Parsing. *Computational Linguistics*. 1999, 25, 2, p. 237–265. Available at: https://aclanthology.org/J99-2004.

Bansal, N. – Chen, X. – Wang, Z. Can We Gain More from Orthogonality Regularizations in Training Deep CNNs? *CoRR*. 2018, abs/1810.09102. Available at: `http://arxiv.org/abs/1810.09102`.

Bau, A. – Belinkov, Y. – Sajjad, H. – Durrani, N. – Dalvi, F. – Glass, J. R. Identifying and Controlling Important Neurons in Neural Machine Translation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. Available at: `https://openreview.net/forum?id=H1z-PsR5KX`.

Belinkov, Y. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*. March 2022a, 48, 1, p. 207–219. doi: 10.1162/coli_a_00422. Available at: `https://aclanthology.org/2022.cl-1.7`.

Belinkov, Y. Probing Classifiers: Promises, Shortcomings, and Advances. *Comput. Linguistics*. 2022b, 48, 1, p. 207–219. doi: 10.1162/COLI_A_00422. Available at: `https://doi.org/10.1162/coli_a_00422`.

Bender, E. M. – Gebru, T. – McMillan-Major, A. – Shmitchell, S. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, p. 610–623, New York, NY, USA, 2021. Association for Computing Machinery. doi: 10.1145/3442188.3445922. Available at: `https://doi.org/10.1145/3442188.3445922`. ISBN 9781450383097.

Bengio, Y. – Ducharme, R. – Vincent, P. – Janvin, C. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.* 2003, 3, p. 1137–1155. Available at: `http://jmlr.org/papers/v3/bengio03a.html`.

Bjerva, J. – Plank, B. – Bos, J. Semantic Tagging with Deep Residual Networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, p. 3531–3541, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. Available at: `https://aclanthology.org/C16-1333`.

Black, S. et al. GPT-NeoX-20B: An Open-Source Autoregressive Language Model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, p. 95–136, virtual+Dublin, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.bigscience-1.9. Available at: `https://aclanthology.org/2022.bigscience-1.9`.

Blevins, T. – Levy, O. – Zettlemoyer, L. Deep RNNs Encode Soft Hierarchical Syntax. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, p. 14–19, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2003. Available at: `https://aclanthology.org/P18-2003`.

Bolukbasi, T. – Chang, K. – Zou, J. Y. – Saligrama, V. – Kalai, A. T. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In Lee, D. D. – Sugiyama, M. – Luxburg, U. – Guyon, I. – Garnett, R. (Ed.) *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, p. 4349–4357, 2016. Available at: `https://proceedings.neurips.cc/paper/2016/hash/a486cd07e4ac3d270571622f4f316ec5-Abstract.html`.

Bommasani, R. et al. On the Opportunities and Risks of Foundation Models. *CoRR*. 2021, abs/2108.07258. Available at: `https://arxiv.org/abs/2108.07258`.

Bond, F. – Foster, R. Linking and Extending an Open Multilingual Wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 1352–1362, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. Available at: `https://aclanthology.org/P13-1133`.

Bowman, S. R. – Angeli, G. – Potts, C. – Manning, C. D. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. Available at: `https://aclanthology.org/D15-1075`.

Brandl, S. – Bugliarello, E. – Chalkidis, I. On the Interplay between Fairness and Explainability. *CoRR*. 2023, abs/2310.16607. doi: 10.48550/ARXIV.2310.16607. Available at: `https://doi.org/10.48550/arXiv.2310.16607`.

Brants, S. – Dipper, S. – Eisenberg, P. – Hansen-Schirra, S. – König, E. – Lezius, W. – Rohrer, C. – Smith, G. – Uszkoreit, H. TIGER: Linguistic Interpretation of a German Corpus. *Research on Language and Computation*. 2004, 2, p. 597–620. Available at: `https://api.semanticscholar.org/CorpusID:62554779`.

Brown, T. B. et al. Language Models are Few-Shot Learners. In Larochelle, H. – Ranzato, M. – Hadsell, R. – Balcan, M. – Lin, H. (Ed.) *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. Available at: `https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html`.

Caliskan, A. – Bryson, J. J. – Narayanan, A. Semantics derived automatically from language corpora contain human-like biases. *Science*. 2017, 356, 6334, p. 183–186. doi: 10.1126/science.aal4230. Available at: `https://www.science.org/doi/abs/10.1126/science.aal4230`.

Chi, E. A. – Hewitt, J. – Manning, C. D. Finding Universal Grammatical Relations in Multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 5564–5577, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.493. Available at: `https://aclanthology.org/2020.acl-main.493`.

Chinchor, N. A. Overview of MUC-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*, 1998. Available at: `https://aclanthology.org/M98-1001`.

Cho, K. – Merriënboer, B. – Gulcehre, C. – Bahdanau, D. – Bougares, F. – Schwenk, H. – Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. Available at: `https://aclanthology.org/D14-1179`.

Choromanski, K. M. – Likhosherstov, V. – Dohan, D. – Song, X. – Gane, A. – Sarlós, T. – Hawkins, P. – Davis, J. Q. – Mohiuddin, A. – Kaiser, L. – Belanger, D. B. – Colwell, L. J. – Weller, A. Rethinking Attention with Performers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. Available at: `https://openreview.net/forum?id=Ua6zuk0WRH`.

Clark, K. – Khandelwal, U. – Levy, O. – Manning, C. D. What Does BERT Look at? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, p. 276–286, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. Available at: `https://aclanthology.org/W19-4828`.

Clark, P. – Cowhey, I. – Etzioni, O. – Khot, T. – Sabharwal, A. – Schoenick, C. – Tafjord, O. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *CoRR*. 2018, abs/1803.05457. Available at: `http://arxiv.org/abs/1803.05457`.

Conneau, A. – Lample, G. Cross-lingual Language Model Pretraining. In Wallach, H. M. – Larochelle, H. – Beygelzimer, A. – d'Alché-Buc, F. – Fox, E. B. – Garnett, R. (Ed.) *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, p. 7057–7067, 2019a. Available at: `https://proceedings.neurips.cc/paper/2019/hash/c04c19c2c2474dbf5f7ac4372c5b9af1-Abstract.html`.

