Master's Thesis

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Arts in Computational Linguistics

---

Understanding Morphosyntactic Representations in Pretrained Language Models

---

Author
Matteo Brivio

Supervisors
Dr. Çağrı Çöltekin
Prof. Dr. Gerhard Jäger

Seminar für Sprachwissenschaft
Eberhard Karls Universität Tübingen

5th December 2023

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst habe, dass ich keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe, dass ich alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe, dass die Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist, dass ich die Arbeit weder vollständig noch in wesentlichen Teilen bereits veröffentlicht habe, dass das in Dateiform eingereichte Exemplar mit den eingereichten gebundenen Exemplaren übereinstimmt.

Tübingen, 5 Dezember 2023

Matteo Brivio

Abstract

This thesis investigates whether Pretrained Language Models (PLMs) are able to encode morphosyntactic information. A novel approach is introduced, using neural probes to map a PLM's contextual representation to a target representation encoding the linguistic information of interest. The focus of this analysis lies on three PLMs architectures – BERT, RoBERTa and Sentence Transformer – each differing in their implementations and pretraining strategies. These architectures are probed for their ability to encode two morphosyntactic features, tense and subject-verb agreement, in three languages with varying degrees of morphological richness: English, Italian and German. The analysis also explores whether the implementation and pretraining objective of a PLM affect its ability to encode linguistic information, and whether one linguistic phenomenon is represented better than the other. The results are not entirely conclusive, but show a promising trend. Overall, PLMs appear to be likely capable of encoding morphosyntactic information. The specific characteristics of a PLM appear to have some influence on how it represents information in its internal states. Notably, agreement seems to be better captured than tense by the three PLMs. These findings open up new avenues for further exploration.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The field of Natural Language Processing (NLP) has witnessed significant advances with the emergence of Pretrained Language Models (PLMs) based on the Transformer architecture (Vaswani et al., 2017). These models can learn from large amounts of text data and generate dense, low-dimensional contextual representations that enhance the performance of many downstream tasks. As a consequence, these representations have progressively replaced more traditional and understandable features as input to machine learning models, resulting in state-of-the-art performance in many NLP benchmarks such as GLUE (Wang et al., 2018) and SQuAD (Rajpurkar et al., 2016, 2018).

However, despite their effectiveness, these models are often referred to as "black boxes" due to their considerable complexity, which appears to be motivated, at least in part, by a correlation between performance and scale. In this respect, Kaplan et al. (2020) observe that language modeling performance improves smoothly as the model size, dataset size and amount of compute used for training are increased. Consequently, concerns about safe and responsible use of PLMs are growing, largely driven by their lack of interpretability.

While lacking transparency, there is evidence that these models may be able to capture useful, transferable linguistic features. However, it remains unclear and poorly understood whether, where, and in what measure this information is encoded by their internal representations. These questions have fueled a growing interest in interpretable machine learning, contributing to this research trend with several approaches, and leading to the establishment of dedicated venues and workshops, such as BlackboxNLP (Bastings et al., 2022). In this regard, probing has emerged as a prominent approach for interpreting and analyzing the information captured by PLMs.

Behavioral and diagnostic probing are among the most widely used methods to assess the linguistic knowledge and capabilities of these models. However, both approaches face some major challenges. The diagnostic probing paradigm has been questioned with respect to the lack of comparative baselines to evaluate a probe's performance, as well as the impact of a probe's architecture on the final result (Belinkov, 2022). Behavioral probing, on the other hand, relies on curated evaluation datasets to draw conclusions on a model as a whole, but fail to provide insights about its individual

components (Lasri et al., 2022).

This thesis aims to shed some light on the inner workings of PLMs and considers three specific architectures: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and Sentence Transformers (Reimers and Gurevych, 2019). The analysis of these models is guided by three research questions.

The first question investigates whether PLMs encode any morphosyntactic information at all, relying on a novel probing method that attempts to address the main limitations of the existing strategies. The presented solution uses neural probes trained on carefully designed datasets to map contextual representations to specific morphosyntactic features: tense and subject-verb agreement. These linguistic phenomena are explored with respect to English, Italian and German, three languages that exhibit varying degrees of morphological richness. Both the proposed probing approach and the datasets represent major contributions of this thesis and are described in chapter 3 and chapter 4, respectively.
The second research question aims to explore whether architectural choices and pretraining strategies have any effect on the way linguistic information is encoded by PLMs.
Finally, the third question considers whether certain morphosyntactic relations are encoded better than others. Experimental results are presented and discussed in chapter 5 and chapter 6, respectively.

The code, data and results for all experiments can be found at `https://github.com/matteobrv/ma_thesis`.

Chapter 2

Background

2.1 Distributional Semantic Models

While natural language is a valuable and rich source of information, it is also discrete, sparse, and ambiguous, making it a challenging input for computational applications. Therefore, for algorithms to operate on spoken and written language, with the latter being the focus of this work, linguistic data must first be transformed into a suitable machine-readable format capable of encoding its meaning.

Over the years, several representation strategies have been developed that rely on statistical measures to model the meanings of lexical items as real-valued vectors or matrices. Most of these methods are ultimately grounded in the Distributional Hypothesis (Harris, 1954), the idea that words occurring in similar contexts tend to have similar meanings or, in other terms, that the meaning of a linguistic expression depends at least in part on its distributional properties, i.e. the linguistic contexts in which it appears (Lenci and Littell, 2008). Firth (1957) later embraced and expanded upon this concept, with his remark "You shall know a word by the company it keeps" turning into an adage of computational linguistics, as a testament of the enduring relevance of this idea.

The Distributional Hypothesis, together with the intuition that lexical items can be mapped to points in a multidimensional space (Osgoodand et al., 1957), contributed to the development of Distributional or Vector Space Semantics, a statistical and data-driven framework that induces semantic representations from contexts of use (Boleda, 2020).

Boleda and Herbelot (2016) describe distributional representations as either vectors or more complex algebraic structures that encode abstractions on the contexts of use observed in large amounts of natural language data. These representations may have up to several hundreds of dimensions, and the semantic information is distributed across all of them, encoded in the form of continuous numerical values. A collection of these objects constitutes a vector or semantic space. In such a space, representations can be manipulated and semantic relations quantified. This can be achieved using well-defined algebraic techniques and measures, such as Cosine Similarity and Euclidean Distance.

Models that learn these representations are referred to as Distributional Semantic

Models (DSMs) and according to Lenci et al. (2022) can be grouped into three different macro-categories: count-based DSMs, prediction DSMs and contextual DSMs.

2.1.1 Count-based Models

The origins of count-based DSMs can be traced back to the Vector Space Model (VSM), a well-known representation method proposed by Salton et al. (1975) in the context of Information Retrieval. The VSM was originally developed to score a set of documents on a query and retrieve the most relevant ones. The model works by encoding both the query and the documents as vectors of features $\mathbf{v} = (t_1, \ldots, t_i, \ldots, t_{|V|})$, where t_i corresponds to the weighted frequency of the ith vocabulary term and $|V|$ is the size of the vocabulary obtained from the set of documents being considered. The documents are ranked by their semantic similarity to the query and presented to the user in a descending order.

The VSM is one of the earliest and most prominent text representation models and is fundamental to a number of information retrieval tasks, such as document clustering and document classification (Manning et al., 2008). Both Information Retrieval and the intuition behind the VSM[1] profoundly affected the development of Distributional Semantics as a whole and paved the way to a variety of count-based DSMs (Clark, 2015). To implement these models, the first step is to construct a co-occurrence term-context matrix $\mathbf{M}$ in which each row represents a lexical item and each column the context in which it appears. Each matrix entry then corresponds to some co-occurrence measure between the lexical item and the context. This can either be a raw frequency value, specifying how many times an item appears in a given context, or some weighted measure. Examples of common measures include Term Frequency - Inverse Document Frequency and Positive Pointwise Mutual Information. These measures reflect the fact that some contexts are more indicative of a term's meaning than others (Clark, 2015).

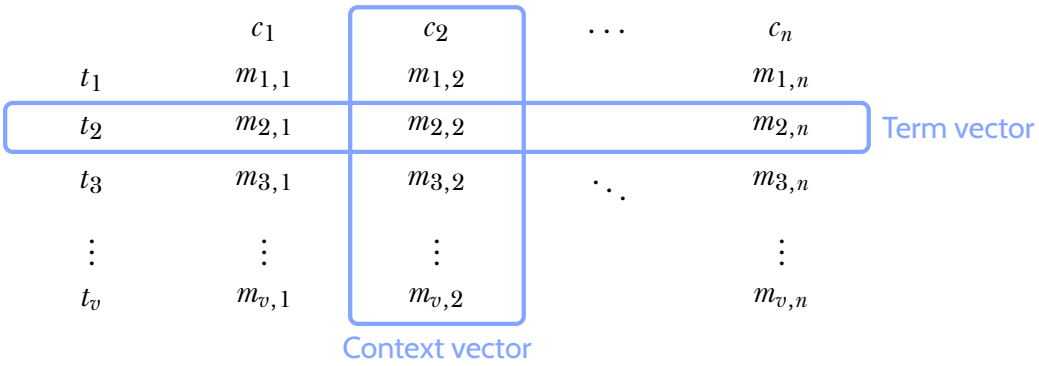

Figure 1: A co-occurrence term-context matrix is a $v \times n$ structure that allows to represent both terms and contexts as row and column vectors, respectively. Each term vector contains n elements, one for every context, while each context vector contains v elements, one for every term.

[1]Turney and Pantel (2010) present an extensive literature review of VSMs, illustrating their relation with the Distributional Hypothesis and their diverse applications in semantics.

Co-occurrence matrices vary in how their linguistic contexts are defined, allowing for meaning and term similarity to be encoded and measured at different levels of granularity. Therefore, the notion of context is not limited to entire documents, as it was the case in the original VSM implementation, but can be narrowed down to paragraphs, sentences or even single terms to capture more fine-grained relations. Nonetheless, regardless of the context, vector representations in co-occurrence matrices are high-dimensional and sparse, i.e. most of the matrix entries are zeroes. As observed by Dhillon and Modha (2001), a few thousand dimensions and a sparsity above 90% are typical.

To mitigate these issues, limit computational complexity and improve the quality of the representations by increasing their generalization power, dimensionality reduction techniques are commonly applied. The core idea is to identify redundancies and correlations between linguistic contexts in the original matrix $\mathbf{M} \in \mathbb{R}^n$ and to transform it into a meaningful representation of reduced intrinsic dimensionality $\mathbf{Y} \in \mathbb{R}^l$, such that $l \ll n$ (van der Maaten et al., 2009). In other words, context features that account only for little variability are discarded so that a large part of the variability in the data can be explained with lower-dimensional representations (Günther et al., 2019). Target items are thus represented in a latent semantic space of dense vectors.

Latent Semantic Analysis (LSA) (Deerwester et al., 1990; Landauer and Dumais, 1997) is a prominent example of a count-based DSM that relies on dimensionality reduction to generate the best k-dimensional approximation of the original co-occurrence matrix $\mathbf{M}$. This is achieved through Singular Value Decomposition (SVD), which factorizes $\mathbf{M}$ into three matrices:

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \tag{1}$$

In short, the diagonal matrix $\mathbf{\Sigma}$ contains the singular values of $\mathbf{M}$ arranged in decreasing order and by retaining only the k largest values it is possible to produce the truncated matrix $\hat{\mathbf{M}}$:

$$\hat{\mathbf{M}} = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^\top. \tag{2}$$

$\hat{\mathbf{M}}$ is the best reduced-rank matrix approximation of $\mathbf{M}$, such that the sum of the squared differences between their respective elements, or the Frobenius norm of the two matrices, is minimized.

However, not all count-based DSMs rely upon dimensionality reduction methods. As an example, instead of collecting co-occurrence statistics into a matrix and then optionally reduce them, random encoding models (Kanerva et al., 2000; Sahlgren, 2005) directly learn dense representations by assigning a random vector to each lexical item and incrementally updating it based on co-occurring contexts.

2.1.2 Prediction Models

The resurgence of neural networks and the rise of deep learning led to a new generation of so-called prediction models that largely overshadowed the more traditional

count-based ones (Lenci et al., 2022).

Prediction DSMs are neural networks that learn dense, low-dimensional vector representations by being trained on a prediction task, typically language modeling which consists in retrieving a specific term given its context, or vice-versa. The resulting vector representations are commonly referred to as *word embeddings* or *neural embeddings*.

Bengio et al. (2003) introduce the first instance of neural language model which, given a context of k-gram terms $w_{1:k}$, tries to predict the ith term w_{k+1} in a sequence. A matrix of free parameters $\mathbf{C} \in \mathbb{R}^{|V| \times m}$, where $|V|$ corresponds to the number of lexical items, maps each term to an m-dimensional vector. The parameters are progressively tuned during training and word embeddings are generated as a by-product of the prediction task. Notably, Westera and Boleda (2019) observe that word embeddings emerge not only from models explicitly designed to produce such representations, but rather from any neural network whose input is a set of terms that need to be encoded for a given task to be carried out.

Arguably, the quality of the representations varies according to a number of factors, including the model's architecture and the nature of the task (Goldberg, 2016). Nonetheless, the resulting learned representations have been proven to be transferable to downstream tasks (Liu et al., 2020). In this respect, `Word2Vec` (Mikolov et al., 2013a,b) and `FastText` (Bojanowski et al., 2017) are worth mentioning as two successful prediction DSMs able to generate high-quality word embeddings.

The architecture of `word2vec` is based on a feed-forward neural network language model and operates in a self-supervised fashion with two learning strategies: Continuous Bag-of-Words (CBOW) and Skip-gram. In essence, the CBOW model tries to predict a target term w_t given its context $w_{t-i}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+i}$, while the Skip-gram model is trained to predict the context from a given target. Regardless of the strategy, once the training is completed, the learned linear transformations in the hidden layer correspond to neural embeddings that can be used as input for other models. `FastText` improves upon the `Word2Vec` Skip-gram model by exploiting subword information. More precisely, the model learns representations for character n-grams and represents lexical items as the sum of their n-gram vectors. This extends the original Skip-gram model, allowing to take the internal structure of words into account, thereby generating more morphologically-aware representations.

A variety of linguistic regularities are believed to be captured by word embeddings. In particular Mikolov et al. (2013c) observe that syntactic and semantic regularities appear to be present as vector offsets in the learned embedding space of `Word2Vec`, such that words sharing a particular relation, like gender, are related by the same constant offset. This is demonstrated by a famous example where, the embedding for the word "queen" is recovered by simple vector arithmetic, using the representations for "king", "man" and "woman": queen = king − man + woman.

Compared to traditional count-based methods, prediction models are easier to train, generalize well and can efficiently scale up to process much larger datasets. This results in richer representations that seem to better capture semantic similarities among lexical items. However, even though prediction DSMs appear to consistently outperform more traditional methods by a substantial margin (Baroni et al., 2014),

the debate on their alleged superiority is not fully resolved. Specifically, the mathematical objective and the sources of information available to prediction DSMs are, in some instances, quite similar to those employed by the more traditional count-based methods (Levy and Goldberg, 2014) and, when properly tuned, the latter can be as good as the former, at least in most intrinsic tasks (Levy et al., 2015; Lenci et al., 2022). Additionally, several sources report a drop in performance for prediction DSMs when the training dataset is small (Sahlgren and Lenci, 2016; Altszyler et al., 2016).

While word embeddings seem to be very effective at capturing semantic and syntactic properties of words, they face a significant limitation which is referred to as the *meaning conflation deficiency* (Camacho-Collados and Pilehvar, 2018). Prediction DSMs generate type-level representations by merging various senses of a word into a single vector, resulting in *static word embeddings*. In simpler terms, "static" refers to the property that a vector representation remains constant regardless of the specific context in which a particular word appears.

2.1.3 Contextual Models

Contextual models go beyond word-level semantics and therefore radically depart from count-based and static prediction DSMs. Instead of learning a single vector representation per word type, countextual DSMs embed each lexical item in a vector that is a function of the whole input sequence, therefore generating inherently contextualized representations for each word token (Lenci et al., 2022). In other words, these models learn which terms are most closely related to the one that is being currently processed, thus encoding relevant aspects of its context in the output vector. Thanks to their representational power, *contextualized embeddings* have seen widespread adoption and substantially improved performance for virtually every NLP task.

Contextual DSMs can be classified according to different architectural features. In particular, it is common to distinguish between shallow and deep models, as well as between bidirectional and unidirectional ones. Liu et al. (2020) provide a detailed and comprehensive overview of the main contextual DSMs, which is beyond the scope of this thesis. Instead, this work focuses on deep bidirectional pretrained language models and inspect a subset thereof (subsection 2.2.1, subsection 2.2.2, subsection 2.2.3).

2.2 Pretrained Language Models

Contextual DSMs are commonly referred to as Ptrained Language Models (PLMs), where "pretraining" is understood as the process of learning meaning representations for words or sentences by processing very large amounts of text data (Jurafsky and Martin., 2023). Despite some exceptions, recent PLMs are built upon the Transformer architecture (Vaswani et al., 2017), a sequence-to-sequence model, dispensing with recurrent connections and instead relying on a scaled dot-product self-attention mechanism.

The key idea of every attention-based method is to compare a target item with a set

of other items to determine how relevant they are in the current context (Jurafsky and Martin., 2023). In the case of self-attention, each item in an input sequence attend to each other item within the same sequence. This is in contrast to other attention-based approaches, such as the sequence-to-sequence model proposed by Bahdanau et al. (2015) where the decoder attends to the hidden states of the encoder. More formally, given an input sequence of n tokens, each item x_i is assigned a set of attention weights $(\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,j}, \ldots, \alpha_{i,n})$ over all input tokens in the sequence, where $\alpha_{i,j}$ is a measure of how much x_i attends to x_j.

Briefly put, for each input token, self-attention relies on a set of three vectors: query $\mathbf{q} \in \mathbb{R}^k$, key $\mathbf{k} \in \mathbb{R}^k$ and value $\mathbf{v} \in \mathbb{R}^v$. These vectors are obtained by multiplying each input token representation with a set of learned weight matrices: $\mathbf{W}_q$, $\mathbf{W}_k$ and $\mathbf{W}_v$. Each value vector is then weighted by a scoring function f of the query with the corresponding key. Ultimately, the output is computed as a weighted sum of the value vectors.

$$f = softmax(\frac{\mathbf{q} \cdot \mathbf{k}}{\sqrt{d_k}}) \tag{3}$$

This mechanism, though implemented in a more efficient way, underlies all of the models considered in this thesis.

2.2.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) is a deeply bidirectional, unsupervised language model, consisting of a stack of 12 Transformer encoders.[2] The model is designed to learn deep bidirectional text representations by jointly conditioning on both the left and the right context in all of its layers. This is achieved by learning two self-supervised tasks in the pretraining phase: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP).

MLM is a fill-in-the-blank task inspired by the cloze test, a gap-filling readability assessment, first presented by Taylor (1953), in which a person scores one cloze unit for correctly closing the gap in a given text sequence with the correct item. The cloze test is adapted as a pretraining task that allows to overcome the drawbacks of unidirectional language modeling. Specifically, by replacing a percentage of the items in the input sequence with a special `[MASK]` token, the model is encouraged to generate bidirectional representations by guessing the masked tokens, relying on their left and the right context.

NSP trains the model to distinguish whether two input sentences follow one another or not. Specifically, given a set of sentence pairs, 50% of the time the second sentence actually follows the first one, while the remaining 50% of the time the second sentence does not and is randomly chosen from the training corpus. The rationale given for this task is that it could assist the model in sentence-level downstream tasks, such as Question Answering and Natural Language Inference.

[2] While this thesis considers BERT-base, a stack of 12 Transformer encoders, Devlin et al. (2019) also introduce BERT-large which consists of 24 Transformer layers.

BERT relies on a special [CLS] classification token to mark the beginning of each input sequence. This classification token is particularly relevant as its final hidden state is commonly used as a sentence-level representation.

2.2.2 RoBERTa

RoBERTa (Robustly optimized BERT approach) (Liu et al., 2019) is a BERT-based model that makes a few changes to the original BERT implementation, thereby achieving substantial performance improvements.

The most significant difference between the two models lies in the fact that RoBERTa drops the Next Sentence Prediction objective because, according to the authors, it contributes little to the model's performance and final representations. Additionally, the authors observe that BERT is significantly undertrained and, therefore, decide to increase the training time, batch size, amount of training data, and length of input sequences. Finally, RoBERTa applies a dynamic strategy to the Masked Language Modeling objective by generating a masking pattern for each input sequence at training time. This ensures that the same input sequence has a different pattern for each training epoch.

2.2.3 Sentence Transformers

In BERT-based models, the final representation of the [CLS] token is commonly used as a sentence-level embedding. This practice was introduced by Devlin et al. (2019) and subsequently embraced by others. However, despite being widespread, Reimers and Gurevych (2019) observe that [CLS] maps sentences to a vector space that is rather unsuitable to be used with common similarity and distance measures, ultimately yielding poor sentence representations.