Conneau, A. – Lample, G. Cross-lingual Language Model Pretraining. In Wallach, H. M. – Larochelle, H. – Beygelzimer, A. – d'Alché-Buc, F. – Fox, E. B. – Garnett, R. (Ed.) *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, p. 7057–7067, 2019b. Available at: `https://proceedings.neurips.cc/paper/2019/hash/c04c19c2c2474dbf5f7ac4372c5b9af1-Abstract.html`.

Conneau, A. – Kruszewski, G. – Lample, G. – Barrault, L. – Baroni, M. 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)*, p. 2126–2136, Melbourne, Australia, July 2018a. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. Available at: `https://aclanthology.org/P18-1198`.

Conneau, A. – Rinott, R. – Lample, G. – Williams, A. – Bowman, S. – Schwenk, H. – Stoyanov, V. XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 2475–2485, Brussels, Belgium, October-November 2018b. Association for Computational Linguistics. doi: 10.18653/v1/D18-1269. Available at: `https://aclanthology.org/D18-1269`.

Conneau, A. – Khandelwal, K. – Goyal, N. – Chaudhary, V. – Wenzek, G. – Guzmán, F. – Grave, E. – Ott, M. – Zettlemoyer, L. – Stoyanov, V. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 8440–8451, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. Available at: `https://aclanthology.org/2020.acl-main.747`.

Conneau, A. – Wu, S. – Li, H. – Zettlemoyer, L. – Stoyanov, V. Emerging Cross-lingual Structure in Pretrained Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 6022–6034, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.536. Available at: `https://aclanthology.org/2020.acl-main.536`.

Dai, D. – Dong, L. – Hao, Y. – Sui, Z. – Chang, B. – Wei, F. Knowledge Neurons in Pretrained Transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 8493–8502, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.581. Available at: `https://aclanthology.org/2022.acl-long.581`.

Dao, T. – Fu, D. Y. – Ermon, S. – Rudra, A. – Ré, C. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In *Advances in Neural Information Processing Systems*, 2022.

De-Arteaga, M. – Romanov, A. – Wallach, H. M. – Chayes, J. T. – Borgs, C. – Chouldechova, A. – Geyik, S. C. – Kenthapadi, K. – Kalai, A. T. Bias in Bios: A Case Study of Semantic Representation Bias in a High-Stakes Setting. In boyd – Morgenstern, J. H. (Ed.) *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019*, p. 120–128. ACM, 2019. doi: 10.1145/3287560.3287572. Available at: `https://doi.org/10.1145/3287560.3287572`.

Marneffe, M. – Manning, C. D. – Nivre, J. – Zeman, D. Universal Dependencies. *Comput. Linguistics*. 2021, 47, 2, p. 255–308. doi: 10.1162/COLI_A_00402. Available at: `https://doi.org/10.1162/coli_a_00402`.

Delobelle, P. – Tokpo, E. – Calders, T. – Berendt, B. Measuring Fairness with Biased Rulers: A Comparative Study on Bias Metrics for Pre-trained Language Models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 1693–1706, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.122. Available at: `https://aclanthology.org/2022.naacl-main.122`.

Devlin, J. – Chang, M.-W. – Lee, K. – Toutanova, K. 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)*, p. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. Available at: https://aclanthology.org/N19-1423.

Dozat, T. – Manning, C. D. 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, 2017. Available at: https://openreview.net/forum?id=Hk95PK9le.

Drozd, A. – Gladkova, A. – Matsuoka, S. Word Embeddings, Analogies, and Machine Learning: Beyond king - man + woman = queen. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, p. 3519–3530, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. Available at: https://aclanthology.org/C16-1332.

Dryer, M. S. – Haspelmath, M. (Ed.). *WALS Online (v2020.3)*. Zenodo, 2013. doi: 10.5281/zenodo.7385533. Available at: https://doi.org/10.5281/zenodo.7385533.

Dufter, P. – Schütze, H. Identifying Elements Essential for BERT's Multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 4423–4437, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.358. Available at: https://aclanthology.org/2020.emnlp-main.358.

Edmonds, J. Optimums Branchings. November 1966, 71B.

Elazar, Y. – Ravfogel, S. – Jacovi, A. – Goldberg, Y. Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals. *Transactions of the Association for Computational Linguistics*. 2021, 9, p. 160–175. doi: 10.1162/tacl_a_00359. Available at: https://aclanthology.org/2021.tacl-1.10.

Francis, W. N. A Standard Corpus of Edited Present-Day American English. *College English*. 1965, 26, 4, p. 267–273. ISSN 00100994. Available at: http://www.jstor.org/stable/373638.

GERDES, K. – GUILLAUME, B. – KAHANE, S. – PERRIER, G. SUD or Surface-Syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, p. 66–74, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6008. Available at: `https://aclanthology.org/W18-6008`.

GEVA, M. – SCHUSTER, R. – BERANT, J. – LEVY, O. Transformer Feed-Forward Layers Are Key-Value Memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. Available at: `https://aclanthology.org/2021.emnlp-main.446`.

GLOROT, X. – BORDES, A. – BENGIO, Y. Deep Sparse Rectifier Neural Networks. In GORDON, G. J. – DUNSON, D. B. – DUDÍK, M. (Ed.) *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, 15 / *JMLR Proceedings*, p. 315–323. JMLR.org, 2011. Available at: `http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf`.

GOLDBERGER, A. – SHENHART, W. – WILKS, S. *Econometric Theory.* WILEY SERIES in PROBABILITY and STATISTICS: APPLIED PROBABILITY and STATIST ICS SECTION Series. J. Wiley, 1964. Available at: `https://books.google.com/books?id=KZq5AAAAIAAJ`. ISBN 978-0-471-31101-0.

GU, A. – DAO, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *CoRR*. 2023, abs/2312.00752. doi: 10.48550/ARXIV.2312.00752. Available at: `https://doi.org/10.48550/arXiv.2312.00752`.

HAJIČ, J. – BEJČEK, E. – HLAVACOVA, J. – MIKULOVÁ, M. – STRAKA, M. – ŠTĚPÁNEK, J. – ŠTĚPÁNKOVÁ, B. Prague Dependency Treebank - Consolidated 1.0. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, p. 5208–5218, Marseille, France, May 2020. European Language Resources Association. Available at: `https://aclanthology.org/2020.lrec-1.641`. ISBN 979-10-95546-34-4.

HAN, W. – JIANG, Y. – TU, K. Enhancing Unsupervised Generative Dependency Parser with Contextual Information. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 5315–5325, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1526. Available at: `https://aclanthology.org/P19-1526`.

HENDEL, R. – GEVA, M. – GLOBERSON, A. In-Context Learning Creates Task Vectors. In BOUAMOR, H. – PINO, J. – BALI, K. (Ed.) *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, p. 9318–9333. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.624. Available at: https://doi.org/10.18653/v1/2023.findings-emnlp.624.

HENDRYCKS, D. – BURNS, C. – BASART, S. – ZOU, A. – MAZEIKA, M. – SONG, D. – STEINHARDT, J. Measuring Massive Multitask Language Understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. Available at: https://openreview.net/forum?id=d7KBjmI3GmQ.

HEWITT, J. – LIANG, P. 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)*, p. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. Available at: https://aclanthology.org/D19-1275.

HEWITT, J. – MANNING, C. D. 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)*, p. 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. Available at: https://aclanthology.org/N19-1419.

HOCHREITER, S. – SCHMIDHUBER, J. Long Short-Term Memory. *Neural Comput.* 1997, 9, 8, p. 1735–1780. doi: 10.1162/NECO.1997.9.8.1735. Available at: https://doi.org/10.1162/neco.1997.9.8.1735.

HOFFMANN, J. et al. Training Compute-Optimal Large Language Models. *CoRR.* 2022, abs/2203.15556. doi: 10.48550/ARXIV.2203.15556. Available at: https://doi.org/10.48550/arXiv.2203.15556.

HU, E. J. – SHEN, Y. – WALLIS, P. – ALLEN-ZHU, Z. – LI, Y. – WANG, S. – WANG, L. – CHEN, W. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022. Available at: https://openreview.net/forum?id=nZeVKeeFYf9.

Junczys-Dowmunt, M. – Grundkiewicz, R. – Dwojak, T. – Hoang, H. – Heafield, K. – Neckermann, T. – Seide, F. – Germann, U. – Aji, A. F. – Bogoychev, N. – Martins, A. F. T. – Birch, A. Marian: Fast Neural Machine Translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, p. 116–121, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-4020. Available at: `https://aclanthology.org/P18-4020`.

K, K. – Wang, Z. – Mayhew, S. – Roth, D. Cross-Lingual Ability of Multilingual BERT: An Empirical Study. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. Available at: `https://openreview.net/forum?id=HJeT3yrtDr`.

Kaplan, J. – McCandlish, S. – Henighan, T. – Brown, T. B. – Chess, B. – Child, R. – Gray, S. – Radford, A. – Wu, J. – Amodei, D. Scaling Laws for Neural Language Models. *CoRR*. 2020, abs/2001.08361. Available at: `https://arxiv.org/abs/2001.08361`.

Karakanta, A. – Vela, M. – Teich, E. EuroParl-UdS: Preserving and Extending Metadata in Parliamentary Debates. 2018. Available at: `https://api.semanticscholar.org/CorpusID:211268215`.

Kim, T. – Choi, J. – Edmiston, D. – Lee, S. Are Pre-trained Language Models Aware of Phrases? Simple but Strong Baselines for Grammar Induction. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. Available at: `https://openreview.net/forum?id=H1xPR3NtPB`.

Kingma, D. P. – Ba, J. Adam: A Method for Stochastic Optimization. In Bengio, Y. – LeCun, Y. (Ed.) *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. Available at: `http://arxiv.org/abs/1412.6980`.

Kocmi, T. – Limisiewicz, T. – Stanovsky, G. Gender Coreference and Bias Evaluation at WMT 2020. In *Proceedings of the Fifth Conference on Machine Translation*, p. 357–364, Online, November 2020. Association for Computational Linguistics. Available at: `https://aclanthology.org/2020.wmt-1.39`.

Koehn, P. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of Machine Translation Summit X: Papers*, p. 79–86, Phuket, Thailand, September 13-15 2005. Available at: `https://aclanthology.org/2005.mtsummit-papers.11`.

KUDO, T. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 66–75, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1007. Available at: `https://aclanthology.org/P18-1007`.

KUDO, T. – RICHARDSON, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, p. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. Available at: `https://aclanthology.org/D18-2012`.

KULMIZEV, A. – RAVISHANKAR, V. – ABDOU, M. – NIVRE, J. Do Neural Language Models Show Preferences for Syntactic Formalisms? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 4077–4091, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.375. Available at: `https://aclanthology.org/2020.acl-main.375`.

KWIATKOWSKI, T. et al. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*. 2019, 7, p. 452–466. doi: 10.1162/tacl_a_00276. Available at: `https://aclanthology.org/Q19-1026`.

LAUSCHER, A. – RAVISHANKAR, V. – VULIĆ, I. – GLAVAŠ, G. From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 4483–4499, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.363. Available at: `https://aclanthology.org/2020.emnlp-main.363`.

LEVESQUE, H. J. The Winograd Schema Challenge. In *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*. AAAI, 2011. Available at: `http://www.aaai.org/ocs/index.php/SSS/SSS11/paper/view/2502`.

LEWIS, M. – LIU, Y. – GOYAL, N. – GHAZVININEJAD, M. – MOHAMED, A. – LEVY, O. – STOYANOV, V. – ZETTLEMOYER, L. 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*, p. 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. Available at: `https://aclanthology.org/2020.acl-main.703`.

LIANG, D. – GONEN, H. – MAO, Y. – HOU, R. – GOYAL, N. – GHAZVININEJAD, M. – ZETTLEMOYER, L. – KHABSA, M. XLM-V: Overcoming the Vocabulary Bottleneck in Multilingual Masked Language Models. In BOUAMOR, H. – PINO, J. – BALI, K. (Ed.) *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, p. 13142–13152. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.813. Available at: `https://doi.org/10.18653/v1/2023.emnlp-main.813`.

LIBOVICKÝ, J. – ROSA, R. – FRASER, A. On the Language Neutrality of Pre-trained Multilingual Representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, p. 1663–1674, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.150. Available at: `https://aclanthology.org/2020.findings-emnlp.150`.