To address this problem, Reimers and Gurevych (2019) introduce Sentence-BERT (SBERT), a BERT-based model that relies on siamese and triplet networks to generate effective sentence embeddings. Each network takes a sentence as input and produces a fixed-sized vector representation using a pooling strategy. While various methods have been explored, the default approach involves calculating the mean of all token representations. As each network generates a representation for a different sentence, a final fine-tuning step is required to update all of the models' parameters. This ensure that the embeddings are semantically meaningful and can be compared using the appropriate similarity metrics.

This framework offers a straightforward approach to calculate dense vector representations for sentences and can be applied to a range of Transformer-based models other than BERT.[3] Moreover, Reimers and Gurevych (2020) present a method for extending monolingual sentence embedding models to new languages, thus enabling the creation of multilingual sentence transformers.

[3] See https://www.sbert.net/ for a comprehensive list of Sentence Transformer models.

2.3 Interpretability

Historically, NLP primarily relied on inherently interpretable architectures, such as rule-based systems and decision trees, commonly referred to as "white-box" models. However, in recent years, popular benchmarks, such as GLUE (Wang et al., 2018) and SQuAD (Rajpurkar et al., 2016, 2018), have been dominated by ever larger and complex "black-box" models whose predictive power has steadily improved at the cost of interpretability. This trade-off between the quality of a model's prediction and its ability to produce explainable outputs seems to be motivated by a correlation between performance and scale. In this respect, Kaplan et al. (2020) show that language modeling performance improves smoothly as the model size, dataset size and amount of compute used for training are increased.

This lack of transparency in how machine learning models reach their conclusions can be problematic, as it may erode trust in the many Artificial Intelligence systems that people interact with on a daily basis (Danilevsky et al., 2020). Furthermore, interpretability is essential to enhance system robustness as well as to prevent bias, unfairness, and discrimination (Confalonieri et al., 2021).

2.3.1 Terminological Clarifications

Despite its importance, there is a lack of consensus surrounding the notion of interpretability within the field of machine learning. While some authors use the term interchangeably with explainability (Jacovi and Goldberg, 2020; Tjoa and Guan, 2021), others prefer to draw a line between them (Gilpin et al., 2018). As noted both by Linardatos et al. (2021) as well as Clinciu and Hastie (2019), a number of attempts have been made to clarify not only these two terms, but also related concepts such as transparency, comprehensibility and intelligibility, which are being used as synonyms leading to further confusion. However, the lack of an established standard for measuring interpretability, coupled with the fact that these notions often depend both on the domain of application as well as on the target explainee, make the task difficult (Marcinkevičs and Vogt, 2020).

As a thorough terminological clarification lies beyond the scope of this thesis, interpretability is regarded as a broader term encompassing explainability, in accordance with the taxonomy proposed by Graziani et al. (2023). Specifically, it is intended as the ability to explain a system's inner workings and outcomes in a human understandable manner.

2.3.2 Probing Linguistic Information

Explanations of machine learning systems can be categorized along two main dimensions (Danilevsky et al., 2020). The first dimension differentiates explanations for individual predictions (local) from those that address the overall prediction process of the model (global). The second dimension distinguishes between explanations that emerge naturally from the prediction process (self-explaining) and those that require any kind of subsequent post-processing (post-hoc), perhaps relying on an additional model.

In NLP a growing number of contributions focus on global explanation methods

which ultimately aim to uncover what kind of linguistic information is captured by Pretrained Language Models (PLMs). In particular, a number of experiments look at BERT and BERT-based models, contributing to a newly acknowledged research domain known as BERTology (Rogers et al., 2020).

Popular strategies to probe the knowledge encoded by a model include, among others, (1) cloze-like tasks, (2) attention weights analysis, (3) predicting linguistic properties from latent representations and (4) intervening in the model's input space, architecture or representation layers. These approaches are briefly outlined, but for a comprehensive analysis of this vast research area, the work by Belinkov and Glass (2019) is a valuable entry point.

Methods relying on the first strategy fall under a broader class of techniques commonly known as behavioral probes. These approaches typically rely on "challenge datasets" constructed to test the model's behavior with respect to specific linguistic phenomena, as well as to evaluate its robustness and ability to generalize (Madsen et al., 2022). As an example, Marvin and Linzen (2018) generate a dataset of grammatical and ungrammatical sentence pairs to test a model's ability to capture relevant aspects of grammar. The model is expected to assign a higher probability to the grammatical sentences than the ungrammatical ones. Linzen et al. (2016) investigates the ability of language models to learn subject-verb agreement. Lastly, Gulordava et al. (2018) test whether RNNs, trained with a language modeling objective, can predict long-distance number agreement.

Experiments that look at attentions weights in Transformer-based models are generally categorized as local self-explaining approaches, as they rely on information emitted directly by the model (Danilevsky et al., 2020). Rogers et al. (2020) provide a thorough survey of the major contributions following this route.

The third approach generates global, post-hoc explanations by relying on so-called probing or diagnostic classifiers. The underlying idea is deceptively straightforward and Belinkov (2022) provides an in-depth summary. It entails taking the representations produced by a model and use them to train a classifier tasked with predicting certain linguistic properties. A strong performance of the classifier implies that the original model has learned valuable linguistic information. Despite the widespread use of this approach, in recent years, various criticisms have been raised. In particular, a growing concern is that, given a large enough training dataset, a sufficiently complex probe can fit any signal. As a consequence, high accuracy on an auxiliary prediction task is not enough to conclude that a given input representation encodes a particular linguistic property (Hewitt and Liang, 2019).

Another shortcoming of diagnostic classifiers is their inability to ascertain the effective use of any linguistic information encoded by a model's representations, despite detecting it. This leads to approaches that focus on how the information is used by PLMs. One such example is the amnesic probing approach proposed by Elazar et al. (2021). This method builds on the intuition that if a property p (e.g. POS) is relevant for a specific task, such as language modeling, then the removal of p would negatively impact the ability of the model to solve the task. The approach intervenes on the model's representation layers to ablate the specific linguistic property being investigated. A similar approach it adopted by Lakretz et al. (2019), who investigate an LSTM language model performing a number agreement task. To get a better

understanding of how number information is stored, the authors progressively ablate units of the network to observe if this would cause a drop in the task being performed. This allows to identify which units are most relevant for the task at hand and ultimately provides valuable insights into how number information is encoded.

Chapter 3

Data

3.1 Data Format

All experiments are based on sets of sentence quadruples (s_1, s_2, s_3, s_4), where the last three sentences differ from s_1 with respect to a particular morphosyntactic feature, such as tense. For each experiment, s_1 is extracted from a chosen corpus, while the remaining sentences are generated, either automatically or manually, by manipulating s_1. In brief, the primary objective of each experiment is to investigate whether it is possible to learn a transformation that maps the embedding of s_1 as close as possible to that of s_2. A comprehensive method overview is provided in chapter 4.

This thesis focuses on three languages that share a common Indo-European root but exhibit distinct linguistic and morphosyntactic features: English, German, and Italian. On the one hand, English, an analytic language, primarily uses word order to express grammatical relationships. On the other hand, German and Italian, as synthetic languages, feature a richer morphology but remain significantly different due to their respective Germanic and Romance origins.

For each language, two experiments are carried out, each requiring a different dataset. This entails obtaining two sets of s_1 sentences from which the two datasets for each language are created. A total of six datasets are compiled for this study. For each dataset, s_1 sentences are chosen to be as short and simple as possible. The aim is to obtain quadruples that differ only in the elements encoding the specific linguistic feature under investigation, and to ensure that the corresponding embeddings primarily reflect these features.

The first experiment focuses on analyzing tense as a morphosyntactic feature. In each of the languages under consideration, s_1 is expected to be a sentence in the present simple tense, while s_2 should be its past simple tense counterpart. The remaining sentences, s_3 and s_4, can be formulated in any other combination of tense, mood, and aspect. For instance, if sentence s_1 is *I **go** to school*, s_2, s_3, and s_4 might respectively correspond to *I **went** to school* (past simple), *I **will go** to school* (future simple), and *I **would go** to school* (present conditional).

The second experiment focuses on subject-verb agreement. In this experiment, each sentence begins with a singular personal pronoun (e.g. "I", "you", "she"), followed by the appropriate verb form. While sentences s_3 and s_4 allow for flexibility in selecting

the pronoun, s_2 sentences must begin with the corresponding plural form of the pronoun used in s_1. For instance, if sentence s_1 starts with the first person singular "I", as in ***I** never **cut** class*, then s_2 should begin with the first person plural "we", as in ***We** never **cut** class*. On the other hand, s_3 and s_4 can use any pronouns except those in the first two sentences: ***They** never **cut** class*, ***He** never **cuts** class*. All sentences in a given quadruple use the same tense. However, tense, mood, and aspect can vary across different quadruples.

For each language, the two sets of s_1 sentences are automatically extracted from a dedicated corpus using the Sketch Engine API[1] (Kilgarriff et al., 2004, 2014), as described in section 3.2. Punctuation is removed, with the exception of a few common symbols (`?`,`-`) that ensure the meaning and grammaticality of certain words and sentences, such as hyphenated compound words (e.g. *know-how*) and German subordinate clauses. The resulting sets contain thousands of sentences. To simplify the generation and inspection of the remaining s_2, s_3, and s_4 sets, only a random sample of 500 s_1 sentences is considered for each experiment. Unsuitable sentences are removed and replaced by hand, thereby introducing a selection bias that makes the sample pseudo-random.

3.2 Data Collection Tools and Sources

All s_1 sentences are obtained through the Sketch Engine Python API. Sketch Engine is a versatile tool that features a user-friendly interface for querying and analyzing multiple text corpora in different languages. It supports various functions, including concordance searches as well as collocation and term frequency distribution analysis.

The concordance function is particularly useful for the purposes of this thesis as it allows, given a chosen corpus, to retrieve instances of a specific target token or sequence of tokens within their contexts. Importantly, each concordance search can be deeply customized, for example to specify the nature of the surroundings of a particular target by excluding or including certain word classes. The highest degree of customization can be achieved using the Corpus Query Language (CQL) which allows to rapidly compute syntactically rich queries on large corpora.

CQL is based on a formalism initially proposed in the context of the IMS Open Corpus Workbench[2] project (Christ, 1994) and was later extended by Jakubíček et al. (2010). CQL queries are patterns designed to match specific tokens or sequences of tokens within a corpus. All tokens are characterized by part-of-speech tags, while corpora usually come with a set of structure tags that mark sequences of tokens, such as sentences (`<s></s>`), paragraphs (`<p></p>`) and documents (`<doc></doc>`). As an example, a query searching the corpus for samples containing instances of *eat* and *vegetable* with two adjectives between them is:

```
[lemma="eat"] [tag="JJ.*"]{2} [lemma="vegetable"]
```

The query matches phrases such as *eat green leafy vegetables*, *eating crunchy raw vegetables*, *ate delicious fresh vegetables* and *eating healthy green vegetable*. As exemplified by these results, the first part of the query `[lemma="eat"]` retrieves any form of the verb

[1] `https://www.sketchengine.eu/`
[2] `https://cwb.sourceforge.io/`

to eat. The second part `[tag="JJ.*"]{2}` matches two generic adjectives with no further attributes specified e.g. comparatives (`JJR`) or superlatives (`JJS`). The final part, `[lemma="vegetable"]`, retrieves any instance of the common noun *vegetable*, whether it is singular or plural. By replacing `[lemma]` with `[word]` it is possible to limit the search only to the exact form of a given target e.g. `[word="vegetables"]` returns only samples containing *vegetable* in its plural form.

For each language, the set of s_1 sentences are extracted from the corresponding TenTen corpus. TenTen corpora (Jakubíček et al., 2013) are a collection of comparable, web-crawled corpora available in over 40 languages. For this thesis, `enTenTen21`, `itTenTen20` and `deTenTen20` are employed. It is worth noting that, despite being a family of corpora that meet uniform standards, each corpus features its own distinctive tagset. As a result, the same query cannot be applied to multiple corpora.

Given a set of s_1 sentences obtained using the Sketch Engine API, the corresponding s_2, s_3 and s_4 sentences are either manually compiled or automatically generated using `mlconjug3` (Diao, 2023), a command-line application and Python library designed to assist in verb conjugation across six languages: French, English, Spanish, Italian, Portuguese, and Romanian.

3.3 English

English data for both experiments come from the `enTenTen21` English Web Corpus.[3] This corpus is compiled from texts collected from over 120 million web pages and comprises about 52 billion words. Part-of-speech annotation is carried out using TreeTagger[4] (Schmid, 1994), relying on a modified version of the original English Penn Treebank tagset.[5]

3.3.1 Tense Dataset

The tense dataset is compiled beginning with a set of s_1 sentences in the present simple tense, which is retrieved using the following query:

```
<s>[tag="DT" | tag="P[P|PZ]" | tag="N[P|PZ|PS|PSZ|NZ|NSZ]"]
[tag!="V.*"]{0,5}
[word="target" & tag="V.*"]
[tag!="V.*"]{1,5}</s>
```

The query looks for sentences (`<s></s>`), starting either with a determiner (`DT`), a personal (`PP`) or possessive (`PPZ`) pronoun, or a noun. The latter can be either a singular or plural proper noun (`NP|NPS`). Alternatively, singular or plural possessive proper nouns (`NPZ|NPSZ`), such as *Britain's*, are also permitted. Lastly, singular, plural or mass possessive nouns (`NNSZ|NNZ`), like *people's* and *world's*, are also an option. The second part of the query allows for the inclusion of up to five optional non-verb tokens. The third segment specifically identifies the target verb in the present simple tense to be used, while the last part of the query allows for the inclusion of one to five additional non-verb tokens.

[3] Corpus version: `ententen21_tt31` (June 2023).

[4] `https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`

[5] `https://www.sketchengine.eu/english-treetagger-pipeline-2/`

For each target, the query is executed once, looking for up to 20 sentences containing it. The targets are obtained from a list of verb lemmas gathered from the corpus. For each lemma, both of its present simple forms are retrieved using the `mlconjug3` Python package. For example, given the infinitive form *to go*, *go* and *goes* are obtained and used as targets. For the English tense dataset the 1000 most frequent verb lemmas in the `enTenTen21` corpus are considered.

The query retrieves a total of 3035 sentences with 79 different lemmas. From this set, a pseudo-random sample of 500 sentences containing 77 different verb lemmas is selected. For each sentence in the sample, the corresponding s_2, s_3, and s_4 sentences are generated. s_2 and s_3 sentences are constructed in the indicative mood, with s_2 in the past simple tense and s_3 in the future simple tense. s_4 sentences, on the other hand, are formed in the present conditional using "would" + <infinitive>.

s_2 sentences can be automatically generated with the `mlconjug3` package by simply replacing the present simple verb in s_1 with the corresponding past simple form. However, this automated process is not always feasible for the remaining two sets of sentences, where the modal ("will" and "would") slightly changes the sentence structure. For example, consider a sentence such as *Maddie finally hears from Buck*. The correct s_4 counterpart is *Maddie* ***would*** *finally* ***hear*** *from Buck*, as in English adverbs of frequency typically appear before the main verb in a sentence. `mlconjug3` only allows to conjugate a given verb and does not handle such cases. As a consequence, s_3 and s_4 sentences are compiled manually.

3.3.2 Agreement Dataset

The agreement dataset is compiled from a set of s_1 sentences that begin with a singular personal pronoun: "I", "you", "he", "she" or "it".

```
<s>[word="I|[Yy]ou|[Hh]e|[Ss]he|[Ii]t" & tag="PP"]
[tag!="V.*"]{0,5}
[lemma="target" & tag="V.*"]
[tag!="V.*"]{1,5}</s>
```

Other than the first part, which matches singular personal pronouns (`PP`), the query differs from the one used to collect tense data in that it targets verb lemmas. As a result, for each lemma, the query retrieves up to 20 sentences in which the target may appear in different tenses, moods, and aspects. For example, given the verb *to increase*, some of the retrieved sentences are *I* ***will increase*** *funding for libraries*, *It also* ***increases*** *the risk of miscarriage* and *It* ***would*** *also* ***increase*** *their influence over government*. For the English agreement dataset, only the 200 most frequent verb lemmas in the corpus are considered, as removing the present simple tense constraint allows to retrieve significantly more sentences.

The query retrieves a sample of 3781 sentences encompassing 200 different lemmas. From this set, a pseudo-random sample of 500 sentences containing 184 different verb lemmas is selected. For each sentence in the sample, the corresponding s_2, s_3, and s_4 counterparts are generated manually. In particular, s_2 sentences are obtained by pluralizing pronouns in s_1 sentences as follows: "I" → "We", "You" → "You", "He"/"She"/"It" → "They". For instance, the s_2 counterpart to ***He*** *completed his PhD in 2021* is ***They*** *completed their PhD in 2021*. On the other hand, sentences s_3 and s_4

are formulated in such a way that they do not begin with the same pronouns as s_1 and s_2.

3.4 Italian

Italian data for both experiments come from the `itTenTen20` Italian Web Corpus.[6] This corpus is compiled from texts collected from over 30 million web pages and comprises about 12 billion words. Part-of-speech annotation and lemmatization is carried out using FreeLing[7] (Padró and Stanilovsky, 2012), an open-source tool, relying on its Italian tagset.[8]

3.4.1 Tense Dataset

The tense dataset is compiled beginning with a set of s_1 sentences in the "presente indicativo" (present simple tense), which is retrieved using the following query:

```
<s>[tag="D[A|D|I|P].*" | tag="P[D|P].*N.*"]
[tag!="V.*"]{0,5}
[word="target" & tag="V.*"]
[tag!="V.*"]{1,5}</s>
```

The query is for the most part identical to the one described for English tense data in subsection 3.3.1. The only difference is in the first part, which tries to match sentences starting with either a determiner (`D`) or a demonstrative or personal nominative pronoun (`P[D|P].*N.*`). Determiners can either be definite articles (`DA`), demonstrative adjectives (`DD`), indefinite adjectives (`DI`) or possessive adjectives (`DP`).

Unlike English, where only the third person singular exhibits some variation, Italian has an average of six different verb forms per lemma. Therefore, only the 200 most frequent verb lemmas in the corpus are chosen and conjugated in their corresponding present tense forms to serve as targets for the query, which attempts to retrieve up to 20 sentences per target. This allows to dramatically reduce both the number of API calls and the time required to collect the s_1 sentences.

A total of 18835 sentences, covering 199 verb lemmas, are collected. From this set a pseudo-random sample of 500 sentences containing 171 different verb lemmas is selected. For each sentence in the sample, the corresponding s_2, s_3, and s_4 sentences are generated. s_2 and s_3 sentences are constructed in the indicative mood, with s_2 in the "passato remoto", which roughly corresponds to the English past simple tense, and s_3 in the "futuro semplice", which is akin to the English future simple tense. Lastly, s_4 sentences are formed in the "condizionale presente", which is similar to the English present conditional tense.

All s_2, s_3 and s_4 sentences are automatically generated with the `mlconjug3` package by replacing the present simple verb form in s_1 with the appropriate one. The final result is checked by hand.

[6] Corpus version: `ittenten20_fl1` (February 2022).

[7] `https://nlp.lsi.upc.edu/freeling/`

[8] `https://www.sketchengine.eu/italian-freeling-part-of-speech-tagset/`

3.4.2 Agreement Dataset

The agreement dataset is compiled from a set of s_1 sentences that begin with a singular personal pronoun. For reference, the Italian personal pronouns include "io" ("I"), "tu" ("you"), "egli"/"lui"/"esso" ("he"), "ella"/"lei"/"essa" ("she"), "noi" ("we"), "voi" ("you"), and "essi"/"esse"/"loro" (they).

```
<s>[tag="PP.*SN.*"]
[tag!="V.*"]{0,5}
[lemma="target" & tag="V.*S.*"]
[tag!="V.*"]{1,5}</s>
```

Except for the first part, which matches singular personal pronouns in the nominative case (`PP.*SN.*`), the query differs from the one used to collect tense data in that it relies on verb lemmas as targets, trying to retrieve singular forms of each given lemma (`V.*S.*`). For each of the 200 most frequent verb lemmas in the corpus, the query returns up to 20 sentences in which the target may appear in different tenses, moods, and aspects. This is consistent with the approach used for compiling the English agreement dataset, as described in subsection 3.3.2.

The query collects a sample of 3624 sentences encompassing 200 different lemmas. From this set, a pseudo-random sample of 500 sentences containing 180 different verb lemmas is selected. For each sentence in the sample, the corresponding s_2, s_3, and s_4 counterparts are generated manually. Specifically, s_2 sentences are obtained by pluralizing pronouns in s_1 sentences as follows: "Io" → "Noi", "Tu" → "Voi", "Egli"/"Esso"/"Lui" → "Loro"/"Essi" and "Ella"/"Essa"/"Lei" → "Loro"/"Esse". The remaining sentences s_3 and s_4 are formulated in such a way that they do not begin with the same pronouns as s_1 and s_2.

3.5 German

German data for both experiments come from the deTenTen20 German Web Corpus.[9] This corpus is compiled from texts collected from over 47 million web pages and comprises about 17 billion words. Part-of-speech annotation is carried out using RFTagger,[10] developed by Schmid and Laws (2008), relying on the German tagset[11] originally proposed by the same authors.

3.5.1 Tense Dataset

The tense dataset is compiled beginning with a set of s_1 sentences in the "Präsens" (present simple tense), which is retrieved using the following query:

```
<s>[tag="PRO.[Pers|Dem|Poss|Indef].*Nom.*" | tag="ART.*Nom.*" ]
[tag!="V.*"]{0,5}
[lemma="target" & tag="VFIN.*Pres.Ind.*"]
[tag!="V.*"]{1,5}</s>
```

[9] Corpus version: `detenten20_rft3` (August 2022).

[10] `https://www.cis.uni-muenchen.de/~schmid/tools/RFTagger/`

[11] `https://www.sketchengine.eu/german-rftagger-part-of-speech-tagset/`

The query tries to retrieve sentences (<s></s>) that begin either with a personal, demonstrative, possessive or indefinite pronoun (PRO) in the nominative case, for example *er*, *dieser*, *meiner* or *jemand*. Alternatively, nominative articles (ART), such as *der*, *die* or *das*, are also permitted. The second part of the query allows for the inclusion of up to five optional non-verb tokens. The third section focuses on a particular verb lemma, with the search limited to finite verbs in the indicative present tense (VFIN.*Pres.Ind.*). Finally, the last section of the query allows for the inclusion of one to five additional non-verb tokens.

For the German tense dataset the 990 most frequent verb lemmas in the corpus are considered and used as targets in the query, which attempts to collect up to 20 sentences for each lemma. A total of 18433 sentences, encompassing 989 different verb lemmas, are retrieved. From this set, a pseudo-random sample of 500 sentences is obtained. The sample includes 384 distinct verb lemmas. For each instance in the sample, the corresponding s_2, s_3, and s_4 sentences are generated manually. Specifically, s_2 and s_3 sentences are constructed in the indicative mood, with s_2 in the "Präteritum", which is equivalent to the English past simple tense, and s_3 in the "Futur I", which corresponds to the English future simple tense. On the other hand, s_4 sentences are formed in the "Konjunktiv II", using "würde" + <verb infinitive>.

3.5.2 Agreement Dataset

The agreement dataset is compiled from a set of s_1 sentences that begin with a singular personal pronoun. For reference, the German personal pronouns include "Ich" ("I"), "Du" ("you"), "Er" ("he"), "Sie" ("she"), "Es" ("it"), "Wir" ("we"), "Ihr" ("you"), and "Sie" ("they").

```
<s>[tag="PRO.Pers.*Nom.Sg.*"]
[tag!="V.*"]{0,5}
[lemma="target" & tag="VFIN.*Sg.*.Ind.*"]
[tag!="V.*"]{1,5}</s>
```

The first part of the query matches singular personal pronouns in the nominative case (PRO.Pers.*Nom.Sg.*). The second and last part of the query are identical to their counterparts used to retrieve tense data. The third part, on the other hand, focuses on a given verb lemma, with the search limited to finite singular verbs in the indicative present tense (VFIN.*Sg.*.Ind.*).

The query returns a total of 3643 sentences, covering 197 verb lemmas. A pseudo-random sample of 500 sentences is chosen from this set, encompassing 181 distinct verb lemmas. For each sentence in the sample, the corresponding s_2, s_3, and s_4 counterparts are generated manually. Specifically, s_2 sentences are obtained by pluralizing pronouns in s_1 sentences as follows: "Ich" → "Wir", "Du" → "Ihr", "Er/Sie/Es" → "Sie". The remaining sentences s_3 and s_4 are formulated in such a way that they do not begin with the same pronouns as s_1 and s_2.

3.6 Data Summary

Table 3.1 presents an overview of the tense and agreement datasets collected for the three languages with respect to the s_1 sentences. Specifically, for the s_1 sentences

in each dataset the table details the number of unique verb lemmas, the length of the shortest and longest sentence, the average sentence length and the Type-Token Ratio (TTR). The latter is a measure of lexical diversity computed by dividing the number of different words (types) in a text by the total number of words (tokens). The value of TTR can range from 0 to 1, with higher values indicating more lexical variation and lower values indicating less. Furthermore, a high TTR suggests a rich or complex morphology, as morphologically complex languages tend to exhibit more lexical variation (Çöltekin and Rama, 2023). This statistic is highly dependent on the sample size and several variations have been suggested to account for this, for example the Moving-Average TTR (Covington and McFall, 2010). However, as all six datasets are of the same length (500 samples), the traditional TTR is used for this analysis.

	Unique verb lemmas	Min. length	Max. length	Avg. length	TTR
EN tense	77	2	11	5.83	0.45
IT tense	171	2	11	6.01	0.48
DE tense	384	2	11	6.08	0.55

	Unique verb lemmas	Min. length	Max. length	Avg. length	TTR
EN agreement	184	2	9	5.20	0.41
IT agreement	180	2	11	5.05	0.4
DE agreement	181	2	7	4.94	0.43

Table 3.1: The top table groups the three tense datasets and reports the corresponding statistics: number of unique verb lemmas, shortest and longest sentence in each dataset, average sentence length, and Type-Token Ratio (TTR). The bottom table shows the same statistics for the agreement datasets. In both cases, the statistics are computed only on the s_1 sentences retrieved for each dataset.

There is a significant difference in the number of unique verb lemmas between the tense datasets. Among the three, the English one has the least with only 77 lemmas, while the German dataset has the most with 384. Italian sits in the middle with a count of 171. Conversely, the average sentence length in all three datasets is close to six tokens. The higher morphological complexity of Italian and German is reflected by their TTR scores of 0.45 and 0.55 respectively, while English exhibits a lower score of 0.45. Turning to the agreement datasets, the number of unique verb lemmas is almost the same across the three languages, and the average sentence length is close to 5 tokens. Surprisingly, despite the slightly higher TTR score for the German dataset, the overall scores are much closer than those of the tense datasets.

It is important to note that the data used in this study has some potential limitations that need to be considered. First, all of the dataset are compiled (either automatically or manually) and checked by only one person. This may lead to grammatical errors and typos that can ultimately compromise the results of the experiments. Second, punctuation removal is the only preprocessing step applied to the sentences. As a consequence, some of the samples exhibit unusual casing patterns, due to the TenTen corpora being web-crawled. For example: *I will NEVER attend another event there* or *Io rispondo A nessuno* (*I answer TO nobody*). Even though the Pretrained Language Models being tested account for case, the effect of these samples on the

results is uncertain as they may substantially deviate from those observed by the model during training. Lastly, for an optimal comparison of the embeddings in each quadruple, the respective sentences should only differ in the linguistic features being investigated. For example, in the case of agreement, only the pronoun and the verb are expected to change: ***It*** *also* ***improves*** *audio quality* → ***They*** *also* ***improve*** *audio quality*. However, this is not always possible: *She also established* ***her*** *own private practice* → *They also established* ***their*** *own private practice*.

Chapter 4

Method and Experimental Setup

4.1 Method Overview

The primary goal of this thesis is to investigate whether transformer-based Pre-trained Language Models (PLMs) encode any morphosyntactic information and, in this respect, this work aligns with the efforts of other authors who have tackled this issue in the context of interpretability, as outlined in subsection 2.3.2. However, contrary to the approaches described so far, the present method does not rely on probing classifiers of any sorts, but instead directly manipulates the contextual embeddings produced by PLMs through affine and non-linear transformations implemented as two neural probes and attempts to align them to specific target representations.

The idea of vector manipulation has been extensively explored in the context of word embeddings. In particular, as anticipated in subsection 2.1.2, word representations appear to capture linguistic regularities which can be expressed as vector offsets between pairs of words sharing a particular relation. For example, with respect to morphosyntactic regularities, an abstract relationship between singular and plural can be obtained from the difference between the two vector representations of *apple* and *apples*. Mikolov et al. (2013c) observe that summing the resulting vector (apples − apple) to the singular representation of another word, for example *car*, allows to retrieve an approximation of its plural form e.g. cars = apples − apple + car. Nicolai et al. (2015) confirm these results for English and replicate the experiment on a set of more morphologically complex languages: Dutch, French, German, and Spanish. The authors explore various morphosyntactic phenomena, including singular/plural and possessive/nominative forms for nouns, as well as comparative/superlative/base forms for adjectives. Additionally, they conduct an analysis of verb forms, looking at the preterite, infinitive, and 3rd person singular present.

The approach developed for this thesis exhibits some parallels with the prior methodology, in that it looks at the same class of phenomena and relies on embedding manipulation. Nonetheless, it also deviates from it by focusing on sentence representations rather than word embeddings and by leveraging affine and non-linear transformations implemented by fully-connected layers.

The method can be summarized as follows: given a PLM denoted as $\mathscr{M}$, a quadruple of sentences (s_1, s_2, s_3, s_4) and two neural probes $\mathscr{P}_1$ and $\mathscr{P}_2$, the embeddings of the

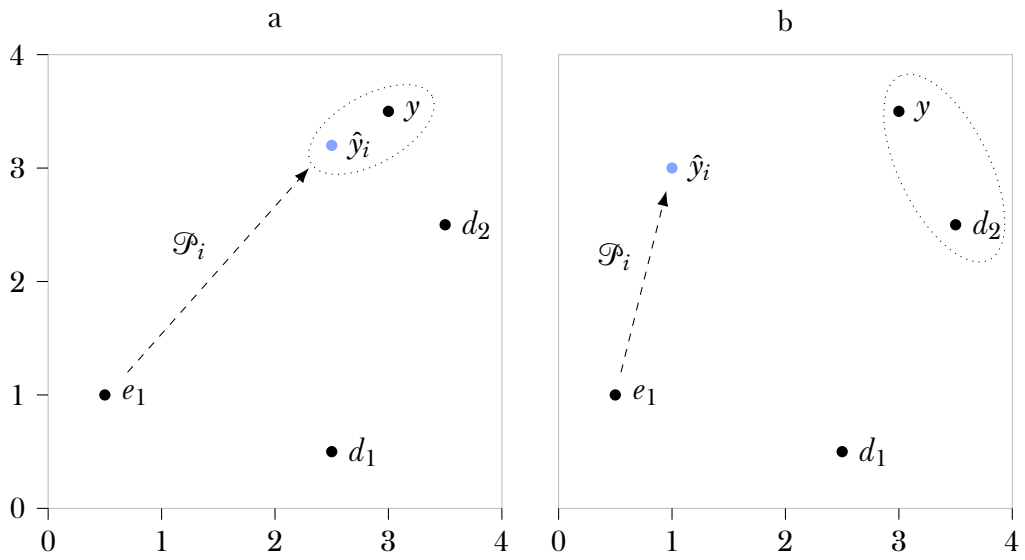

Figure 2: The plot on the left represents an ideal scenario where the affine transformation $\hat{y}_i$, produced by a probe $\mathcal{P}_i$, corresponds to the closest point to the target representation y. The plot on the right, pictures a scenario in which the target y is closer to one of the distractors (d_2) than to the affine transformation $\hat{y}_i$. In both plots, the closest representations are highlighted by an ellipse.

four sentences are first retrieved, such that $e_i = \mathcal{M}(s_i)$. The representation of s_i corresponds to the `[CLS]` token retrieved from the last layer of $\mathcal{M}$. In the case of BERT and RoBERTa, all of the 12 layers `[CLS]` representations are tested. However, for clarity, results other than those from the last layer are reported only if they are significantly different.

The embedding of the first sentence e_1 is transformed by each probe with the goal of mapping it to the target representation e_2, from now on referred to as y. These two transformations, $\hat{y}_1 = \mathcal{P}_1(e_1)$ and $\hat{y}_2 = \mathcal{P}_2(e_1)$, are subsequently compared with the three embeddings y, e_3 and e_4. The latter two serve as distractors and are referred to as d_1 and d_2.

To elaborate further, for each quadruple of sentences in the dataset, five comparisons are carried out, computing a similarity score for each pair of embeddings: $(\hat{y}_i, y)$, $(\hat{y}_i, d_1)$, $(\hat{y}_i, d_2)$, (y, d_1), (y, d_2), where $\hat{y}_i$ refers to either one of the two transformations obtained from e_1 with the two neural probes. Each comparison is performed twice, using both consine similarity and Euclidean distance as similarity metrics.

If, among the five comparisons, the highest similarity score is observed between the transformed representation $\hat{y}_i$ and the target representation y (as shown in Figure 2a), this suggests that the initial embedding e_1 contains some latent features that can be manipulated to approximate the target. This observation, akin to the results discussed earlier for word embeddings, implies that the transformer-based model $\mathcal{M}$ may encode the specific morphosyntactic phenomenon expressed by the first two sentences in the quadruple.

As an example, consider the set of sentences: *We talk often*, *We talked often*, *We will talk often*, and *We are talking often*. In this context, observing a high similarity between

$\hat{y}_i$ and y may indicate that $\mathscr{M}$ encodes information about the English past tense inflection i.e. "talk"→"talked". However, there may be scenarios where, even though $\hat{y}_i$ and y exhibit the highest similarity among the first three comparisons, y is, in fact, closer to one or both of the distractors (as shown in Figure 2b).

Two classes of morphosyntactic phenomena, tense and subject-verb agreement, are the focus of the described approach, and both are investigated in the context of three languages, English, Italian and German. With respect to tense, the two probes attempt to map the representation of the present tense sentence s_1 to its past tense counterpart s_2, as outlined in the previous example. For subject-verb agreement, on the other hand, the focus is on pluralization. As described in chapter 3, each sentence s_1 in the agreement dataset begins with a singular personal pronoun and both probes try to map its representation to that of its pluralized s_2 counterpart. The pluralization process affects both the subject personal pronoun, the verb and potentially other elements of the sentence as well. As an example, consider the following quadruple: *I cheerfully offer **my** assistance*, *We cheerfully offer **our** assistance*, *She cheerfully offers **her** assistance* and *They cheerfully offer **their** assistance*.

4.2 Models

For each experiment, three Pretrained Language Models (PLMs) based on the architectures described in section 2.2 are used to generate the embeddings of the sentences in each quadruple. All models are accessed through the Transformers API (Wolf et al., 2020).

For English, `bert-base-cased`[1] and `roberta-base`[2] are the chosen implementations for BERT and RoBERTa. Both of them are monolingual, case-sensitive implementations of the architectures originally presented by Devlin et al. (2019) and Liu et al. (2019), respectively.

In the case of Italian, `dbmdz/bert-base-italian-cased`[3] is the selected BERT implementation. The model is case-sensitive and adheres to the original BERT architecture, implementing both Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) as pretraining objectives. The second implementation is UmBERTo[4] (Parisi et al., 2020), a monolingual, case-sensitive RoBERTa-based Language Model. Unlike RoBERTa, UmBERTo relies on Whole Word Masking, a strategy that masks only whole words and not subwords. The choice of UmBERTo is motivated by a lack of any other Italian monolingual RoBERTa models at the time of writing.

For German, the BERT implementation is `dbmdz/bert-base-german-cased`.[5] The model is case-sensitive and conforms to the original BERT architecture. Turning to RoBERTa, `benjamin/roberta-base-wechsel-german`[6] is the chosen implementation. The model is trained with WECHSEL (Minixhofer et al., 2022), a method that

[1] `https://huggingface.co/bert-base-cased`
[2] `https://huggingface.co/roberta-base`
[3] `https://huggingface.co/dbmdz/bert-base-italian-cased`
[4] `https://huggingface.co/Musixmatch/umberto-commoncrawl-cased-v1`
[5] `https://huggingface.co/dbmdz/bert-base-german-cased`
[6] `https://huggingface.co/benjamin/roberta-base-wechsel-german`

allows to obtain a German version of RoBERTa from the English one. As for the Italian UmBERTo, this is the only German monolingual RoBERTa model available at the time of writing.

The third PLM, `paraphrase-multilingual-mpnet-base-v2`[7], is a multilingual Sentence Transformer model based on MPNET-base (Song et al., 2020) and trained on parallel data for over 50 languages. MPNET-base relies on a novel pretraining strategy called Masked and Permuted Language Modeling (MPNet). MPNet builds upon two strategies MLM and Permuted Language Modeling (PLM) (Yang et al., 2019).

4.3 Two neural probes $\mathscr{P}_1$ and $\mathscr{P}_2$

All experiments are carried out twice, using two neural probes implemented in PyTorch (Paszke et al., 2019) and the embeddings obtained from the transformer-based model $\mathscr{M}$ under investigation. This setup allows to examine the impact of the transformations resulting from the two different probe configurations on the task at hand.

The first probe, denoted as $\mathscr{P}_1$, consists of a single-layer, fully-connected neural network. The probe takes a batch of n sentence embeddings $\mathbf{E} \in \mathrm{R}^{n \times m}$ as input and produces their affine transformation using a learned matrix $\mathbf{W} \in \mathrm{R}^{m \times m}$ along with a bias term b:

$$\mathscr{P}_1(\mathbf{E}) = \mathbf{E}\mathbf{W} + b. \tag{4}$$

On the other hand, the second probe, referred to as $\mathscr{P}_2$, is a double-layer, fully-connected neural network that includes a non-linear activation function $\phi(x) = \max(x, 0)$, the Rectified Linear Unit (ReLU). This second probe performs two consecutive transformations, relying on learned matrices $\mathbf{W}_1 \in \mathrm{R}^{m \times h}$ and $\mathbf{W}_2 \in \mathrm{R}^{h \times m}$, along with the corresponding bias terms b_1 and b_2:

$$\mathscr{P}_2(\mathbf{E}) = \phi(\mathbf{E}\mathbf{W}_1 + b_1)\mathbf{W}_2 + b_2. \tag{5}$$

Both probes map the input embeddings back to their original dimension m. This is an essential step that allows for a direct comparison between the transformed input representation $\hat{y}$ and the target and distractor sentence embeddings y, d_1 and d_2.

4.3.1 Hyperparameter Tuning and Training

Hyperparameter tuning is carried out with RayTune (Liaw et al., 2018), for each layer of every Pretrained Language Model (PLM), using the default random-search algorithm to explore the following parameter space:

1. learning rate: `(1e-7, 1e-4)`
2. weight decay: `(1e-3, 1e-2)`

[7] `https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2`

3. beta 1: `(5e-1, 9e-1)`
4. beta 2: `(5e-1, 9e-1)`
5. batch size: `[8, 16, 32]`

The first four hyperparameters pertain to the chosen `AdamW` optimizer and are selected from log-uniform distributions, while the last one decides the number of samples processed during each forward pass.

Hyperparameter tuning is executed anew for each experiment within the array of languages and PLMs under analysis. The best hyperparameter configurations are presented in section A.1.

For every experiment, each probe is assessed on 10 different, randomly selected hyperparameter configurations. Specifically, for each configuration, each probe is trained for 5 epochs to map the source embedding e_1 to the corresponding target embedding y. The training is carried out on 80% of the data collected for the ongoing experiment. The remaining 20% of the data is allocated for testing. The selected evaluation metric is the Mean Squared Error (MSE) between $\hat{y}_i$ and y. Out of the 10 configurations, the one yielding the lowest MSE is selected and used for the actual training which lasts 15 epochs.

Neural networks can be affected by their initialization and therefore exhibit a varying performance. To address this issue both probes are trained using a 5-fold cross-validation approach. This method entails dividing the dataset into k subsets. One subset is reserved for validation, while the remaining $k - 1$ are used for training.

During the validation step, similarity is computed using both cosine similarity and Euclidean distance. Results are then recorded in `CSV` files, detailing, for each input sample, the corresponding quadruplet of sentences and the six comparisons made between their embeddings: $(\hat{y}_i, y)$, $(\hat{y}_i, e_3)$, $(\hat{y}_i, e_4)$, (y, e_3), (y, e_4) and (e_3, e_4).

Finally, the files are merged together, and the six similarity scores are plotted. This graphical representation helps determine whether, on average, the similarity between $\hat{y}_i$ and the target representation y is the highest.

The total number of experiments (36) is determined by taking the Cartesian product between the set of PLMs {BERT, RoBERTa, Sentence Transformer}, the set of probes $\{\mathcal{P}_1, \mathcal{P}_2\}$, the languages under consideration {English, German, Italian} and the two linguistic features for which each PLM is probed {tense, agreement}.

Chapter 5

Results

This section presents the results of the experiments conducted on the sentence quadruples in the tense and agreement datasets compiled for English, Italian, and German. For each language, three Pretrained Language Models (PLMs) – BERT, RoBERTa, and a Sentence Transformer model – are probed for two morphosyntactic features.

As detailed in chapter 4, each experiment is carried out twice, using two probes, $\mathcal{P}_1$ and $\mathcal{P}_2$, to generate a transformation of the first sentence embedding in each quadruple. If the similarity score between the probe's transformation $\hat{y}$ and the target embedding y is the highest, this may indicate that the PLM used to generate the four embeddings actually encodes the specific morphosyntactic phenomenon under consideration. For each experiment, two similarity metrics are considered: consine similarity and Euclidean distance.