LIMISIEWICZ, T. – MARECEK, D. Syntax Representation in Word Embeddings and Neural Networks - A Survey. In HOLENA, M. – HORVÁTH, T. – KELEMENOVÁ, A. – MRÁZ, F. – PARDUBSKÁ, D. – PLÁTEK, M. – SOSÍK, P. (Ed.) *Proceedings of the 20th Conference Information Technologies - Applications and Theory (ITAT 2020), Hotel Tyrapol, Oravská Lesná, Slovakia, September 18-22, 2020*, 2718 / *CEUR Workshop Proceedings*, p. 40–50. CEUR-WS.org, 2020. Available at: `https://ceur-ws.org/Vol-2718/paper16.pdf`.

LIMISIEWICZ, T. – MAREČEK, D. Examining Cross-lingual Contextual Embeddings with Orthogonal Structural Probes. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 4589–4598, Online and Punta Cana, Dominican Republic, November 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.376. Available at: `https://aclanthology.org/2021.emnlp-main.376`.

Limisiewicz, T. – Mareček, D. Introducing Orthogonal Constraint in Structural Probes. 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)*, p. 428–442, Online, August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.36. Available at: `https://aclanthology.org/2021.acl-long.36`.

Limisiewicz, T. – Mareček, D. Don't Forget About Pronouns: Removing Gender Bias in Language Models Without Losing Factual Gender Information. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, p. 17–29, Seattle, Washington, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.gebnlp-1.3. Available at: `https://aclanthology.org/2022.gebnlp-1.3`.

Limisiewicz, T. – Mareček, D. – Rosa, R. Universal Dependencies According to BERT: Both More Specific and More General. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, p. 2710–2722, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.245. Available at: `https://aclanthology.org/2020.findings-emnlp.245`.

Limisiewicz, T. – Balhar, J. – Marecek, D. Tokenization Impacts Multilingual Language Modeling: Assessing Vocabulary Allocation and Overlap Across Languages. In Rogers, A. – Boyd-Graber, J. L. – Okazaki, N. (Ed.) *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, p. 5661–5681. Association for Computational Linguistics, 2023a. doi: 10.18653/V1/2023.FINDINGS-ACL.350. Available at: `https://doi.org/10.18653/v1/2023.findings-acl.350`.

Limisiewicz, T. – Marecek, D. – Musil, T. Debiasing Algorithm through Model Adaptation. *CoRR*. 2023b, abs/2310.18913. doi: 10.48550/ARXIV.2310.18913. Available at: `https://doi.org/10.48550/arXiv.2310.18913`.

Limisiewicz, T. – Blevins, T. – Gonen, H. – Ahia, O. – Zettlemoyer, L. MYTE: Morphology-Driven Byte Encoding for Better and Fairer Multilingual Language Modeling. *CoRR*. 2024, abs/2403.10691. doi: 10.48550/ARXIV.2403.10691. Available at: `https://doi.org/10.48550/arXiv.2403.10691`.

LIU, N. F. – GARDNER, M. – BELINKOV, Y. – PETERS, M. E. – SMITH, N. A. 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)*, p. 1073–1094, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1112. Available at: `https://aclanthology.org/N19-1112`.

LIU, Y. – OTT, M. – GOYAL, N. – DU, J. – JOSHI, M. – CHEN, D. – LEVY, O. – LEWIS, M. – ZETTLEMOYER, L. – STOYANOV, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*. 2019b, abs/1907.11692. Available at: `http://arxiv.org/abs/1907.11692`.

LIU, Y. – GU, J. – GOYAL, N. – LI, X. – EDUNOV, S. – GHAZVININEJAD, M. – LEWIS, M. – ZETTLEMOYER, L. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*. 2020a, 8, p. 726–742. doi: 10.1162/tacl_a_00343. Available at: `https://aclanthology.org/2020.tacl-1.47`.

LIU, Y. – OTT, M. – GOYAL, N. – DU, J. – JOSHI, M. – CHEN, D. – LEVY, O. – LEWIS, M. – ZETTLEMOYER, L. – STOYANOV, V. Ro{BERT}a: A Robustly Optimized {BERT} Pretraining Approach, 2020b. Available at: `https://openreview.net/forum?id=SyxS0T4tvS`.

LIU, Z. – WANG, Y. – VAIDYA, S. – RUEHLE, F. – HALVERSON, J. – SOLJACIC, M. – HOU, T. Y. – TEGMARK, M. KAN: Kolmogorov-Arnold Networks. *CoRR*. 2024, abs/2404.19756. doi: 10.48550/ARXIV.2404.19756. Available at: `https://doi.org/10.48550/arXiv.2404.19756`.

MANNING, C. D. – SCHÜTZE, H. *Foundations of statistical natural language processing*. MIT Press, 2001. ISBN 978-0-262-13360-9.

MARCHISIO, K. – PARK, Y. – SAAD-ELDIN, A. – ALYAKIN, A. – DUH, K. – PRIEBE, C. – KOEHN, P. An Analysis of Euclidean vs. Graph-Based Framing for Bilingual Lexicon Induction from Word Embedding Spaces. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, p. 738–749, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.64. Available at: `https://aclanthology.org/2021.findings-emnlp.64`.

MARCUS, M. P. – SANTORINI, B. – MARCINKIEWICZ, M. A. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*. 1993, 19, 2, p. 313–330. Available at: https://aclanthology.org/J93-2004.

MAREČEK, D. – ROSA, R. From Balustrades to Pierre Vinken: Looking for Syntax in Transformer Self-Attentions. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, p. 263–275, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4827. Available at: https://aclanthology.org/W19-4827.

MARECEK, D. – ROSA, R. From Balustrades to Pierre Vinken: Looking for Syntax in Transformer Self-Attentions. In LINZEN, T. – CHRUPALA, G. – BELINKOV, Y. – HUPKES, D. (Ed.) *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP at ACL 2019, Florence, Italy, August 1, 2019*, p. 263–275. Association for Computational Linguistics, 2019. doi: 10.18653/V1/W19-4827. Available at: https://doi.org/10.18653/v1/W19-4827.

MAYHEW, S. – BLEVINS, T. – LIU, S. – SUPPA, M. – GONEN, H. – IMPERIAL, J. M. – KARLSSON, B. F. – LIN, P. – LJUBESIC, N. – MIRANDA, L. – PLANK, B. – RIABI, A. – PINTER, Y. Universal NER: A Gold-Standard Multilingual Named Entity Recognition Benchmark. *CoRR*. 2023, abs/2311.09122. doi: 10.48550/ARXIV.2311.09122. Available at: https://doi.org/10.48550/arXiv.2311.09122.