Additionally, comparing the results obtained from the two probes may provide insights into the nature of the embedding space for each PLM. Specifically, a higher similarity score between $\hat{y}$ and y with respect to $\mathcal{P}_2$ may hint at a non-linear space. On the other hand, a better performance of the affine transformation, could be an indication that the embedding space is somewhat linear.

5.1 English

5.1.1 BERT Tense Results

This section presents the results for the English tense dataset with respect to BERT. The `[CLS]` token from BERT's final layer is used to represent every sentence in each quadruple. Table 5.1 summarizes the similarity scores for all embedding pairs, considering both probes, $\mathcal{P}_1$ and $\mathcal{P}_2$. Results are broadly similar across all layers of BERT. Therefore, for the sake of clarity, an overview of all layers is presented in the Appendix (see Table A.37, Table A.38, Table A.39 and Table A.40).

The transformation $\hat{y}$ generated by $\mathcal{P}_1$ is closer to the target y, than to the distractors d_1 and d_2. This holds true both for cosine similarity and Euclidean distance with an average score of 0.9745 and 3.6241, respectively. $\mathcal{P}_2$, on the other hand, does not perform as well. The non-linear transformation generated by the second probe is

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathscr{P}_1$	0.9745 ± 0.0082	0.9694 ± 0.0071	0.9729 ± 0.0066	0.9794 ± 0.0087	0.9847 ± 0.0079
$\mathscr{P}_2$	0.9657 ± 0.0099	0.9647 ± 0.0082	0.9685 ± 0.0072	0.9794 ± 0.0087	0.9847 ± 0.0079

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathscr{P}_1$	3.6241 ± 0.5266	4.0187 ± 0.4347	3.7958 ± 0.4359	3.2541 ± 0.6666	2.7867 ± 0.6893
$\mathscr{P}_2$	4.1892 ± 0.5404	4.3081 ± 0.4609	4.0840 ± 0.4289	3.2541 ± 0.6666	2.7867 ± 0.6893

Table 5.1: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathscr{P}_1$ and $\mathscr{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

closer to d_2 than to the desired target representation, with an average cosine similarity of 0.9685 and an Euclidean distance of 4.084. Nevertheless, despite the promising results observed for $\mathscr{P}_1$, the similarity between $\hat{y}$ and y for both probes is consistently lower than that observed between the target y and each of the distractors. This is clearly portraied in Figure 3 where the median similarity score for y-d_1 and y-d_2 is the highest across both probes and metrics.

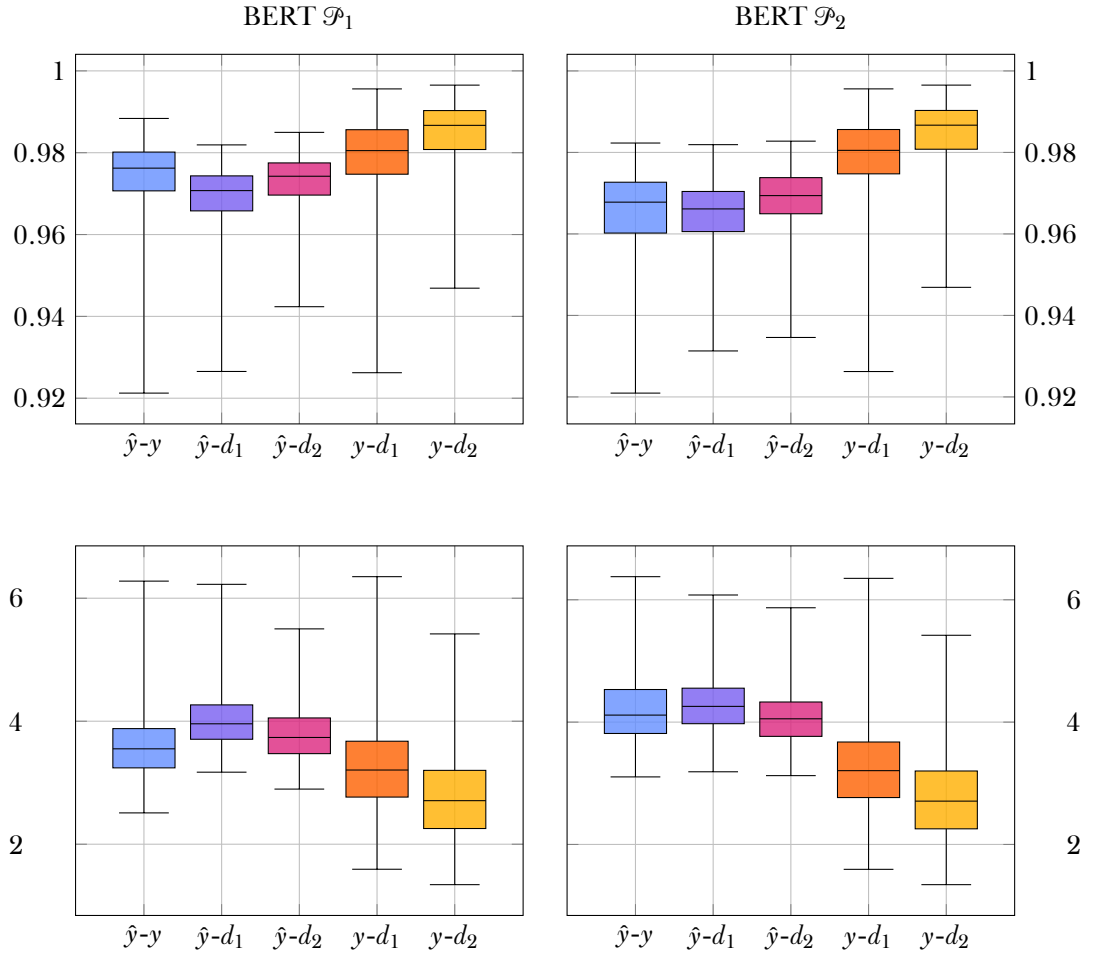

Figure 3: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathscr{P}_1$ (left) and the second probe $\mathscr{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathscr{P}_1$ (left) and $\mathscr{P}_2$ (right).

5.1.2 RoBERTa Tense Results

This section describes the RoBERTa results on the English tense dataset. The [CLS] token retrieved from the last layer of the model is used to represent every sentence in each quadruple. Table 5.2 summarizes the similarity scores for all embedding pairs, considering both probes. A complete overview covering all layers is provided in the Appendix (see Table A.41, Table A.42, Table A.43 and Table A.44).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	0.6500 ± 0.0401	0.6493 ± 0.0401	0.6497 ± 0.0401	0.9998 ± 0.0001	0.9998 ± 0.0001
$\mathcal{P}_2$	0.9489 ± 0.0045	0.9486 ± 0.0046	0.9488 ± 0.0045	0.9998 ± 0.0001	0.9998 ± 0.0001

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	9.6367 ± 0.1417	9.6632 ± 0.1419	9.6299 ± 0.1391	0.2408 ± 0.0542	0.2158 ± 0.0506
$\mathcal{P}_2$	7.2405 ± 0.1322	7.2673 ± 0.1336	7.2336 ± 0.1312	0.2408 ± 0.0542	0.2158 ± 0.0506

Table 5.2: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathcal{P}_1$ and $\mathcal{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

The similarity scores for the three comparisons $\hat{y}$-y, $\hat{y}$-d_1 and $\hat{y}$-d_2 are very close to each other across both probes and metrics. Nonetheless, looking at the cosine similarity values, the transformation generated by $\mathcal{P}_1$ is slightly closer to the target than to the distractors, registering an average similarity of 0.65. Although very weak, this trend does not hold for the Euclidean distance, where the comparison $\hat{y}$-d_2 scores a lower value (9.6299) than $\hat{y}$-y (9.6367). A similar discrepancy between the two metrics is observed for $\mathcal{P}_2$. While $\hat{y}$ is marginally closer to the target than the distractors with an average cosine similarity of 0.9489, the Euclidean distance between $\hat{y}$ and d_2 (7.2336) is lower than that between $\hat{y}$ and y (7.2405). Nevertheless, as portrayed in Figure 4, the similarity between $\hat{y}$ and the target is markedly lower than the one observed for the pairs y-d_1 and y-d_2.

While the similarity between $\hat{y}$ and y never exceeds that between the target and the distractors, something curious happens in the third layer of the model (Table A.41). For $\mathcal{P}_1$, $\hat{y}$ scores an average cosine similarity of -0.0285 for all three comparisons $\hat{y}$-y, $\hat{y}$-d_1 and $\hat{y}$-d_2. A negative cosine similarity suggests that in this layer, for the specific set of hyper-parameters obtained for the probe, the affine transformation performed by $\mathcal{P}_1$ tends to pull $\hat{y}$ away not only from the distractors, but also from the target.

5.1.3 Sentence Transformer Tense Results

This section outlines the results for the English tense dataset with respect to the Sentence Trasformer model. Sentence representations are obtained from the model's pooling layer that generates a fixed-length embedding for each sentence in a given quadruple. Table 5.3 summarizes the similarity scores for all embedding pairs, considering both probes, $\mathcal{P}_1$ and $\mathcal{P}_2$.

For both probes, the pair $\hat{y}$-y scores a higher similarity than $\hat{y}$-d_1 and $\hat{y}$-d_2, with an average cosine similarity of 0.6294 and 0.6129 and an average Euclidean distance

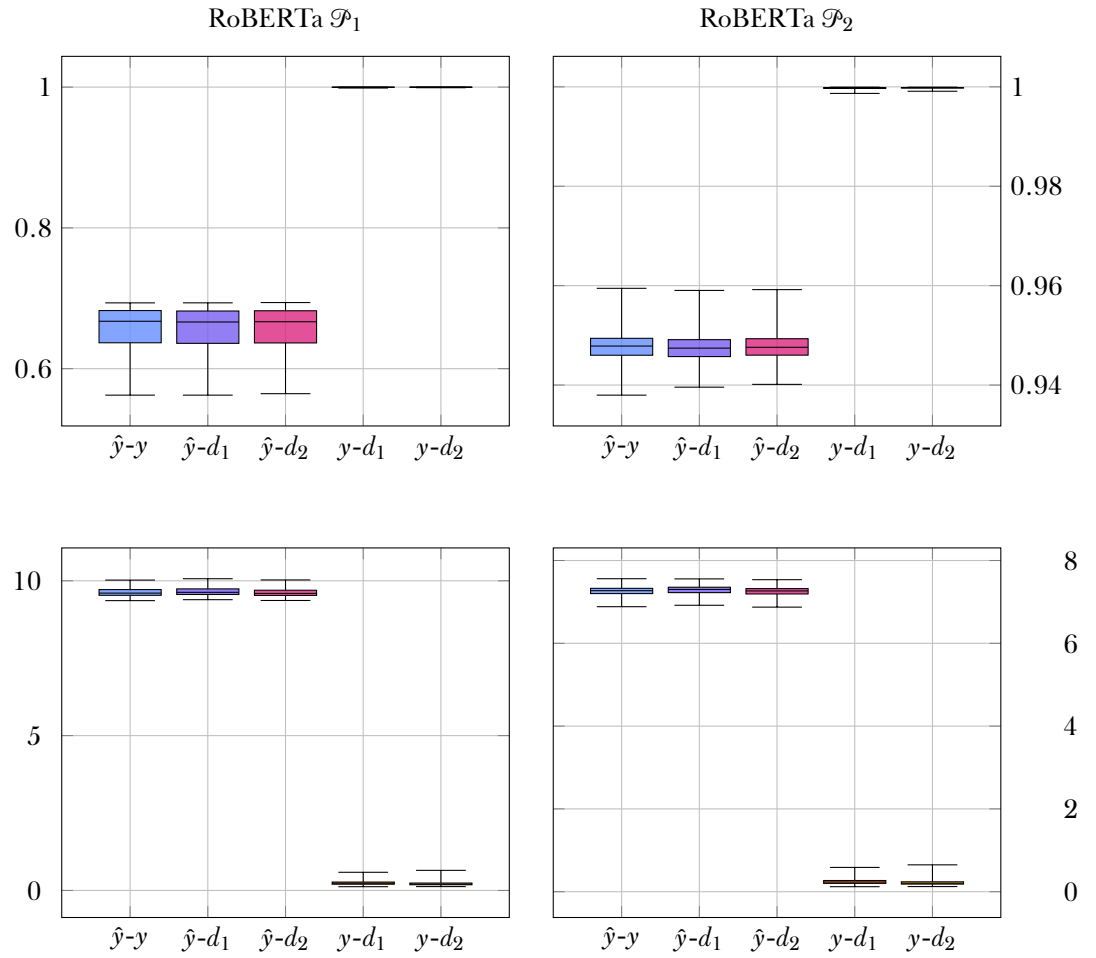

Figure 4: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathscr{P}_1$ (left) and the second probe $\mathscr{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathscr{P}_1$ (left) and $\mathscr{P}_2$ (right).

of 2.4034 and 2.4519, respectively. Nevertheless, as shown in Figure 5, the target $\hat{y}$ is significantly closer to the distractors than to the transformation y, for both probes and metrics.

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathscr{P}_1$	0.6294 ± 0.1086	0.5545 ± 0.0889	0.5821 ± 0.0893	0.9156 ± 0.0591	0.9250 ± 0.0627
$\mathscr{P}_2$	0.6129 ± 0.1064	0.5223 ± 0.0934	0.5626 ± 0.0898	0.9156 ± 0.0591	0.9250 ± 0.0627

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathscr{P}_1$	2.4034 ± 0.2582	2.5976 ± 0.0205	2.5190 ± 0.2134	1.2140 ± 0.4687	1.1273 ± 0.4768
$\mathscr{P}_2$	2.4519 ± 0.2275	2.6331 ± 0.1876	2.5452 ± 0.1943	1.2140 ± 0.4687	1.1273 ± 0.4768

Table 5.3: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathscr{P}_1$ and $\mathscr{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

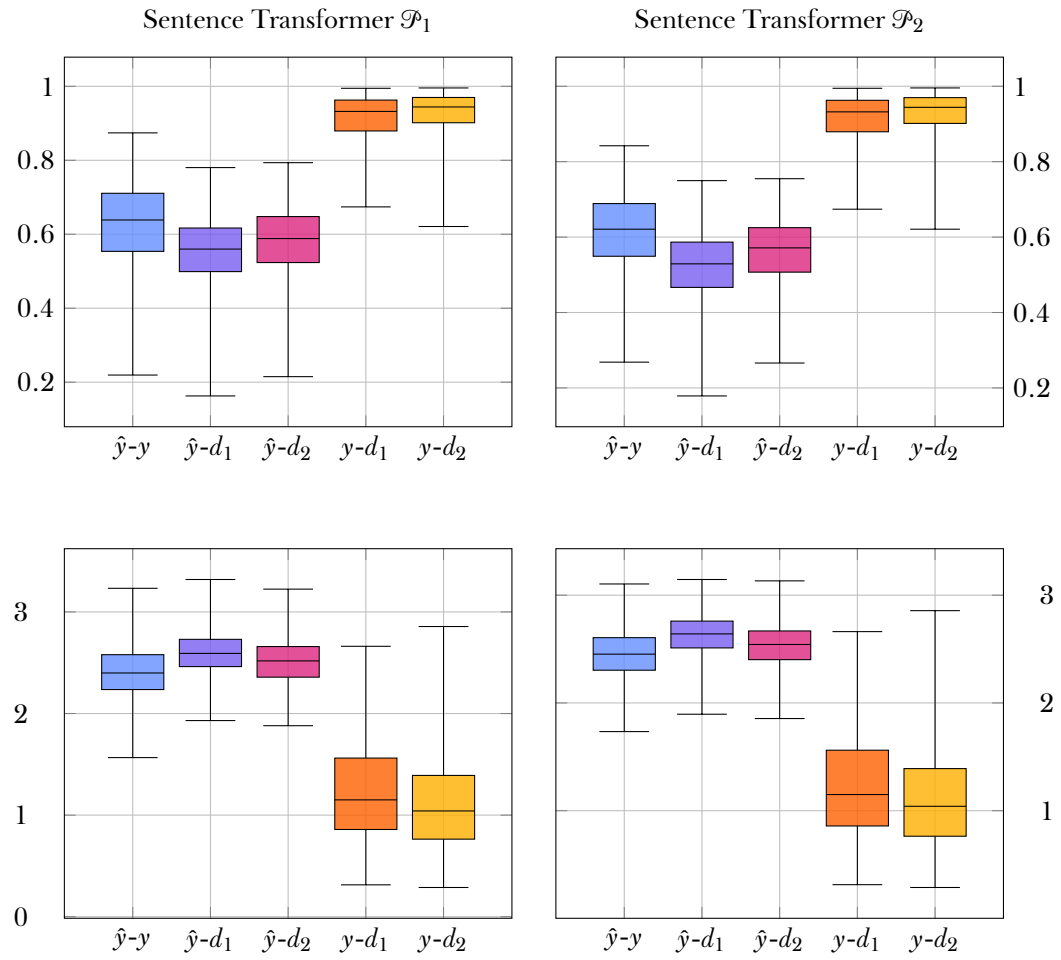

Figure 5: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathcal{P}_1$ (left) and the second probe $\mathcal{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathcal{P}_1$ (left) and $\mathcal{P}_2$ (right).

5.1.4 BERT Agreement Results

This section returns to BERT and presents the results for the English agreement dataset. Table 5.4 summarizes the similarity scores for all pairs of embeddings across both probes. An overview of the results obtained for each layer is presented in the Appendix (see Table A.45, Table A.46, Table A.47 and Table A.48).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	0.8572 ± 0.0201	0.8523 ± 0.0208	0.8517 ± 0.0209	0.9784 ± 0.0116	0.9815 ± 0.0109
$\mathcal{P}_2$	0.9735 ± 0.0079	0.9660 ± 0.0105	0.9667 ± 0.0102	0.9784 ± 0.0116	0.9815 ± 0.0109

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	8.9439 ± 0.4801	8.9564 ± 0.4768	8.9980 ± 0.4616	3.2639 ± 0.8321	3.0060 ± 0.8219
$\mathcal{P}_2$	3.6841 ± 0.5218	4.1414 ± 0.5907	4.1085 ± 0.5913	3.2639 ± 0.8321	3.0060 ± 0.8219

Table 5.4: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathcal{P}_1$ and $\mathcal{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

Across both probes, $\hat{y}$ is closer to the target y, than to the distractors. However, looking at the results for $\mathcal{P}_1$, the three comparisons $\hat{y}$-y, $\hat{y}$-d_1 and $\hat{y}$-d_2 show only a marginal difference in their cosine similarity and Euclidean distance scores. This is not true for $\mathcal{P}_2$, whose non-linear transformation $\hat{y}$ is noticeably closer to the target than to both distractors, with average cosine similarity and Euclidean distance scores of 0.9735 and 3.6841, respectively. Nonetheless, across both probes, the similarity between $\hat{y}$ and y remains lower than that observed between the target and each of the distractors, as depicted in Figure 6.

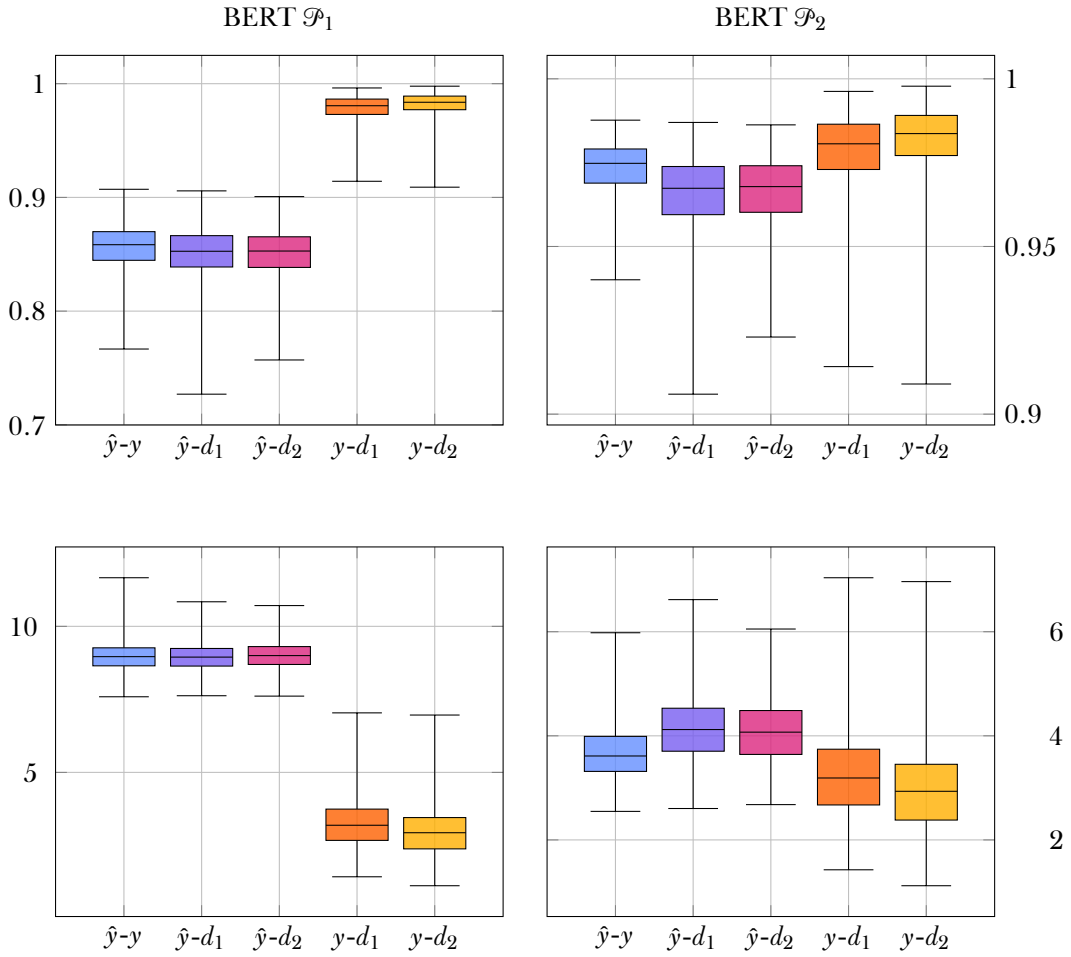

Figure 6: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathcal{P}_1$ (left) and the second probe $\mathcal{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathcal{P}_1$ (left) and $\mathcal{P}_2$ (right).