MCDONALD, R. – NIVRE, J. – QUIRMBACH-BRUNDAGE, Y. – GOLDBERG, Y. – DAS, D. – GANCHEV, K. – HALL, K. – PETROV, S. – ZHANG, H. – TÄCKSTRÖM, O. – BEDINI, C. – BERTOMEU CASTELLÓ, N. – LEE, J. Universal Dependency Annotation for Multilingual Parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, p. 92–97, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. Available at: https://aclanthology.org/P13-2017.

MENG, K. – BAU, D. – ANDONIAN, A. – BELINKOV, Y. Locating and Editing Factual Associations in GPT. In KOYEJO, S. – MOHAMED, S. – AGARWAL, A. – BELGRAVE, D. – CHO, K. – OH, A. (Ed.) *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. Available at: http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Meng, K. – Sharma, A. S. – Andonian, A. J. – Belinkov, Y. – Bau, D. Mass-Editing Memory in a Transformer. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. Available at: https://openreview.net/pdf?id=MkbcAHIYgyS.

Merullo, J. – Eickhoff, C. – Pavlick, E. Language Models Implement Simple Word2Vec-style Vector Arithmetic. *CoRR*. 2023, abs/2305.16130. doi: 10.48550/ARXIV.2305.16130. Available at: https://doi.org/10.48550/arXiv.2305.16130.

Mielke, S. J. – Alyafeai, Z. – Salesky, E. – Raffel, C. – Dey, M. – Gallé, M. – Raja, A. – Si, C. – Lee, W. Y. – Sagot, B. – Tan, S. Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP. *CoRR*. 2021, abs/2112.10508. Available at: https://arxiv.org/abs/2112.10508.

Mihaylov, T. – Clark, P. – Khot, T. – Sabharwal, A. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. Available at: https://aclanthology.org/D18-1260.

Mikolov, T. Using Neural Networks for Modeling and Representing Natural Languages. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Tutorial Abstracts*, p. 3–4, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. Available at: https://aclanthology.org/C14-3002.

Mikolov, T. – Chen, K. – Corrado, G. – Dean, J. Efficient Estimation of Word Representations in Vector Space. In Bengio, Y. – LeCun, Y. (Ed.) *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. Available at: http://arxiv.org/abs/1301.3781.

Miller, G. A. WORDNET: a Lexical Database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, USA, February 23-26, 1992*. Morgan Kaufmann, 1992. Available at: https://aclanthology.org/H92-1116/.

Miller, G. A. – Leacock, C. – Tengi, R. – Bunker, R. T. A Semantic Concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*, 1993. Available at: https://aclanthology.org/H93-1061.

Musil, T. Examining Structure of Word Embeddings with PCA. In Ekstein, K. (Ed.) *Text, Speech, and Dialogue - 22nd International Conference, TSD 2019, Ljubljana, Slovenia, September 11-13, 2019, Proceedings*, 11697 / *Lecture Notes in Computer Science*, p. 211–223. Springer, 2019. doi: 10.1007/978-3-030-27947-9_18. Available at: `https://doi.org/10.1007/978-3-030-27947-9_18`.

Nadeem, M. – Bethke, A. – Reddy, S. StereoSet: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, p. 5356–5371, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.416. Available at: `https://aclanthology.org/2021.acl-long.416`.

Nangia, N. – Vania, C. – Bhalerao, R. – Bowman, S. R. CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1953–1967, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.154. Available at: `https://aclanthology.org/2020.emnlp-main.154`.

Nangia, N. – Vania, C. – Bhalerao, R. – Bowman, S. R. CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models. In Webber, B. – Cohn, T. – He, Y. – Liu, Y. (Ed.) *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, p. 1953–1967. Association for Computational Linguistics, 2020b. doi: 10.18653/V1/2020.EMNLP-MAIN.154. Available at: `https://doi.org/10.18653/v1/2020.emnlp-main.154`.

Navigli, R. – Conia, S. – Ross, B. Biases in Large Language Models: Origins, Inventory, and Discussion. *ACM J. Data Inf. Qual.* 2023, 15, 2, p. 10:1–10:21. doi: 10.1145/3597307. Available at: `https://doi.org/10.1145/3597307`.

Nivre, J. – Marneffe, M.-C. – Ginter, F. – Hajič, J. – Manning, C. D. – Pyysalo, S. – Schuster, S. – Tyers, F. – Zeman, D. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, p. 4034–4043, Marseille, France, May 2020. European Language Resources Association. Available at: `https://aclanthology.org/2020.lrec-1.497`. ISBN 979-10-95546-34-4.

OpenAI. GPT-4 Technical Report. *CoRR*. 2023, abs/2303.08774. doi: 10.48550/ARXIV.2303.08774. Available at: `https://doi.org/10.48550/arXiv.2303.08774`.

Palmer, M. – Gildea, D. – Kingsbury, P. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*. 2005, 31, 1, p. 71–106. doi: 10.1162/0891201053630264. Available at: `https://aclanthology.org/J05-1004`.

Pan, X. – Zhang, B. – May, J. – Nothman, J. – Knight, K. – Ji, H. Cross-lingual Name Tagging and Linking for 282 Languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 1946–1958, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1178. Available at: `https://aclanthology.org/P17-1178`.

Patil, V. – Talukdar, P. – Sarawagi, S. Overlap-based Vocabulary Generation Improves Cross-lingual Transfer Among Related Languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 219–233, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.18. Available at: `https://aclanthology.org/2022.acl-long.18`.

Pearl, J. *Causality*. Cambridge University Press, 2009. Available at: `https://books.google.cz/books?id=LLkhAwAAQBAJ`. ISBN 9781139643986.

Pearl, J. Direct and Indirect Effects. In Breese, J. S. – Koller, D. (Ed.) *UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, University of Washington, Seattle, Washington, USA, August 2-5, 2001*, p. 411–420. Morgan Kaufmann, 2001. Available at: `https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=126&proceeding_id=17`.

Pennington, J. – Socher, R. – Manning, C. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. Available at: `https://aclanthology.org/D14-1162`.

Perfetti, C. A. – Liu, Y. Orthography to Phonology and Meaning: Comparisons Across and within Writing Systems. *Reading and Writing*. 2005, 18, p. 193–210. Available at: `https://api.semanticscholar.org/CorpusID:9613632`.

Peters, M. E. – Neumann, M. – Iyyer, M. – Gardner, M. – Clark, C. – Lee, K. – Zettlemoyer, L. 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)*, p. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. Available at: `https://aclanthology.org/N18-1202`.

Petrov, S. – Das, D. – McDonald, R. A Universal Part-of-Speech Tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, p. 2089–2096, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). Available at: `http://www.lrec-conf.org/proceedings/lrec2012/pdf/274_Paper.pdf`.

Pimentel, T. – Saphra, N. – Williams, A. – Cotterell, R. Pareto Probing: Trading Off Accuracy for Complexity. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 3138–3153, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.254. Available at: `https://aclanthology.org/2020.emnlp-main.254`.

Pires, T. – Schlinger, E. – Garrette, D. How Multilingual is Multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1493. Available at: `https://aclanthology.org/P19-1493`.

Pires, T. – Lopes, A. V. – Assogba, Y. – Setiawan, H. One Wide Feedforward Is All You Need. In Koehn, P. – Haddon, B. – Kocmi, T. – Monz, C. (Ed.) *Proceedings of the Eighth Conference on Machine Translation, WMT 2023, Singapore, December 6-7, 2023*, p. 1031–1044. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.WMT-1.98. Available at: `https://doi.org/10.18653/v1/2023.wmt-1.98`.

Popel, M. – Žabokrtský, Z. – Vojtek, M. Udapi: Universal API for Universal Dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, p. 96–101, Gothenburg, Sweden, May 2017. Association for Computational Linguistics. Available at: `https://aclanthology.org/W17-0412`.

Press, O. – Wolf, L. Using the Output Embedding to Improve Language Models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, p. 157–163, Valencia, Spain, April 2017. Association for Computational Linguistics. Available at: https://aclanthology.org/E17-2025.

Press, O. – Smith, N. A. – Lewis, M. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. Available at: https://openreview.net/forum?id=R8sQPpGCv0.

Radford, A. – Narasimhan, K. Improving Language Understanding by Generative Pre-Training. 2018. Available at: https://api.semanticscholar.org/CorpusID:49313245.

Radford, A. – Wu, J. – Child, R. – Luan, D. – Amodei, D. – Sutskever, I. Language Models are Unsupervised Multitask Learners. 2019.

Raffel, C. – Shazeer, N. – Roberts, A. – Lee, K. – Narang, S. – Matena, M. – Zhou, Y. – Li, W. – Liu, P. J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 2020, 21, p. 140:1–140:67. Available at: http://jmlr.org/papers/v21/20-074.html.

Raganato, A. – Tiedemann, J. An Analysis of Encoder Representations in Transformer-Based Machine Translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, p. 287–297, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5431. Available at: https://aclanthology.org/W18-5431.

Rogers, A. – Kovaleva, O. – Rumshisky, A. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*. 2020, 8, p. 842–866. doi: 10.1162/tacl_a_00349. Available at: https://aclanthology.org/2020.tacl-1.54.

Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* 2019, 1, 5, p. 206–215. doi: 10.1038/S42256-019-0048-X. Available at: https://doi.org/10.1038/s42256-019-0048-x.

Rudinger, R. – Naradowsky, J. – Leonard, B. – Van Durme, B. Gender Bias in Coreference Resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, p. 8–14, New Orleans, Louisiana, June 2018a. Association for Computational Linguistics. doi: 10.18653/v1/N18-2002. Available at: https://aclanthology.org/N18-2002.

Rudinger, R. – Teichert, A. – Culkin, R. – Zhang, S. – Van Durme, B. Neural-Davidsonian Semantic Proto-role Labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 944–955, Brussels, Belgium, October-November 2018b. Association for Computational Linguistics. doi: 10.18653/v1/D18-1114. Available at: https://aclanthology.org/D18-1114.

Rust, P. – Pfeiffer, J. – Vulić, I. – Ruder, S. – Gurevych, I. How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, p. 3118–3135, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.243. Available at: https://aclanthology.org/2021.acl-long.243.

Sennrich, R. – Haddow, B. – Birch, A. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. Available at: https://aclanthology.org/P16-1162.

Silveira, N. – Dozat, T. – Marneffe, M.-C. – Bowman, S. – Connor, M. – Bauer, J. – Manning, C. A Gold Standard Dependency Corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, p. 2897–2904, Reykjavik, Iceland, May 2014a. European Language Resources Association (ELRA). Available at: http://www.lrec-conf.org/proceedings/lrec2014/pdf/1089_Paper.pdf.

Silveira, N. – Dozat, T. – Marneffe, M.-C. – Bowman, S. – Connor, M. – Bauer, J. – Manning, C. D. A Gold Standard Dependency Corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, 2014b.

Søgaard, A. – Ruder, S. – Vulić, I. On the Limitations of Unsupervised Bilingual Dictionary Induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 778–788, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1072. Available at: `https://aclanthology.org/P18-1072`.

Song, X. – Salcianu, A. – Song, Y. – Dopson, D. – Zhou, D. Fast WordPiece Tokenization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 2089–2103, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.160. Available at: `https://aclanthology.org/2021.emnlp-main.160`.

Stanczak, K. – Augenstein, I. A Survey on Gender Bias in Natural Language Processing. *CoRR*. 2021, abs/2112.14168. Available at: `https://arxiv.org/abs/2112.14168`.

Stanovsky, G. – Smith, N. A. – Zettlemoyer, L. Evaluating Gender Bias in Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 1679–1684, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1164. Available at: `https://aclanthology.org/P19-1164`.

Student. The probable error of a mean. *Biometrika*. 1908, p. 1–25.

Su, J. – Ahmed, M. H. M. – Lu, Y. – Pan, S. – Bo, W. – Liu, Y. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*. 2024, 568, p. 127063. doi: 10.1016/J.NEUCOM.2023.127063. Available at: `https://doi.org/10.1016/j.neucom.2023.127063`.

Sulubacak, U. – Gokirmak, M. – Tyers, F. – Çöltekin, Ç. – Nivre, J. – Eryiğit, G. Universal Dependencies for Turkish. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, p. 3444–3454, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. Available at: `https://aclanthology.org/C16-1325`.