For this experiment, it is worth considering the results observed in the remaining layers, as well. In particular, the non-linear transformation operated by $\mathcal{P}_2$ achieves an encouraging result in the seventh layer (Table A.46). Here, the comparison $\hat{y}$-y scores the highest cosine similarity (0.9818), exceeding both y-d_1 (0.9795) and y-d_2 (0.9816). Looking at the corresponding Euclidean distance results in Table A.48, however, $\hat{y}$-y (4.5025) surpasses y-d_1 (4.5776) but not y-d_2 (4.2867).

5.1.5 RoBERTa Agreement Results

This section comes back to RoBERTa to look at the English agreement dataset. Table 5.5 summarizes the similarity scores for all embedding pairs, considering both probes. A complete overview covering all layers is provided in the Appendix (see Table A.49, Table A.50, Table A.51 and Table A.52).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	0.4725 ± 0.0962	0.4727 ± 0.0963	0.4724 ± 0.0964	0.9996 ± 0.0002	0.9997 ± 0.0002
$\mathcal{P}_2$	0.9993 ± 0.0003	0.9992 ± 0.0003	0.9993 ± 0.0003	0.9996 ± 0.0002	0.9997 ± 0.0002

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	10.1998 ± 0.2247	10.2160 ± 0.2196	10.1972 ± 0.2218	0.2992 ± 0.0064	0.2829 ± 0.0669
$\mathcal{P}_2$	0.4300 ± 0.0846	0.4652 ± 0.0829	0.4437 ± 0.0876	0.2992 ± 0.0064	0.2829 ± 0.0669

Table 5.5: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathcal{P}_1$ and $\mathcal{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

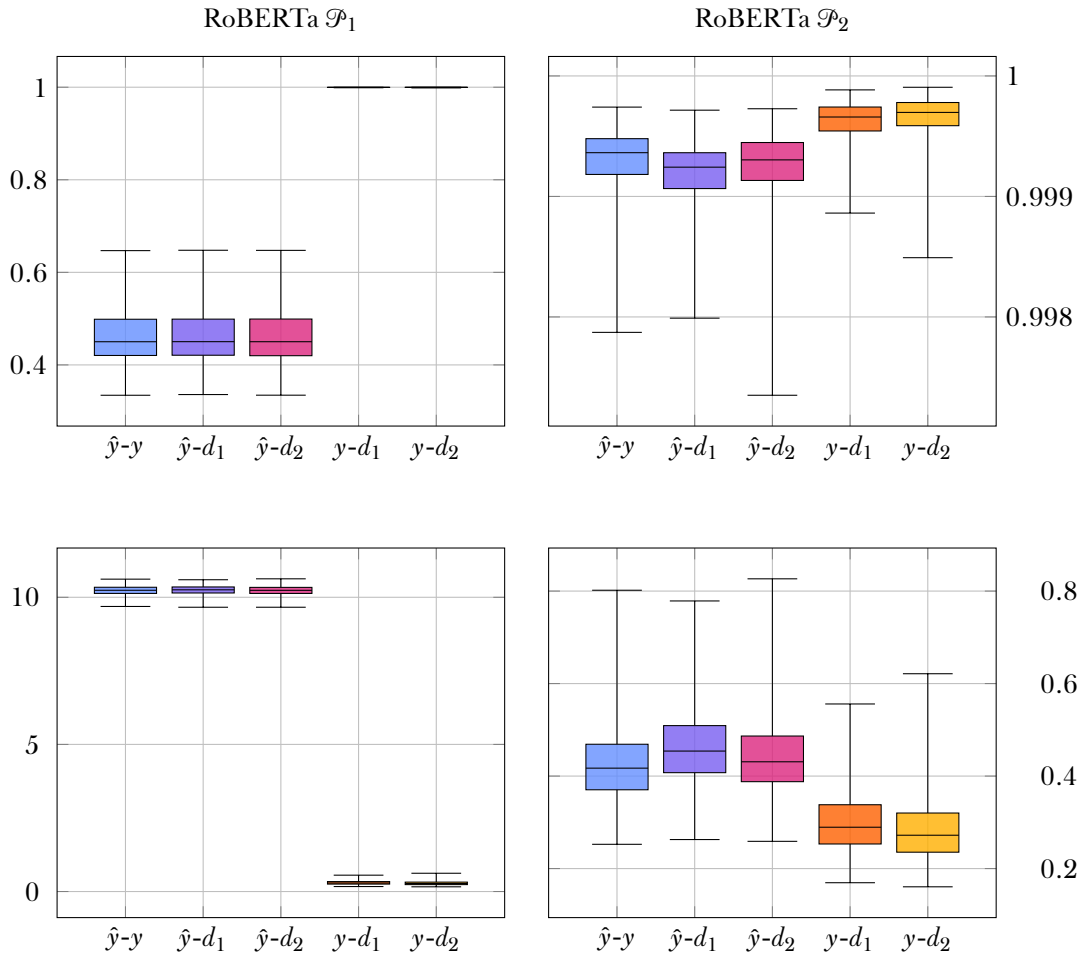

Figure 7: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathcal{P}_1$ (left) and the second probe $\mathcal{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathcal{P}_1$ (left) and $\mathcal{P}_2$ (right).

When considering cosine similarity, the average scores for the three comparisons $\hat{y}$-y, $\hat{y}$-d_1 and $\hat{y}$-d_2 are nearly identical across both probes. For $\mathcal{P}_1$, the pair $\hat{y}$-d_1 exhibits a marginally higher similarity than $\hat{y}$-y and $\hat{y}$-d_2, with an average score of 0.4727. However, for $\mathcal{P}_2$, the results are so close that it is difficult to determine which comparison is the best. Euclidean distance provides a clearer perspective. In the case of $\mathcal{P}_2$, $\hat{y}$ is closer to the target than to the distractors, with an average distance of 0.43. Turning to $\mathcal{P}_1$, $\hat{y}$ appears to be slightly closer to d_2 than to d_1 and y, which goes against the cosine similarity results. As shown in Figure 7, the similarity between $\hat{y}$ and the target is lower than that observed for the pairs y-d_1 and y-d_2, for both probes and metrics.

5.1.6 Sentence Transformer Agreement Results

This section returns to the Sentence Trasformer model and outlines the results for the English agreement dataset. Similarity scores for both probes are summarized in Table 5.6.

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	0.6913 ± 0.0946	0.5921 ± 0.0835	0.6025 ± 0.0883	0.8739 ± 0.0062	0.8884 ± 0.0064
$\mathcal{P}_2$	0.3330 ± 0.0993	0.2922 ± 0.0921	0.3048 ± 0.0969	0.8739 ± 0.0062	0.8884 ± 0.0064

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	2.2755 ± 0.2923	2.5833 ± 0.2297	2.5570 ± 0.2447	1.5544 ± 0.3992	1.4503 ± 0.4284
$\mathcal{P}_2$	3.0063 ± 0.1326	3.0402 ± 0.1332	3.0350 ± 0.1321	1.5544 ± 0.3992	1.4503 ± 0.4284

Table 5.6: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathcal{P}_1$ and $\mathcal{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

For both probes, the pair $\hat{y}$-y scores a higher similarity than $\hat{y}$-d_1 and $\hat{y}$-d_2. In particular, when examining $\mathcal{P}_1$, $\hat{y}$ is markedly closer to the target than to the distractors, with average cosine similarity and Euclidean distance of 0.6913 and 2.2755, respectively. Turning to $\mathcal{P}_2$, the difference in similarity between $\hat{y}$-y, $\hat{y}$-d_1 and $\hat{y}$-d_2 is less pronounced and the scores, in particular with respect to cosine similarity, are significantly lower. Nonetheless, $\hat{y}$ is closer to the target than to the two distractors, with an average cosine similarity and Euclidean distance of 0.333 and 3.0063, respectively. As shown in Figure 8, the target y remains consistently closer to d_1 and d_2 than to $\hat{y}$ across both probes and metrics.

5.2 Italian

5.2.1 BERT Tense Results

This section describes the results for the Italian tense dataset with respect to BERT. Table 5.7 reports the similarity scores for all the embedding pairs obtained from the `CLS` token in BERT's final layer. A comprehensive overview, covering all of the model's layers is provided in the Appendix (see Table A.53, Table A.54, Table A.55 and Table A.56).

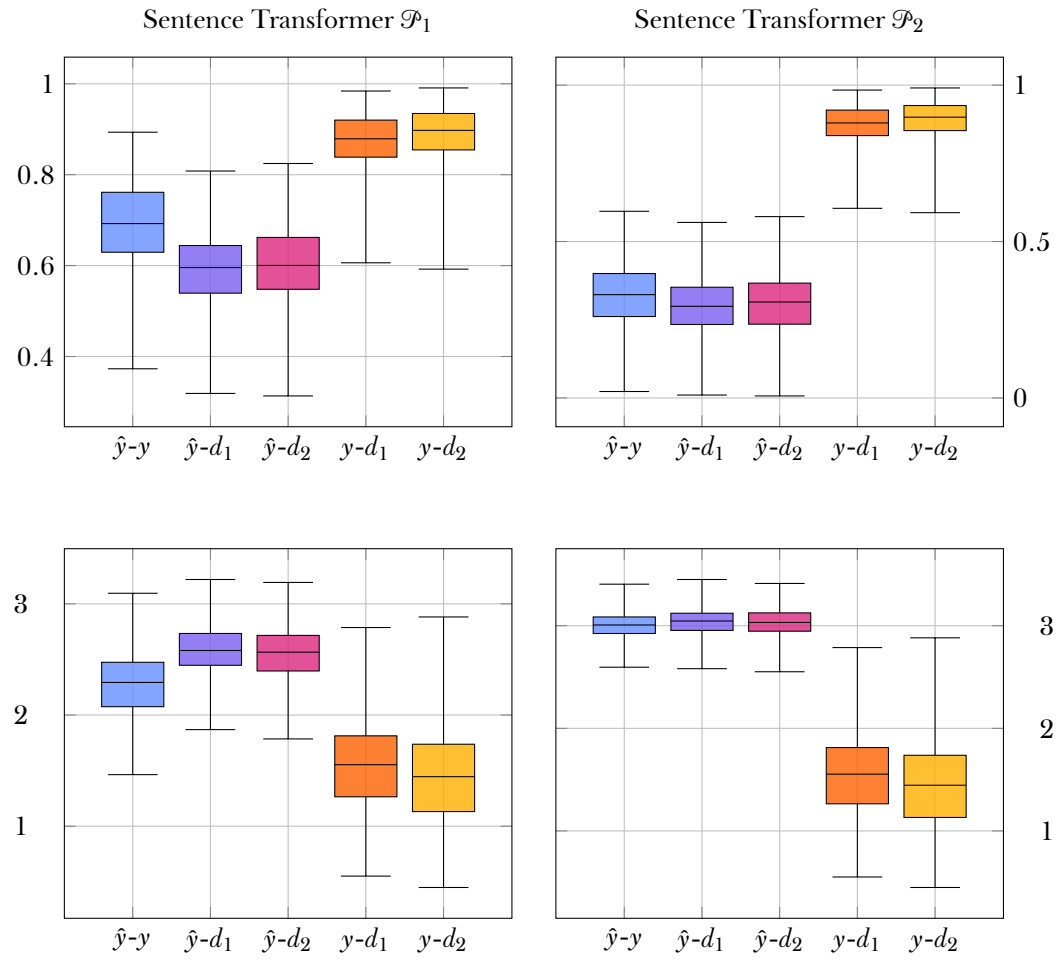

Figure 8: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathcal{P}_1$ (left) and the second probe $\mathcal{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathcal{P}_1$ (left) and $\mathcal{P}_2$ (right).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	0.7958 ± 0.1248	0.7322 ± 0.1477	0.7343 ± 0.1452	0.8446 ± 0.1244	0.8523 ± 0.1271
$\mathcal{P}_2$	0.8508 ± 0.0903	0.8037 ± 0.1114	0.8016 ± 0.1154	0.8446 ± 0.1244	0.8523 ± 0.1271

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	5.4168 ± 1.4592	5.6789 ± 1.3979	5.5829 ± 1.3917	4.812 ± 2.114	4.5978 ± 2.1659
$\mathcal{P}_2$	4.8267 ± 1.5438	5.1376 ± 1.5137	5.1066 ± 1.5778	4.812 ± 2.114	4.5978 ± 2.1659

Table 5.7: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathcal{P}_1$ and $\mathcal{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

Across both probes and metrics, the transformation $\hat{y}$ is closer to the target than to the distractors. Specifically, for $\mathcal{P}_1$, the pair $\hat{y}$-y registers an average cosine similarity and Euclidean distance of 0.7958 and 5.4168, respectively. For $\mathcal{P}_2$, the two scores are 0.8508 and 4.8267. Interestingly, $\hat{y}$-y registers a higher cosine similarity than y-d_1 in the case of $\mathcal{P}_2$, although this is hardly noticeable from the medians depicted in Figure 9. This trend is not reflected in the Euclidean distance scores for $\mathcal{P}_2$. Finally,

turning back to $\mathcal{P}_1$, the target exhibits higher similarity scores with d_1 and d_2 across both metrics.

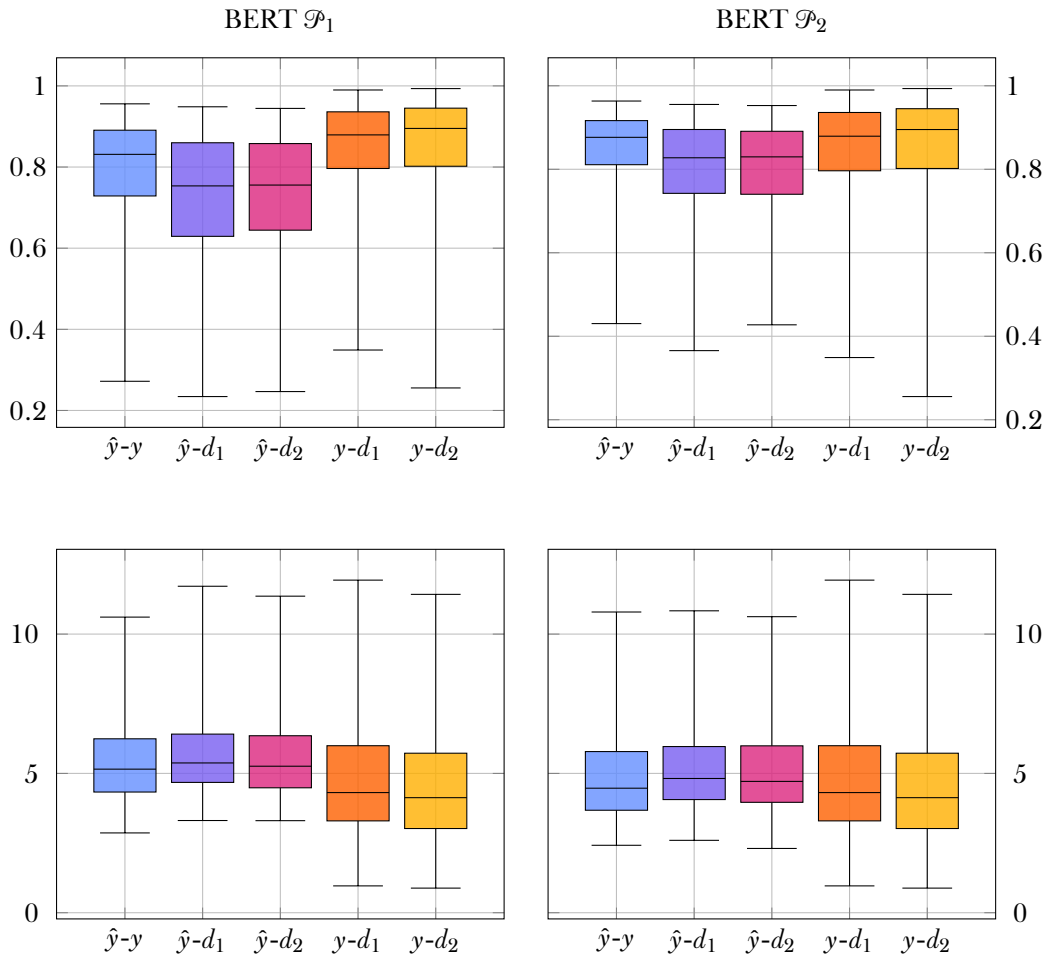

Figure 9: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathcal{P}_1$ (left) and the second probe $\mathcal{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathcal{P}_1$ (left) and $\mathcal{P}_2$ (right).

5.2.2 RoBERTa Tense Results

This section reports on the results for the Italian tense dataset with respect to RoBERTa. Table 5.8 summarizes the similarity scores for all the embedding pairs obtained from the `CLS` token in the model's final layer. A comprehensive overview of all layers is provided in the Appendix (see Table A.57, Table A.58, Table A.59 and Table A.60).

Overall, the similarity scores for the first three comparisons are considerably close, across both probes and metrics. Nevertheless, the first comparison $\hat{y}$-y registers a higher similarity than $\hat{y}$-d_1 and $\hat{y}$-d_2, in both probes. In the case of $\mathcal{P}_1$, the average cosine similarity between the transformation and the target is 0.8609, while the average Euclidean distance is 2.9717. For $\mathcal{P}_2$, the two average scores are 0.9545 and 1.7113, respectively. As portrayed in in Figure 10, the target y remains closer to d_1 and d_2 than to $\hat{y}$.

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	0.8609 ± 0.0551	0.8587 ± 0.0459	0.8572 ± 0.0477	0.9749 ± 0.0372	0.9723 ± 0.0378
$\mathcal{P}_2$	0.9545 ± 0.0348	0.9532 ± 0.0165	0.9506 ± 0.0216	0.9749 ± 0.0372	0.9723 ± 0.0378

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	2.9717 ± 0.3568	3.0018 ± 0.2914	3.0102 ± 0.3027	1.2108 ± 0.0519	1.2772 ± 0.5304
$\mathcal{P}_2$	1.7113 ± 0.3772	1.7568 ± 0.2534	1.7939 ± 0.2923	1.2108 ± 0.0519	1.2772 ± 0.5304

Table 5.8: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathcal{P}_1$ and $\mathcal{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

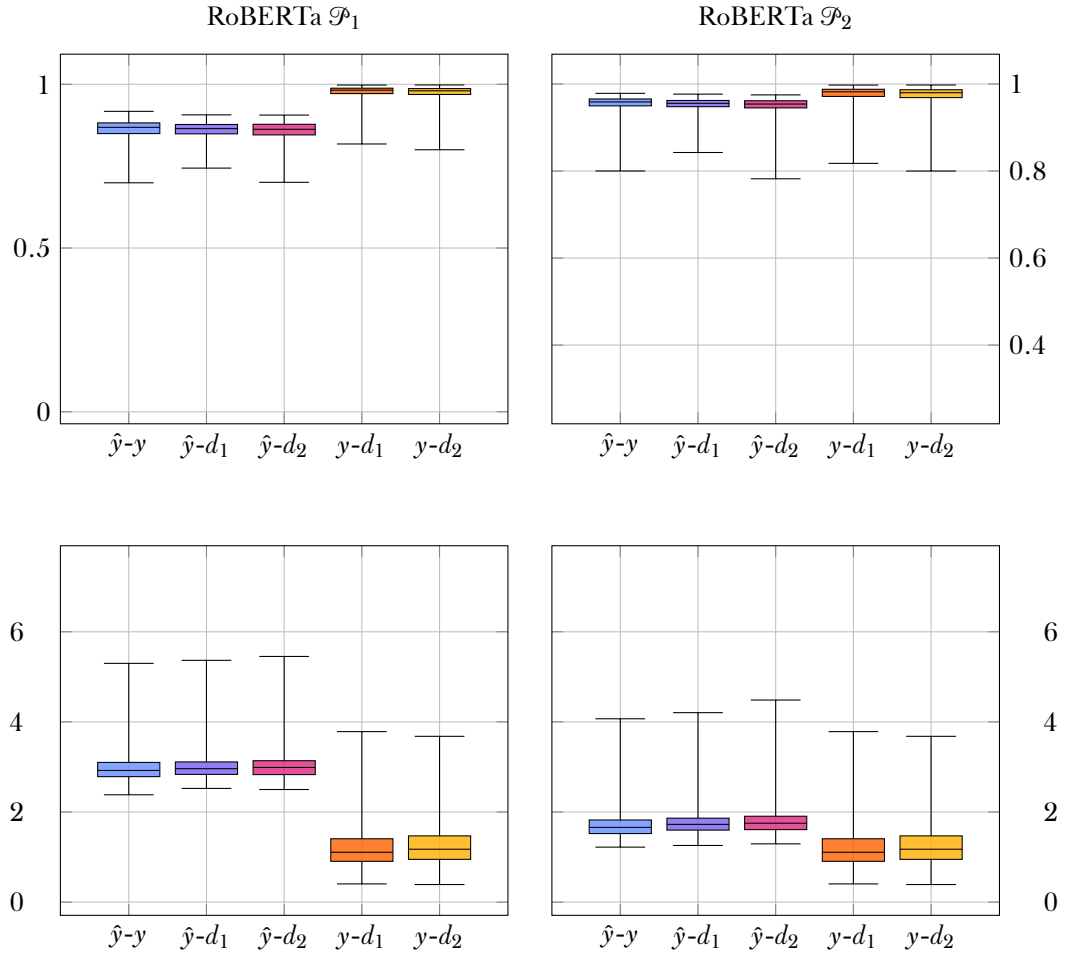

Figure 10: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathcal{P}_1$ (left) and the second probe $\mathcal{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathcal{P}_1$ (left) and $\mathcal{P}_2$ (right).