Teichert, A. R. – Poliak, A. – Durme, B. V. – Gormley, M. R. Semantic Proto-Role Labeling. In Singh, S. – Markovitch, S. (Ed.) *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, p. 4459–4466. AAAI Press, 2017. doi: 10.1609/AAAI.V31I1.11165. Available at: `https://doi.org/10.1609/aaai.v31i1.11165`.

Tenney, I. – Das, D. – Pavlick, E. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. Available at: `https://aclanthology.org/P19-1452`.

Tibshirani, R. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*. 1996, 58, 1, p. 267–288. doi: https://doi.org/10.1111/j.2517-6161.1996.tb02080.x. Available at: `https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1996.tb02080.x`.

Tjong Kim Sang, E. F. – De Meulder, F. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, p. 142–147, 2003. Available at: `https://aclanthology.org/W03-0419`.

Touvron, H. – Lavril, T. – Izacard, G. – Martinet, X. – Lachaux, M. – Lacroix, T. – Rozière, B. – Goyal, N. – Hambro, E. – Azhar, F. – Rodriguez, A. – Joulin, A. – Grave, E. – Lample, G. LLaMA: Open and Efficient Foundation Language Models. *CoRR*. 2023, abs/2302.13971. doi: 10.48550/ARXIV.2302.13971. Available at: `https://doi.org/10.48550/arXiv.2302.13971`.

Van Der Wal, O. – Jumelet, J. – Schulz, K. – Zuidema, W. The Birth of Bias: A case study on the evolution of gender bias in an English language model. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, p. 75–75, Seattle, Washington, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.gebnlp-1.8. Available at: `https://aclanthology.org/2022.gebnlp-1.8`.

Wal, O. – Bachmann, D. – Leidinger, A. – Maanen, L. – Zuidema, W. H. – Schulz, K. Undesirable Biases in NLP: Addressing Challenges of Measurement. *J. Artif. Intell. Res.* 2024, 79, p. 1–40. doi: 10.1613/JAIR.1.15195. Available at: `https://doi.org/10.1613/jair.1.15195`.

VanderWeele, T. *Explanation in causal inference: methods for mediation and interaction.* Oxford University Press, 2015.

Vaswani, A. – Shazeer, N. – Parmar, N. – Uszkoreit, J. – Jones, L. – Gomez, A. N. – Kaiser, L. – Polosukhin, I. Attention is All you Need. In Guyon, I. – Luxburg, U. – Bengio, S. – Wallach, H. M. – Fergus, R. – Vishwanathan, S. V. N. – Garnett, R. (Ed.) *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, p. 5998–6008, 2017. Available at: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Vidra, J. – Žabokrtský, Z. – Ševčíková, M. – Kyjánek, L. DeriNet 2.0: Towards an All-in-One Word-Formation Resource. In *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, p. 81–89, Prague, Czechia, September 2019. Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics. Available at: https://aclanthology.org/W19-8510.

Vig, J. – Belinkov, Y. Analyzing the Structure of Attention in a Transformer Language Model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, p. 63–76, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4808. Available at: https://aclanthology.org/W19-4808.

Vig, J. – Gehrmann, S. – Belinkov, Y. – Qian, S. – Nevo, D. – Singer, Y. – Shieber, S. M. Causal Mediation Analysis for Interpreting Neural NLP: The Case of Gender Bias. *CoRR*. 2020, abs/2004.12265. Available at: https://arxiv.org/abs/2004.12265.

Voita, E. – Talbot, D. – Moiseev, F. – Sennrich, R. – Titov, I. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 5797–5808, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. Available at: https://aclanthology.org/P19-1580.

Vulić, I. – Ruder, S. – Søgaard, A. Are All Good Word Vector Spaces Isomorphic? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 3178–3192, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.257. Available at: https://aclanthology.org/2020.emnlp-main.257.

Wang, Y. – Che, W. – Guo, J. – Liu, Y. – Liu, T. Cross-Lingual BERT Transformation for Zero-Shot Dependency Parsing. 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)*, p. 5721–5727, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1575. Available at: `https://aclanthology.org/D19-1575`.

Webster, K. – Costa-jussà, M. R. – Hardmeier, C. – Radford, W. Gendered Ambiguous Pronoun (GAP) Shared Task at the Gender Bias in NLP Workshop 2019. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, p. 1–7, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3801. Available at: `https://aclanthology.org/W19-3801`.

Weischedel, R. M. – Hovy, E. H. – Marcus, M. P. – Palmer, M. OntoNotes : A Large Training Corpus for Enhanced Processing. 2017. Available at: `https://api.semanticscholar.org/CorpusID:204845447`.

Williams, A. – Nangia, N. – Bowman, S. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, p. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. Available at: `https://aclanthology.org/N18-1101`.

Wolf, T. et al. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, p. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. Available at: `https://aclanthology.org/2020.emnlp-demos.6`.

Workshop, B. et al. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model, 2023.

Wu, S. – Dredze, M. Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness 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)*, p. 833–844, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1077. Available at: `https://aclanthology.org/D19-1077`.

Wu, S. – Dredze, M. Do Explicit Alignments Robustly Improve Multilingual Encoders? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 4471–4482, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.362. Available at: https://aclanthology.org/2020.emnlp-main.362.

Wu, Z. – Nguyen, T.-S. – Ong, D. Structured Self-AttentionWeights Encode Semantics in Sentiment Analysis. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, p. 255–264, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.24. Available at: https://aclanthology.org/2020.blackboxnlp-1.24.

Xu, H. – Sharaf, A. – Chen, Y. – Tan, W. – Shen, L. – Durme, B. V. – Murray, K. – Kim, Y. J. Contrastive Preference Optimization: Pushing the Boundaries of LLM Performance in Machine Translation. *CoRR*. 2024, abs/2401.08417. doi: 10.48550/ARXIV.2401.08417. Available at: https://doi.org/10.48550/arXiv.2401.08417.

Xue, L. – Constant, N. – Roberts, A. – Kale, M. – Al-Rfou, R. – Siddhant, A. – Barua, A. – Raffel, C. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 483–498, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.41. Available at: https://aclanthology.org/2021.naacl-main.41.

Xue, L. – Barua, A. – Constant, N. – Al-Rfou, R. – Narang, S. – Kale, M. – Roberts, A. – Raffel, C. ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models. *Transactions of the Association for Computational Linguistics*. 2022, 10, p. 291–306. doi: 10.1162/tacl_a_00461. Available at: https://aclanthology.org/2022.tacl-1.17.