5.2.3 Sentence Transformer Tense Results

This section outlines the results for the Italian tense dataset with respect to the Sentence Trasformer model. Table 5.9 summarizes the similarity scores for all embedding pairs, covering both probes and metrics.

For both probes, the similarity scored by the pair $\hat{y}$-y is higher than that registered

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	0.6983 ± 0.1058	0.6499 ± 0.0949	0.6698 ± 0.0995	0.9104 ± 0.0598	0.9281 ± 0.0534
$\mathcal{P}_2$	0.7650 ± 0.0806	0.7145 ± 0.0075	0.735 ± 0.077	0.9104 ± 0.0598	0.9281 ± 0.0534

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	1.7996 ± 0.3789	1.9091 ± 0.3422	1.8322 ± 0.0357	1.0129 ± 0.3351	0.8970 ± 0.3268
$\mathcal{P}_2$	1.6234 ± 0.3438	1.7488 ± 0.3093	1.6670 ± 0.3214	1.0129 ± 0.3351	0.8970 ± 0.3268

Table 5.9: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathcal{P}_1$ and $\mathcal{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

by $\hat{y}$-d_1 and $\hat{y}$-d_2, as clearly illustrated in Figure 11. In the case of $\mathcal{P}_1$, the average cosine similarity is 0.6983, and the Euclidean distance score is 1.7996. On the other hand, for $\mathcal{P}_1$, the corresponding scores are 0.765 and 1.6234, respectively. In both cases, however, the target y is significantly closer to d_1 and d_2 than to $\hat{y}$.

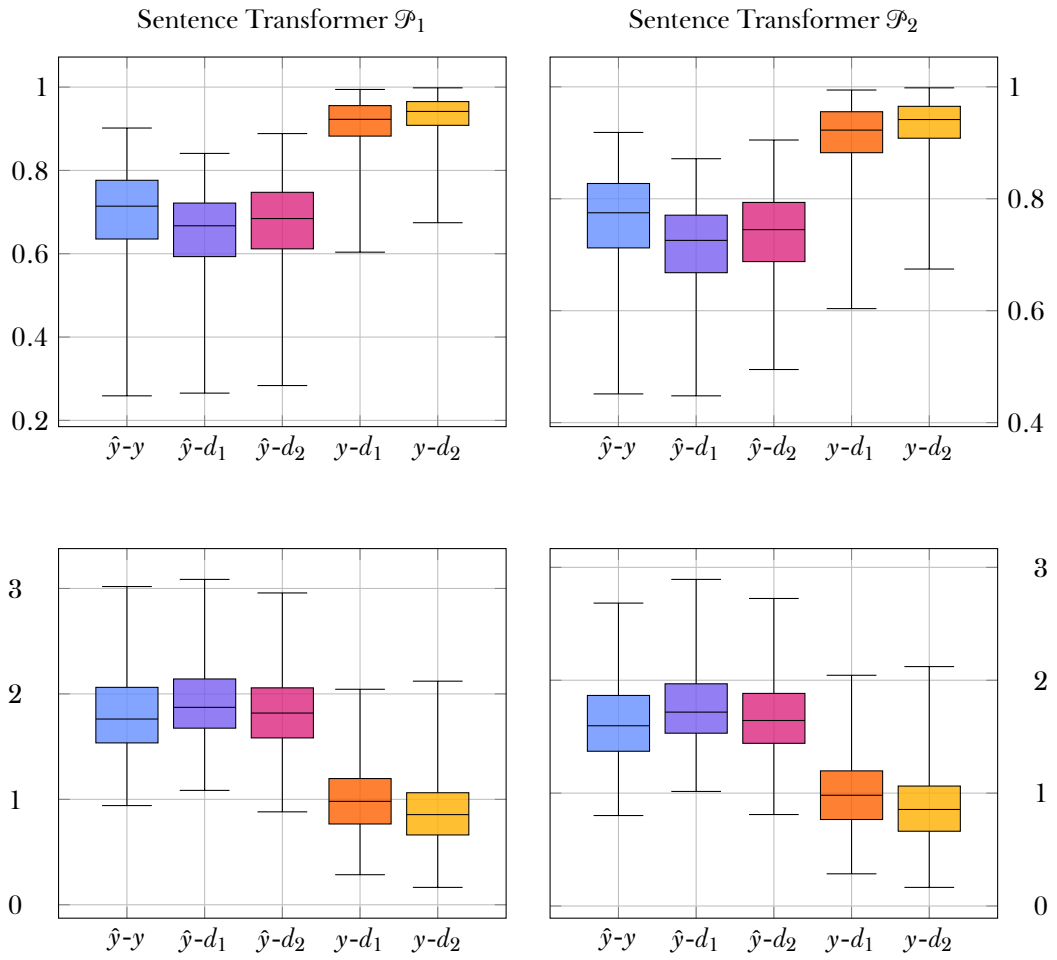

Figure 11: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathcal{P}_1$ (left) and the second probe $\mathcal{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathcal{P}_1$ (left) and $\mathcal{P}_2$ (right).

5.2.4 BERT Agreement Results

This section comes back to BERT to describe the results for the Italian agreement dataset. Table 5.10 collects the similarity scores for all pairs of embeddings across both probes. An overview of the results obtained for each layer is presented in the Appendix (see Table A.61, Table A.62, Table A.63 and Table A.64).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	0.7601 ± 0.1161	0.7033 ± 0.1351	0.6843 ± 0.1491	0.7739 ± 0.1682	0.7563 ± 0.1793
$\mathcal{P}_2$	0.8367 ± 0.0941	0.7845 ± 0.1221	0.7611 ± 0.1347	0.7739 ± 0.1682	0.7563 ± 0.1793

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	5.4438 ± 1.5626	5.8417 ± 1.6362	6.2832 ± 1.9404	5.5343 ± 2.6616	5.8744 ± 2.7034
$\mathcal{P}_2$	4.6092 ± 1.6326	5.1607 ± 1.8439	5.6487 ± 2.1204	5.5343 ± 2.6616	5.8744 ± 2.7034

Table 5.10: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathcal{P}_1$ and $\mathcal{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

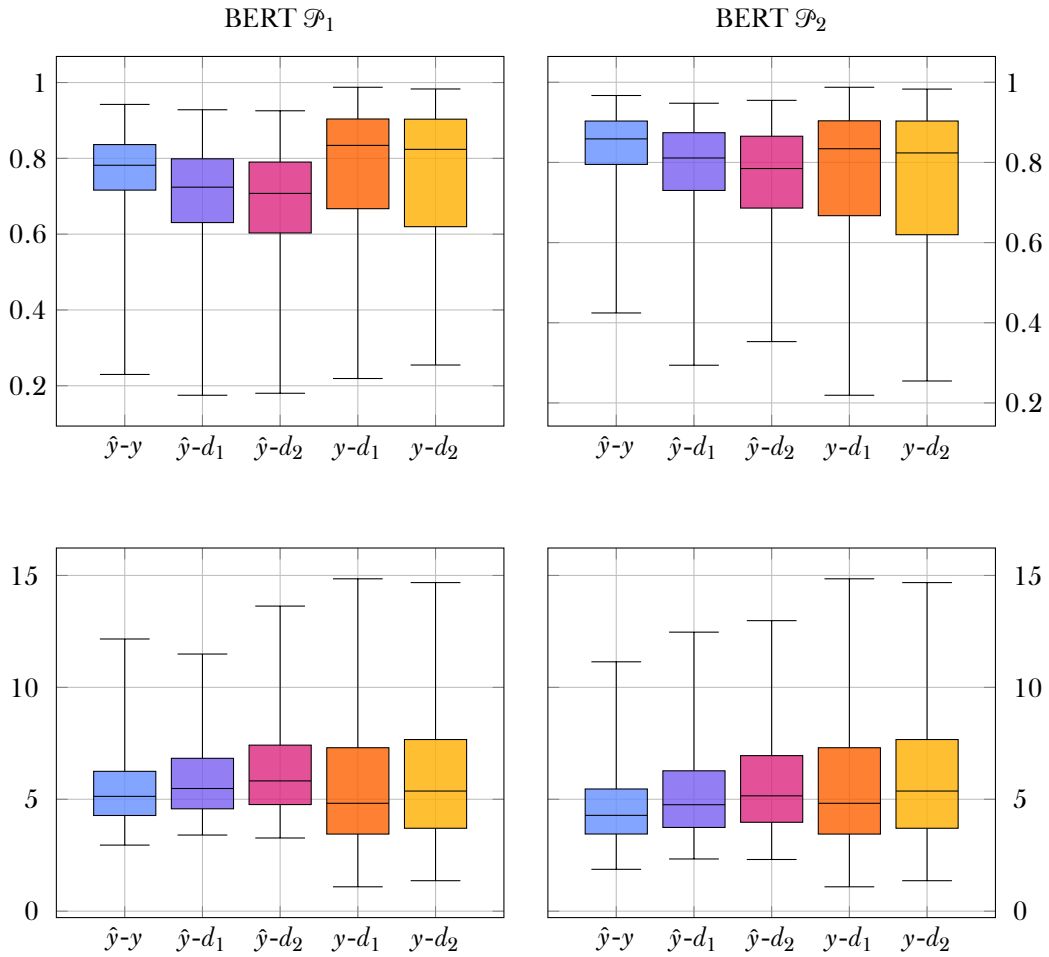

Figure 12: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathcal{P}_1$ (left) and the second probe $\mathcal{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathcal{P}_1$ (left) and $\mathcal{P}_2$ (right).

Looking at $\mathcal{P}_1$, the first comparison $\hat{y}$-y exhibits a higher average cosine similarity (0.7601) than $\hat{y}$-d_1, $\hat{y}$-d_2 and y-d_2, while the target appears to remain closer to the first distractor d_1. On the other hand, $\hat{y}$-y scores the best Euclidean distance (5.4438) among all five comparisons.

These results, however, are not clearly confirmed by Figure 12. From the box-plots, the distributions appear considerably skewed across both metrics. Moreover, for y-d_1 and y-d_1, the results variation is higher than that in the other comparisons. In this situation, the median scores may offer a more accurate picture. With a median cosine similarity of 0.7822, the pair $\hat{y}$-y surpasses both $\hat{y}$-d_1 and $\hat{y}$-d_2. However, the last two comparisons, y-d_1 and y-d_2, exhibit higher median scores of 0.8342 and 0.8238, respectively. Similarly, the median Euclidean distances paint a different picture. The transformation $\hat{y}$ is closer to the target than to the distractors, exhibiting a distance of 5.1295. However, while y-d_2 scores a higher value (5.3965), this is not the case for y-d_1 (4.8278).

Turning to $\mathcal{P}_2$, $\hat{y}$ and y are the closest representations, with an average cosine similarity and Euclidean distance of 0.8367 and 4.6092, respectively. These results are supported by the plots.

For this experiment, it is also worth considering the remaining layers. In particular, looking at $\mathcal{P}_1$, Table A.61 shows that, for the first BERT layer, $\hat{y}$ is closer to the target y, with an average cosine similarity of 0.9872. This result is confirmed by the Euclidean distance score of 1.9481 reported in Table A.63 for the same layer. Moreover, turning to the tenth layer, the same table indicates an average Euclidean distance of 5.3161, suggesting that $\hat{y}$ and y are closer than y and d_2 (5.4926).

Of particular interest is the observation pertaining to the non-linear transformation generate by $\mathcal{P}_2$. With respect to both cosine similarity (Table A.62) and Euclidean distance (Table A.64), the comparison $\hat{y}$-y exhibits the highest score across all layers.

5.2.5 RoBERTa Agreement Results

This section returns to RoBERTa to present the results on the Italian agreement dataset. Table 5.11 summarizes the similarity scores for all embedding pairs. A complete overview covering all layers is provided in the Appendix (see Table A.65, Table A.66, Table A.67 and Table A.68).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	0.9347 ± 0.0123	0.9361 ± 0.0876	0.9387 ± 0.0982	0.9474 ± 0.0013	0.9490 ± 0.1381
$\mathcal{P}_2$	0.9403 ± 0.1126	0.9438 ± 0.0548	0.9445 ± 0.0883	0.9474 ± 0.0013	0.9490 ± 0.1381

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	1.8521 ± 0.8929	1.9373 ± 0.6401	1.8724 ± 0.7093	1.5541 ± 1.1323	1.4949 ± 1.1736
$\mathcal{P}_2$	1.7828 ± 0.0839	1.8645 ± 0.5381	1.8002 ± 0.6551	1.5541 ± 1.1323	1.4949 ± 1.1736

Table 5.11: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathcal{P}_1$ and $\mathcal{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

With respect to cosine similarity, the average scores for the three comparisons $\hat{y}$-y, $\hat{y}$-d_1 and $\hat{y}$-d_2 are very similar for both probes. In the case of $\mathcal{P}_1$, $\hat{y}$ shows a slightly lower similarity with y compared to d_1 and d_2, with an average score of 0.9347. For $\mathcal{P}_2$, the results follow the same trend, with $\hat{y}$-y scoring an average cosine similarity of 0.9403. However, these results are not fully supported by the cosine similarity boxplots in Figure 13. In particular, for $\mathcal{P}_1$, the respective median cosine similarity for the three comparisons are 0.9596, 0.951 and 0.9551. Turning to $\mathcal{P}_2$, the median scores are 0.9624, 0.9538 and 0.9581. Despite appearing somewhat skewed, the Euclidean distance measurements align with the median scores, indicating that for both probes the comparison $\hat{y}$-y achieves an higher similarity than $\hat{y}$-d_1 and $\hat{y}$-d_2. Nevertheless, both for $\mathcal{P}_1$ and $\mathcal{P}_2$, the target y is closer to the distractors.

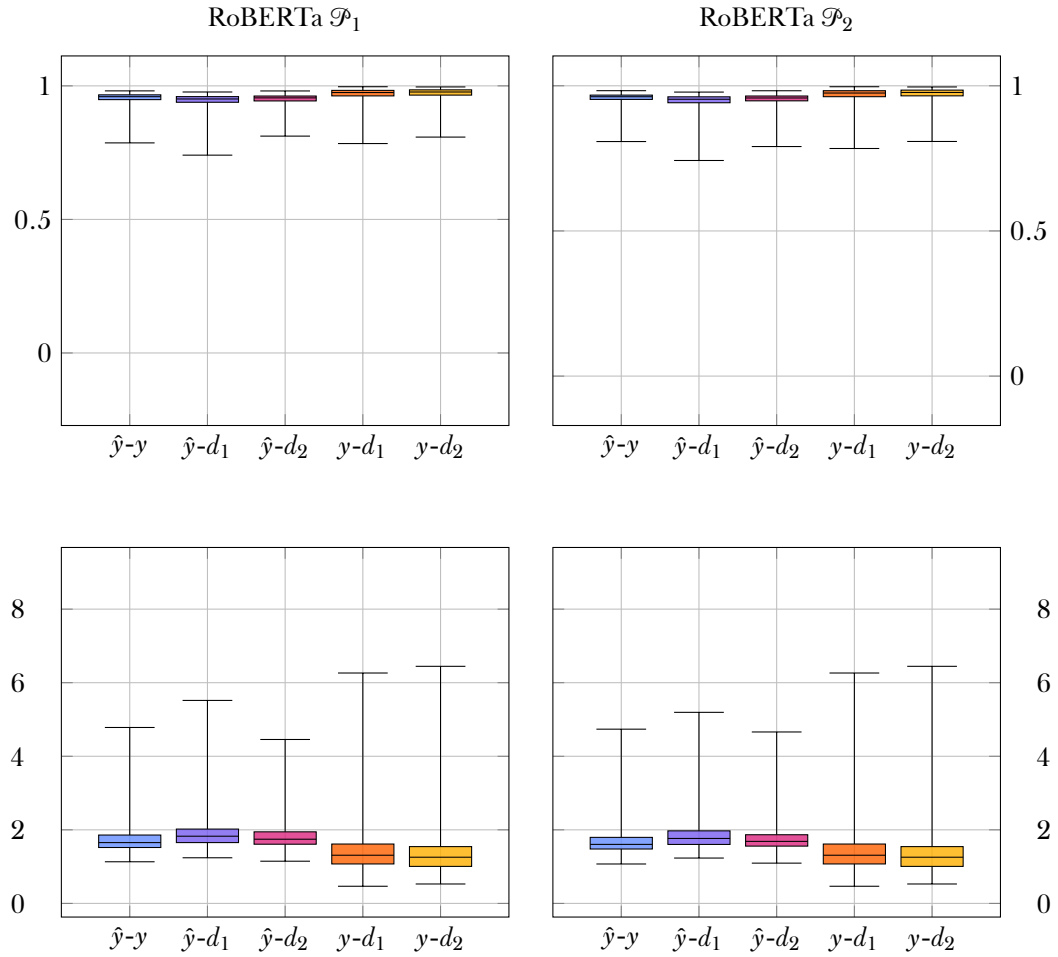

Figure 13: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathcal{P}_1$ (left) and the second probe $\mathcal{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathcal{P}_1$ (left) and $\mathcal{P}_2$ (right).

A closer look at the remaining layers reveals a few peculiar trends for $\mathcal{P}_2$. In particular, with respect to cosine similarity, Table A.66 shows that $\hat{y}$-y obtains an higher average score than y-d_2 in the eleventh layer and surpasses y-d_1 in the tenth and ninth layers. The respective median similarity scores, however, calls into question these results. In the eleventh layer, $\hat{y}$-y scores a lower median (0.9979) than both y-d_1 and y-d_2 (0.9994). In the tenth layer, $\hat{y}$-y, y-d_1 and y-d_2 exhibit a median similarity of 0.9802, 0.9846 and 0.9861, respectively. Finally, in the ninth layer, the

three respective median similarities amount to 0.9527, 0.9593 and 0.9629.

5.2.6 Sentence Transformer Agreement Results

This section presents the Sentence Transformer model results on the Italian agreement dataset. Similarity scores are summarized in Table 5.12.

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathscr{P}_1$	0.6261 ± 0.1399	0.5199 ± 0.1227	0.5459 ± 0.0126	0.8220 ± 0.0971	0.8546 ± 0.0771
$\mathscr{P}_2$	0.8082 ± 0.0874	0.6693 ± 0.0898	0.6971 ± 0.0833	0.8220 ± 0.0971	0.8546 ± 0.0771

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathscr{P}_1$	1.9524 ± 0.4719	2.1434 ± 0.0412	2.1027 ± 0.4227	1.4235 ± 0.3758	1.2961 ± 0.3119
$\mathscr{P}_2$	1.4628 ± 0.4045	1.8444 ± 0.3318	1.7842 ± 0.3283	1.4235 ± 0.3758	1.2961 ± 0.3119

Table 5.12: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathscr{P}_1$ and $\mathscr{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

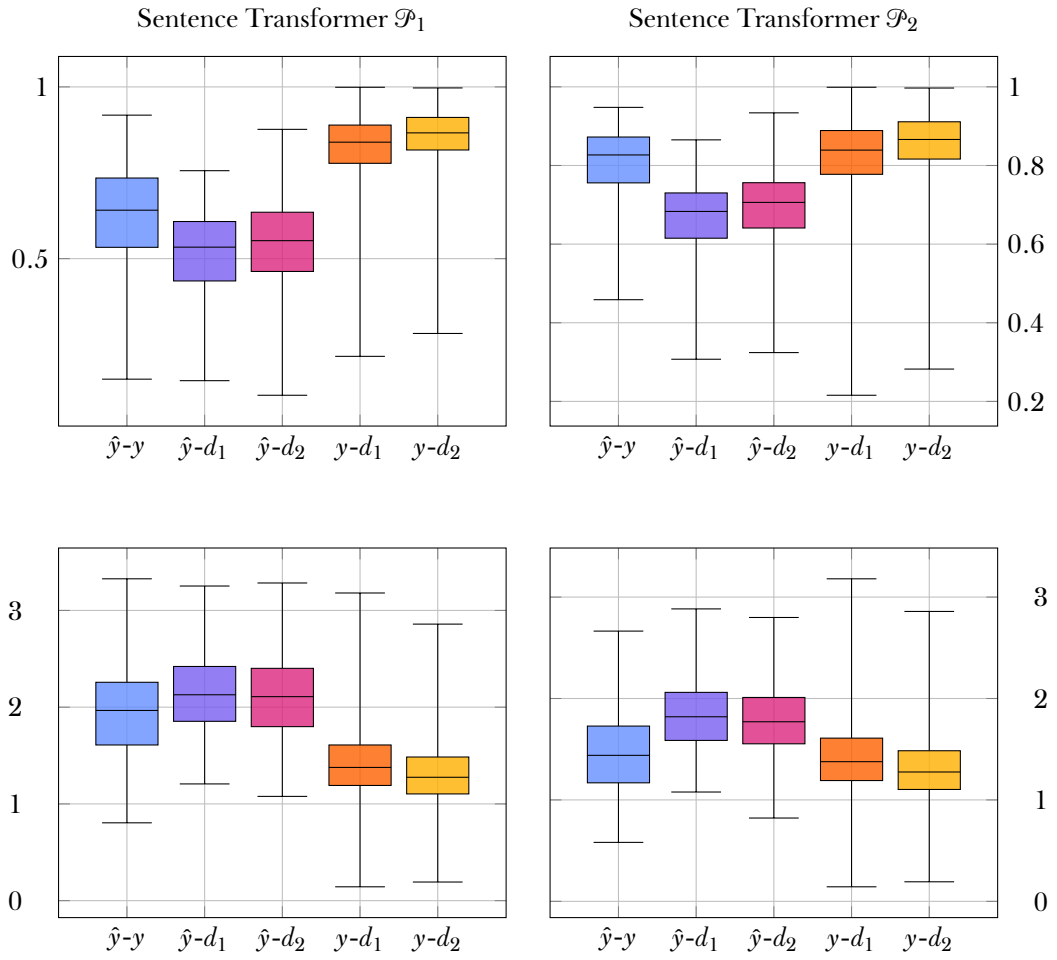

Figure 14: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathscr{P}_1$ (left) and the second probe $\mathscr{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathscr{P}_1$ (left) and $\mathscr{P}_2$ (right).

As portrayed in Figure 14, the comparison $\hat{y}$-y achieves a markedly higher similarity score than $\hat{y}$-d_1 and $\hat{y}$-d_2 in both probes. Specifically, when examining $\mathcal{P}_1$, $\hat{y}$ is closer to y than to the distractors, with an average cosine similarity and Euclidean distance of 0.6261 and 1.9524, respectively. Turning to $\mathcal{P}_2$, the difference in similarity between $\hat{y}$-y, $\hat{y}$-d_1 and $\hat{y}$-d_2 is more pronounced. The pair $\hat{y}$-y scores an average cosine similarity of 0.8082 and an Euclidean distance of 1.4628. Nevertheless, despite the promising results, the target y exhibits a greater similarity to d_1 and d_2 across both probes.

5.3 German

5.3.1 BERT Tense Results

This section describes the results for the German tense dataset with respect to BERT. Table 5.13 presents the similarity scores of all the embedding pairs for the final BERT layer. A comprehensive overview covering all layers is provided in the Appendix (see Table A.69, Table A.70, Table A.71 and Table A.72).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	0.9306 ± 0.0286	0.8948 ± 0.0351	0.9049 ± 0.0341	0.9151 ± 0.0431	0.9218 ± 0.0415
$\mathcal{P}_2$	0.9210 ± 0.0264	0.8964 ± 0.0308	0.9083 ± 0.0288	0.9151 ± 0.0431	0.9218 ± 0.0415

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	7.1654 ± 1.2778	8.7077 ± 0.1361	8.2758 ± 0.1352	7.8640 ± 1.9187	7.5265 ± 1.8929
$\mathcal{P}_2$	7.6078 ± 1.2105	8.6605 ± 0.1263	8.1485 ± 1.2084	7.8640 ± 1.9187	7.5265 ± 1.8929

Table 5.13: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathcal{P}_1$ and $\mathcal{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

Both for $\mathcal{P}_1$ and $\mathcal{P}_2$, the transformation $\hat{y}$ is closer to the target than to the distractors. For the first probe, the comparison $\hat{y}$-y scores an average cosine similarity of 0.9306 and an Euclidean distance of 7.1654. Turning to the second probe, the two metrics are 0.921 and 7.6078, respectively. In the case of $\mathcal{P}_1$, the average scores for $\hat{y}$-y are the highest across all comparisons. This is also confirmed by the corresponding box plots shown in Figure 15. The same trend is not observed for $\mathcal{P}_2$, where the pair y-d_2 scores the highest similarity.

When considering the remaining layers, the affine transformation $\hat{y}$ produced by $\mathcal{P}_1$ exhibits the highest cosine similarity with the target y in the fifth layer (Table A.69). This result is confirmed by the corresponding Euclidean distance score reported in Table A.71. Turning to $\mathcal{P}_2$, $\hat{y}$-y achieves the highest average cosine similarity and Euclidean distance in the ninth layer, as reported in Table A.70 and Table A.72, respectively.

5.3.2 RoBERTa Tense Results

This section describes the RoBERTa results for the German tense dataset. Table 5.14 collects the similarity scores of every embedding pair for both probes.

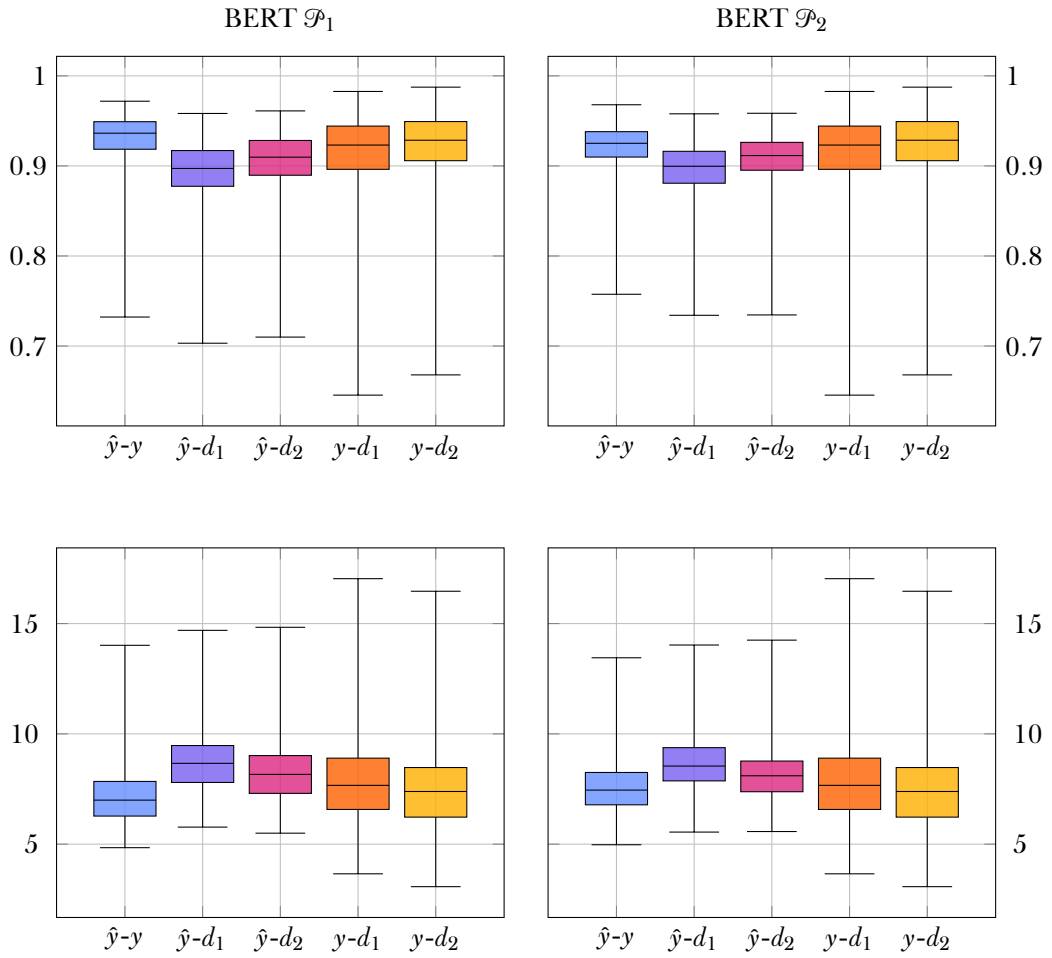

Figure 15: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathscr{P}_1$ (left) and the second probe $\mathscr{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathscr{P}_1$ (left) and $\mathscr{P}_2$ (right).

An overview of the results obtained for each layer is presented in the Appendix (see Table A.73, Table A.74, Table A.75 and Table A.76).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathscr{P}_1$	0.6116 ± 0.0641	0.6054 ± 0.0649	0.6087 ± 0.0654	0.9964 ± 0.0013	0.9963 ± 0.0122
$\mathscr{P}_2$	0.8687 ± 0.0251	0.8662 ± 0.0244	0.8684 ± 0.0241	0.9964 ± 0.0013	0.9963 ± 0.0122

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathscr{P}_1$	11.4892 ± 0.2696	11.4253 ± 0.0269	11.4062 ± 0.2696	0.8608 ± 0.6957	0.8954 ± 0.6588
$\mathscr{P}_2$	10.1257 ± 0.2935	10.0543 ± 0.0285	10.0351 ± 0.2912	0.8608 ± 0.6957	0.8954 ± 0.6588

Table 5.14: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathscr{P}_1$ and $\mathscr{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

Overall, the similarity values for both probes and measures are very close. When examining the first probe, the pair $\hat{y}$-y exhibits a slightly higher cosine similarity than $\hat{y}$-d_1 and $\hat{y}$-d_2, with an average score of 0.6116. A similar pattern is observed

in $\mathcal{P}_2$, with an average cosine similarity of 0.8687. However, the Euclidean distance scores do not follow this trend. Among the three comparisons ($\hat{y}$-y, $\hat{y}$-d_1 and $\hat{y}$-d_2), the last one yields the best result for both probes. Nonetheless, as clearly shown in Figure 16, the target y remains significantly closer to the distractors than to the transformations generated by $\mathcal{P}_1$ and $\mathcal{P}_2$.

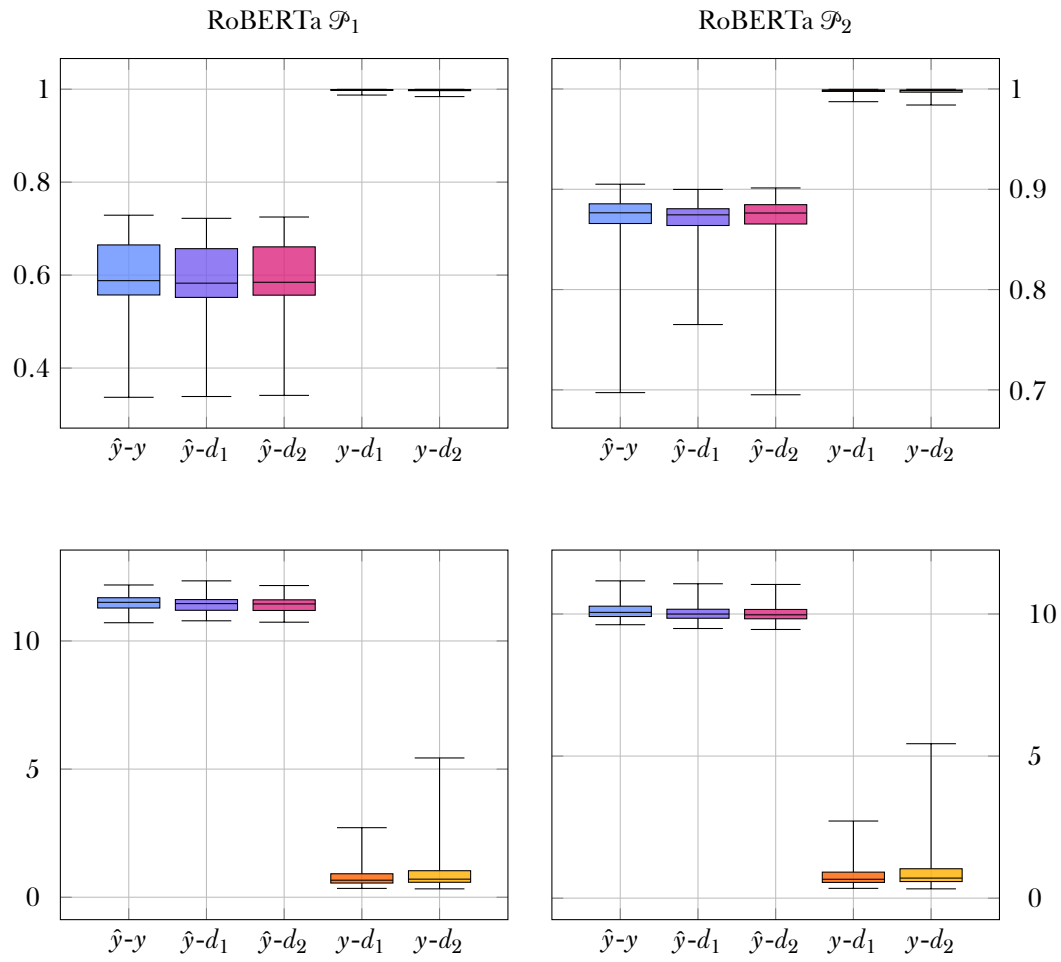

Figure 16: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathcal{P}_1$ (left) and the second probe $\mathcal{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathcal{P}_1$ (left) and $\mathcal{P}_2$ (right).

Looking at the remaining layers, $\mathcal{P}_2$ registers an interesting result in the sixth one, where $\hat{y}$-y achieves an average cosine similarity of 0.9993, exceeding all other comparisons (Table A.74). This result is consistent with the Euclidean distance (0.6061) (Table A.76).

5.3.3 Sentence Transformer Tense Results

This section considers the Sentence Trasformer model results on the German tense dataset. Table 5.15 summarizes the similarity scores for all embedding pairs, covering both probes and metrics.

Across both probes, the transformation $\hat{y}$ is closer to y than to the distractors. In the case of $\mathcal{P}_1$, the average cosine similarity for $\hat{y}$-y is 0.7195, while the Euclidean dis-

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathscr{P}_1$	0.7195 ± 0.0938	0.6715 ± 0.0781	0.694 ± 0.008	0.9157 ± 0.0658	0.9278 ± 0.0539
$\mathscr{P}_2$	0.7816 ± 0.0717	0.7315 ± 0.0597	0.7532 ± 0.0593	0.9157 ± 0.0658	0.9278 ± 0.0539

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathscr{P}_1$	1.6456 ± 0.3756	1.7579 ± 0.3151	1.6668 ± 0.3248	0.9081 ± 0.3258	0.8447 ± 0.2747
$\mathscr{P}_2$	1.4838 ± 0.3422	1.6142 ± 0.2807	1.5210 ± 0.2874	0.9081 ± 0.3258	0.8447 ± 0.2747

Table 5.15: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathscr{P}_1$ and $\mathscr{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

tance is 1.6456. Turning to $\mathscr{P}_2$, the two scores are 0.7816 and 1.4838, respectively. In both cases, the target y is considerably closer to d_1 and d_2 than to $\hat{y}$.

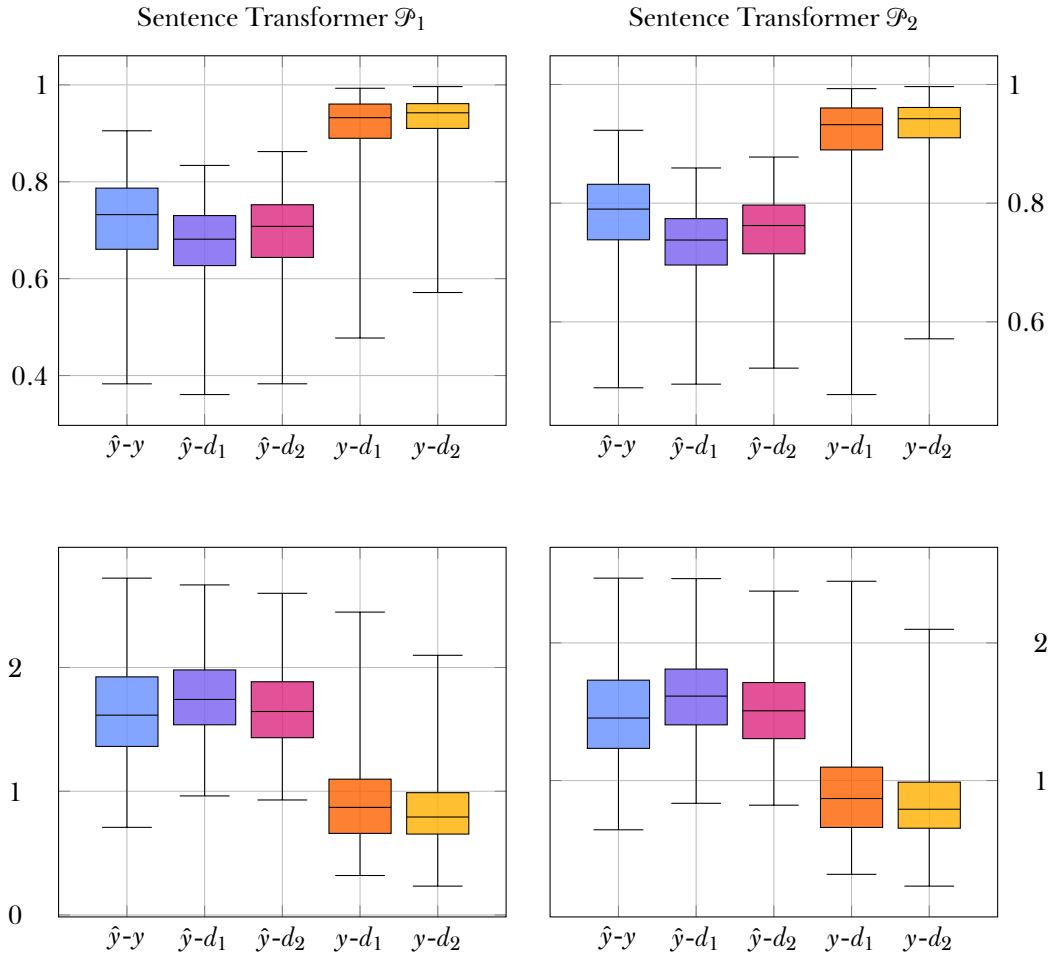

Figure 17: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathscr{P}_1$ (left) and the second probe $\mathscr{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathscr{P}_1$ (left) and $\mathscr{P}_2$ (right).