Zeman, D. Reusable Tagset Conversion Using Tagset Drivers. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). Available at: http://www.lrec-conf.org/proceedings/lrec2008/pdf/66_paper.pdf.

Zhao, J. – Wang, T. – Yatskar, M. – Ordonez, V. – Chang, K.-W. Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, p. 15–20, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2003. Available at: `https://aclanthology.org/N18-2003`.

Terms and Abbervations

NLP natural language processing. 1, 17, 23, 25, 87, 91, 101, 106, 108, 109

LM language model. 1, 2, 3, 4, 7, 11, 13, 15, 17, 19, 20, 27, 34, 57, 58, 61, 71, 73, 76, 80, 92, 93, 98, 107, 109, 147

Transformer Neural network architecture based on the attention mechanism.. 1, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 79, 81, 101, 102, 103, 104, 105, 109, 147, 149, 150

FF feed-forward. 4, 9, 10, 12, 15, 57, 60, 61, 62, 63, 66, 69, 70, 71, 72, 103, 148

Att. attention. 4, 8, 9, 10, 11, 12, 13, 15, 60, 61, 69, 70, 73, 74, 75, 76, 82, 86, 103, 148

$\vec{x}$ Vector of neural network inputs.. 7

$\vec{y}$ Vector of neural network outputs.. 7

$\vec{h}$ Vector of latent representations (input of the layer).. 7, 8, 9

$\vec{h}'$ Vector of latent representations (output of the layer).. 7, 9

SM softmax. 8, 11, 12

attention head one layer implementing attention mechanism.. 8, 58, 73, 77, 79, 80, 82, 83, 86

LLM large language model. 9, 104

LN layer normalization. 10

GPT Generative Pre-trained Transformer. 12, 15, 16, 101

BART Bidirectional and Auto-Regressive Transformers. 13

T5 Text-to-Text Transfer Transformer. 14

MLM masked language model. 14, 15, 53, 54, 97, 143

BERT Bidirectional Encoder Representations from Transformers. 14, 34, 53, 73, 74, 75, 76, 80, 81, 82, 101, 103, 105, 108

RoBERTa Robustly optimized BERT approach. 14, 15, 88

XLM Cross-lingual Language Model. 15

XLM-R XLM-RoBERTa. 15, 87, 88, 90, 103

LLaMA Large Language Model Meta AI. 16, 57, 60, 61, 62, 65, 66, 67, 68, 69, 70, 71, 72, 107, 144, 148

MRR mean reciprocal rank. 19, 92, 97, 98, 99

GATS Google Analogy Test Set. 21, 22, 27

POS part of speech. 21, 22, 23, 33, 34, 39, 85, 92, 95, 96, 97, 98, 99, 100, 108, 149

UAS unlabeled attachment score. 22, 50, 79, 80, 81, 102, 144, 149

LAS labeled attachment score. 22, 79, 80, 81, 144

UD universal dependencies. 22, 39, 74, 75, 76, 78, 79, 80, 82, 86, 92, 108, 110, 144

NER named entity recognition. 23, 33, 34, 92, 95, 96, 97, 98, 99, 100, 149

WSC Winograd Schema Challenge. 24, 29

NLI natural language inference. 25, 95, 99, 149

XNLI cross-lingual natural language inference. 25, 92, 99, 110

QA question answering. 25

OBQA OpenBookQA. 25, 66, 110

NQ Natural Questions. 25

ARC AI2 Reasoning Challenge. 26, 66, 110

MMLU Measuring Massive Multitask Language Understanding. 26, 66, 110

WEAT Word Embedding Association Test. 27, 28

StereoS StereoSet. 28, 66, 68, 110

LMS Language Modeling Score. 28

SS Sterotype Score. 28

ICAT Idealized Context Association Test. 28

WB WinoBias. 29, 66

WG WinoGender. 29

WM WinoMT. 29, 30

orthogonal probe a linear probe with orthogonality constraint enabling mapping mutually orthogonal dimensions to a specific probed feature.. 33, 36, 37, 38, 39, 40, 46, 49, 50, 52, 54, 55, 107, 108

SRL semantic role labeling. 34

structural probe a linear probe optimized to predict a structure encoded in the neural network, i.e. hypernymy tree.. 34, 36, 39, 40, 46, 47, 143

SVD singular value decomposition. 36

scaling vector a loading vector used in orthogonal probes, denotes the importance of a specific feature for a given task.. 36, 37, 38, 39, 40, 41, 42, 45, 46, 52, 55, 143

orthogonal transformation a linear orthogonal transformation used in orthogonal probes.. 46, 50, 52, 55

orthogonal filter a filter based on orthogonal probes, mitigates the latent representation of a specific feature by nullifying the corresponding dimensions.. 53, 55

causal tracing a method to identify the components of the network encoding signals that affect its function.. 57, 58, 59, 61, 69, 70, 71, 104, 107, 110, 148

causal mediation analysis a method identifying potential pathways that could explain observed associations between an exposure and an outcome.. 57, 58, 147

MLP multi-layer perceptron. 58, 61, 62, 63, 64, 67, 68

DAMA Debiasing through Model Adapatation. 61, 62, 63, 65, 66, 67, 68, 69, 71, 72, 107, 144, 148

PLS partial least squares. 63

DepAl dependency alignment. 73

DepAcc dependency accuracy. 74, 76, 77, 81

GSD Google Universal Dependency Treebank. 74

SUD surface-syntactic universal dependencies. 75

head ensemble a set of attention heads aligned with a linguistic structure.. 77, 80, 81, 85, 86, 148

SOV subject-object-verb. 81

SVO subject-verb-object. 81

OOV Out-of-vocabulary. 88

vocabulary allocation the allocation of tokens for representing a specific language or domain.. 88, 90, 93, 95, 98, 99, 100, 104, 144, 145, 149

vocabulary overlap the similarity between two token vocabularies.. 88, 91, 93, 97, 98, 99, 100, 145, 149

AR average rank. 90, 93, 97, 98, 99, 100

CPT characters per token. 91, 93, 98, 100

JSD Jensen-Shanon divergence. 91, 92, 93, 94, 95, 98, 99, 145, 148

List of Tables

List of Figures