5.3.4 BERT Agreement Results

This section comes back to BERT to present the results for the German agreement dataset. Table 5.16 collects the similarity scores for all pairs of embeddings across both probes. An overview of the results obtained for each layer is presented in the Appendix (see Table A.77, Table A.78, Table A.79 and Table A.80).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	0.9318 ± 0.0258	0.8787 ± 0.0371	0.8838 ± 0.0337	0.8950 ± 0.0474	0.9036 ± 0.0422
$\mathcal{P}_2$	0.9248 ± 0.0258	0.8814 ± 0.0324	0.8856 ± 0.0003	0.8950 ± 0.0474	0.9036 ± 0.0422

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	7.1467 ± 1.2801	9.4443 ± 1.4381	9.2809 ± 1.3426	8.8372 ± 2.1063	8.4829 ± 1.9814
$\mathcal{P}_2$	7.4671 ± 1.2514	9.3278 ± 1.2924	9.1903 ± 1.2255	8.8372 ± 2.1063	8.4829 ± 1.9814

Table 5.16: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathcal{P}_1$ and $\mathcal{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

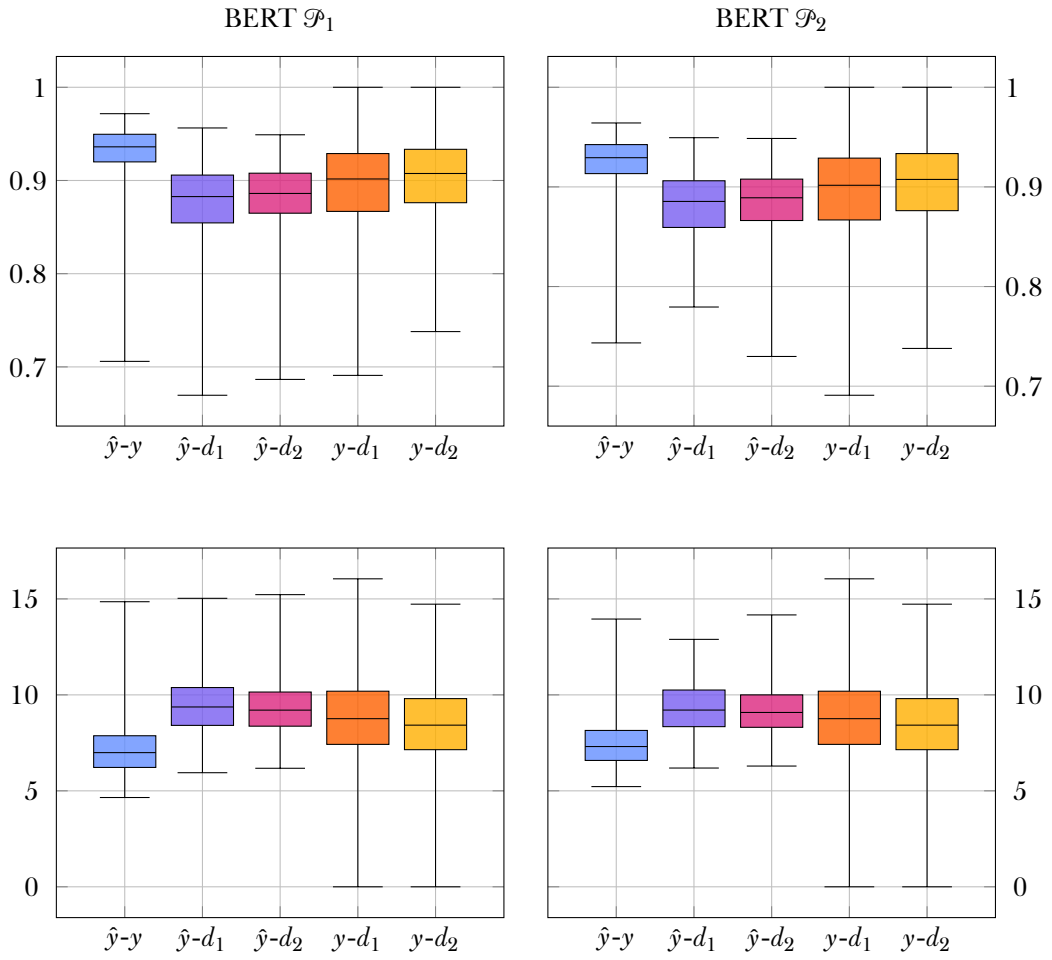

Figure 18: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathcal{P}_1$ (left) and the second probe $\mathcal{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathcal{P}_1$ (left) and $\mathcal{P}_2$ (right).

The results obtained for both probes follow a similar trend, with the comparison $\hat{y}$-y exhibiting the highest average cosine similarity and Euclidean distance scores. In the case of $\mathcal{P}_1$, the average cosine similarity is 0.9318, while the Euclidean distance is 7.1467. Turning to $\mathcal{P}_2$, the two metrics achieve a score of 0.9248 and 7.4671, respectively. The plots in Figure 18 clearly show that, compared to $\mathcal{P}_2$, the affine transformation carried out by $\mathcal{P}_1$ results in a higher similarity between $\hat{y}$ and the target.

The values reported in Table A.77 indicate that, for $\mathcal{P}_1$, the comparison $\hat{y}$-y achieves the highest cosine similarity in the eighth and sixth layers as well, with an average score of 0.9646 and 0.9804, respectively. These results are comparable with the average Euclidean distance values (5.7252 and 4.3414) observed for the same layers in Table A.79.

Turning to the second probe, Table A.78 shows that $\hat{y}$-y achieves the highest average cosine similarity in the eleventh, tenth and ninth layers, with a score of 0.9501, 0.9375 and 0.9536, respectively. In the eighth layer, the score (0.9527) does not exceed the similarity registered between y and d_2 (0.9572). The Euclidean distance results reported in Table A.80 are in line with the previous findings.

5.3.5 RoBERTa Agreement Results

This section returns to RoBERTa to describe the results obtained on the German agreement dataset. Table 5.17 summarizes the similarity scores for all embedding pairs. An overview of all layers is provided in the Appendix (see Table A.81, Table A.82, Table A.83 and Table A.84).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	0.5824 ± 0.0394	0.5798 ± 0.0392	0.5785 ± 0.0399	0.9968 ± 0.0062	0.9972 ± 0.0059
$\mathcal{P}_2$	0.9957 ± 0.0135	0.9948 ± 0.0137	0.9949 ± 0.0015	0.9968 ± 0.0062	0.9972 ± 0.0059

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	11.6617 ± 0.1955	11.6864 ± 0.1913	11.7035 ± 0.1954	0.9264 ± 0.4942	0.8694 ± 0.4498
$\mathcal{P}_2$	1.3383 ± 0.6582	1.4348 ± 0.6484	1.4233 ± 0.0685	0.9264 ± 0.4942	0.8694 ± 0.4498

Table 5.17: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathcal{P}_1$ and $\mathcal{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

In the case of $\mathcal{P}_1$, $\hat{y}$ exhibits a marginally higher similarity with y than with d_1 and d_2, showing an average cosine similarity of 0.5824 and an Euclidean distance score of 11.6617. Overall, the average scores for the first three comparisons are very close to each other across both metrics. Turning to $\mathcal{P}_2$, the average cosine similarity and Euclidean distance results for the first comparison are 0.9957 and 1.3383, respectively. As for $\mathcal{P}_1$, results across both metrics are only marginally different. However, the non-linear transformation performed by the second probe seems to bring $\hat{y}$ much closer to y, d_1 and d_2. Finally, as illustrated in Figure 19, the target exhibits a higher similarity with d_1 and d_2, across both probes.

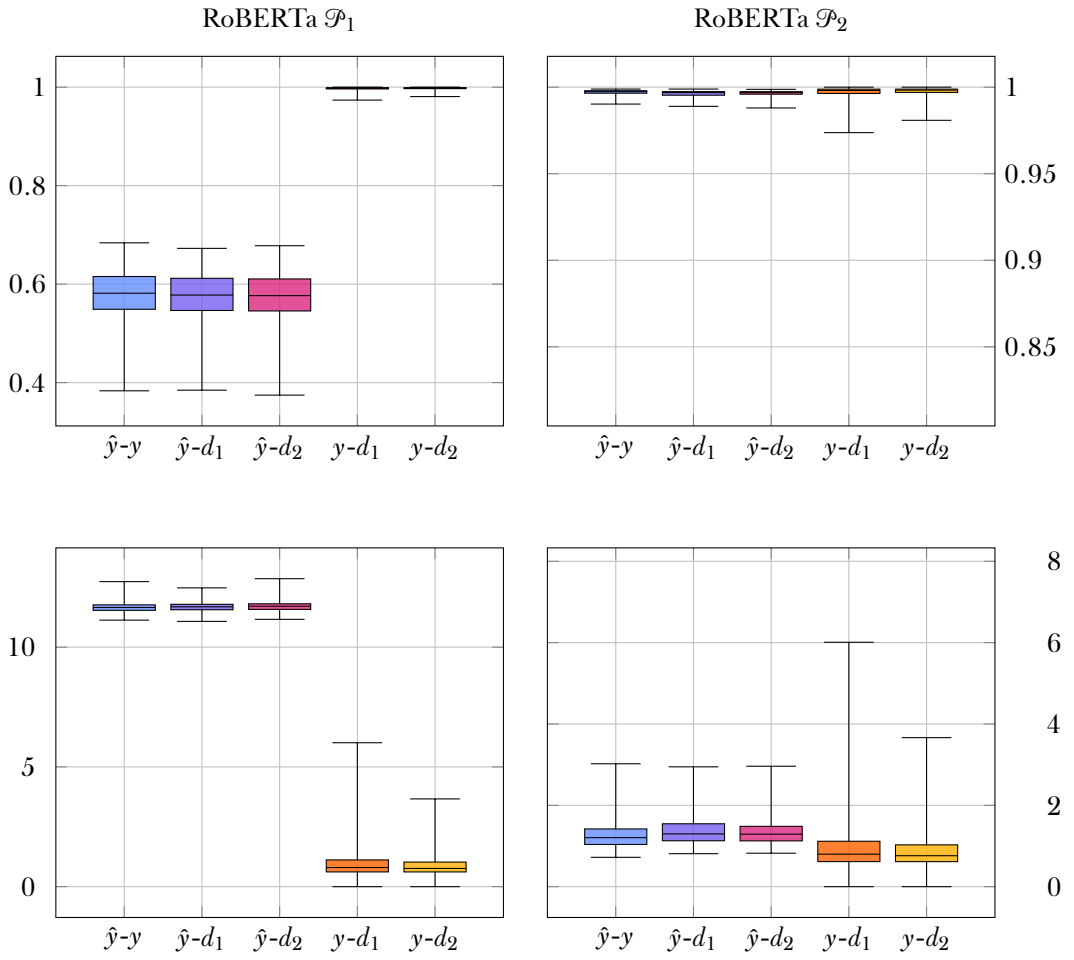

Figure 19: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathcal{P}_1$ (left) and the second probe $\mathcal{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathcal{P}_1$ (left) and $\mathcal{P}_2$ (right).

5.3.6 Sentence Transformer Agreement Results

This section covers the Sentence Transformer model results for the German agreement dataset. Sentence representations are obtained from the model's pooling layer and similarity scores for both probes are summarized in Table 5.18.

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	0.7686 ± 0.0989	0.6646 ± 0.0969	0.7070 ± 0.0987	0.8633 ± 0.0736	0.8970 ± 0.0698
$\mathcal{P}_2$	0.7093 ± 0.1073	0.6116 ± 0.1081	0.6642 ± 0.1058	0.8633 ± 0.0736	0.8970 ± 0.0698

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
$\mathcal{P}_1$	1.4558 ± 0.0402	1.7184 ± 0.3736	1.6229 ± 0.0375	1.1516 ± 0.3365	0.9808 ± 0.3713
$\mathcal{P}_2$	1.6085 ± 0.4096	1.7966 ± 0.3879	1.7043 ± 0.3807	1.1516 ± 0.3365	0.9808 ± 0.3713

Table 5.18: The top table presents the average cosine similarity scores and standard deviations for both probes $\mathcal{P}_1$ and $\mathcal{P}_2$. The bottom table reports the corresponding Euclidean distance scores.

As illustrated in Figure 20, the comparison $\hat{y}$-y achieves a rather higher similarity

score than $\hat{y}$-d_1 and $\hat{y}$-d_2 in both probes. Specifically, when examining $\mathcal{P}_1$, $\hat{y}$ is closer to y than to the distractors, with an average cosine similarity and Euclidean distance of 0.7686 and 1.4558, respectively. Turning to $\mathcal{P}_2$, the difference in similarity between $\hat{y}$-y, $\hat{y}$-d_1 and $\hat{y}$-d_2 is less pronounced. The pair $\hat{y}$-y achieves an average cosine similarity of 0.7093 and an Euclidean distance of 1.6085. Nonetheless, in both probes, the target y remains closer to d_1 and d_2 than any other representation.

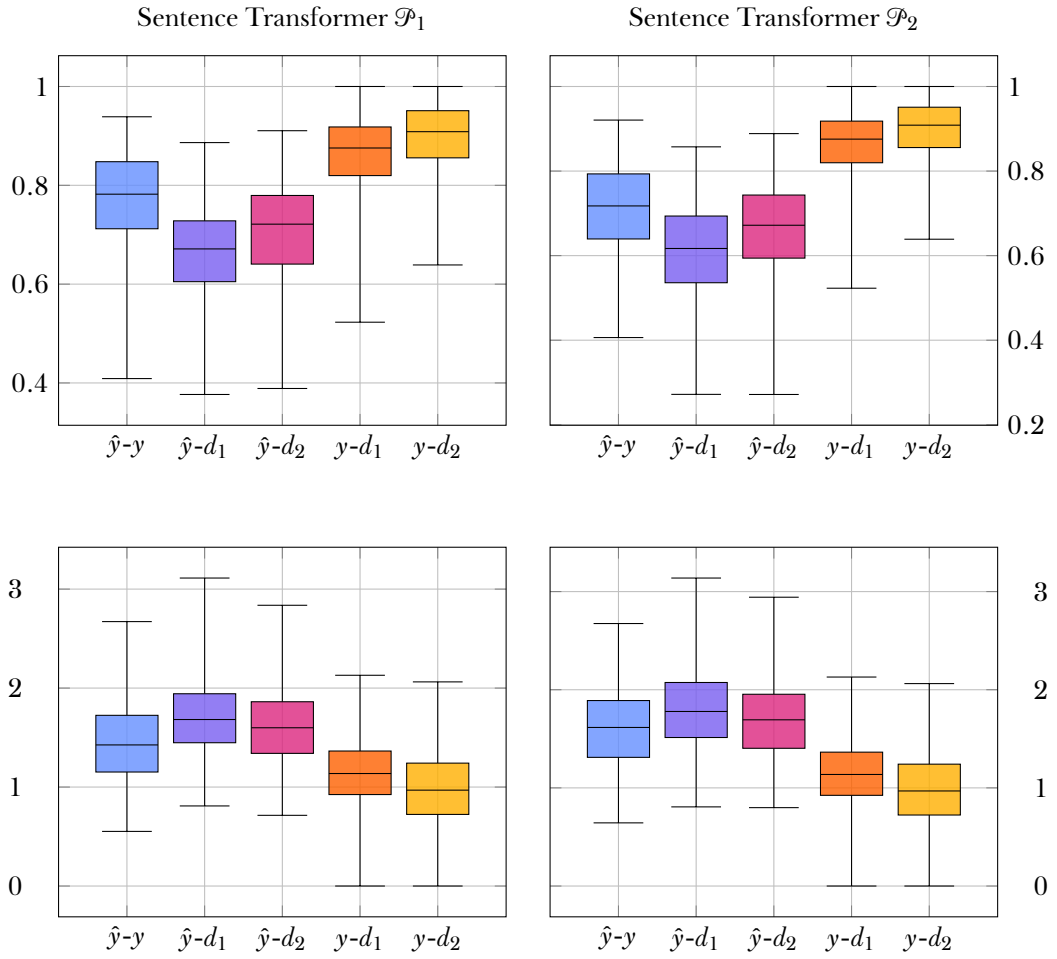

Figure 20: On the top, box plots of the cosine similarity scores obtained for the first probe $\mathcal{P}_1$ (left) and the second probe $\mathcal{P}_2$ (right). On the bottom, Euclidean distance scores observed for $\mathcal{P}_1$ (left) and $\mathcal{P}_2$ (right).

Chapter 6

Discussion

The ability of Pretrained Language Models (PLMs) to encode morphosyntactic information is hypothesized by several studies. This thesis provides experimental evidence that partially supports this hypothesis. Overall, the results outlined in chapter 5 exhibit a consistent trend across all languages and models, with the transformation $\hat{y}$ scoring a higher average similarity with the target y compared to the distractors d_1 and d_2. This pattern is particularly pronounced in Italian and German, while English displays a few inconsistencies (see subsection 5.1.1 and subsection 5.1.5). Nonetheless, despite the encouraging results, there are only a few instances where the transformation scores the closest similarity with the target. The most prominent cases are observed for BERT on the Italian and German agreement datasets, as described in subsection 5.2.4 and subsection 5.3.4, respectively.

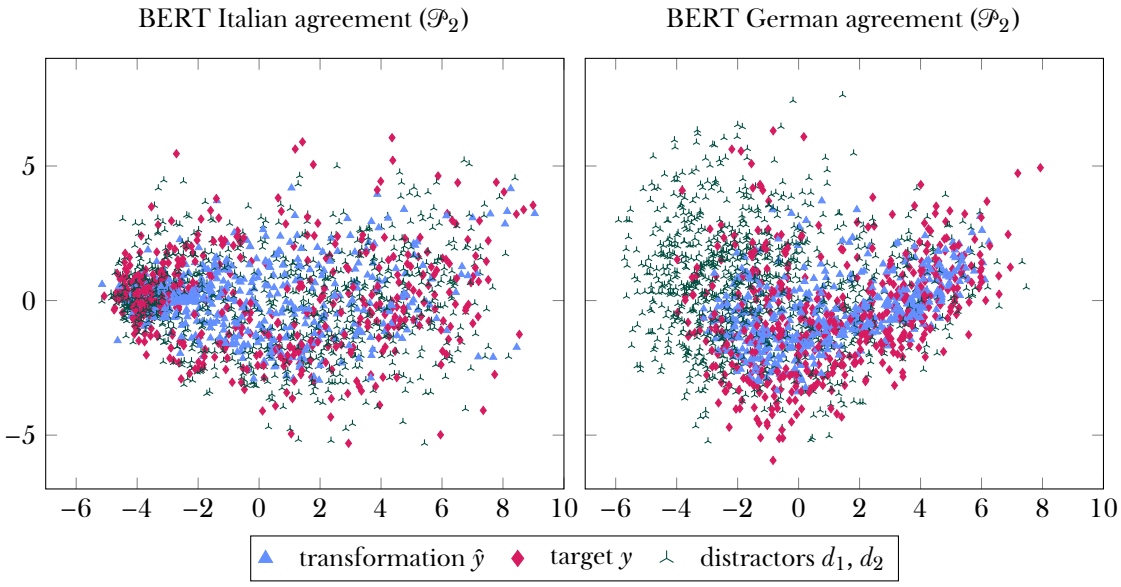

Figure 21: Plots of the Italian and German agreement embeddings obtained using BERT and the second probe $\mathcal{P}_2$. Distractors d_1 and d_2 are denoted by the same marker for clarity.

To get a clearer picture, Figure 21 portrays the embeddings extracted from the two best-performing BERT experiments carried out on the Italian and German

agreement datasets. The transformation $\hat{y}$ is generated by the second probe, $\mathcal{P}_2$. Each embedding is first reduced to its two primary components using Principal Component Analysis (PCA). If the PLM successfully captured morphosyntactic information, one would ideally expect to observe a clear pattern between $\hat{y}$ and y. This is not the case. The Italian embeddings show a relatively homogeneous distribution, in which all four types are mixed together without any clear structure. On the other hand, the German scatter plot shows a barely discernible pattern, with the $\hat{y}$ and y representations appearing in the bottom-right part of the cluster and the distractors occupying the top-left part.

While the scatter plots in Figure 21 are comparable to those observed for most of the remaining experiments, this does not provide conclusive evidence against PLMs being sensitive to morphosyntactic information. A plausible explanation for the plots may be that PCA neglected the embedding components encoding the morphosyntactic features under consideration, favoring and retaining more prominent ones. This seems reasonable, especially considering the tight similarity scores registered when comparing the four representations in most experiments.

To gain a better understanding of the results, and to attempt to disentangle the influence of the specific features of each PLM, it is worth considering how the information captured by these models is affected by their respective architectural choices and pretraining strategies. In this respect, RoBERTa stands out as the architecture displaying the most peculiar behavior. Overall, the average similarity scores between the target and the distractors tend to exhibit high values and a consistently low variation. Furthermore, from the results of the experiments, two trends can be identified. On the one hand, the three comparisons $\hat{y}$-y, $\hat{y}$-d_1 and $\hat{y}$-d_2 obtain markedly lower similarity scores than y-d_1 and y-d_2. For example, this is the case of the German tense experiment carried out with $\mathcal{P}_1$ (subsection 5.3.2). On the other hand, all five comparisons show very high similarity scores, making it hard to tell them apart. This is the case, for instance, of the Italian tense dataset results obtained using $\mathcal{P}_2$ (subsection 5.2.2).

These two trends are reflected in the embedding plots of the respective experiments. An example is provided in Figure 22, where the two scatter plots portray the embeddings for the previously mentioned German and Italian experiments. In the German case, the transformations sit far apart from the other embeddings. This is consistent with the lower similarity observed between $\hat{y}$ and the remaining representations compared to y-d_1 and y-d_2. In the case of Italian, all of the embeddings are closely clustered together, mirroring the tight average cosine similarity scores registered for each comparison. Notably, the $\hat{y}$ representations are not scattered but instead sit at the center of the cluster. This may justify the better average Euclidean distances of y-d_1 and y-d_2 compared to $\hat{y}$-y, $\hat{y}$-d_1 and $\hat{y}$-d_2.

Turning to the multilingual Sentence Transformer model, results are consistent across all languages and probes. Even though y-d_1 and y-d_2 remain the highest scoring pairs, $\hat{y}$-y exhibits a higher similarity than $\hat{y}$-d_1 and $\hat{y}$-d_2 in every experiment. Upon inspection, the resulting embeddings form a homogeneous cluster with no sign of a clear pattern between $\hat{y}$ and y. An example is the left scatter plot in Figure 23, which portrays the embeddings for the English agreement dataset with $\hat{y}$ obtained using $\mathcal{P}_1$ (subsection 5.1.6). On the other hand, an interesting exception is

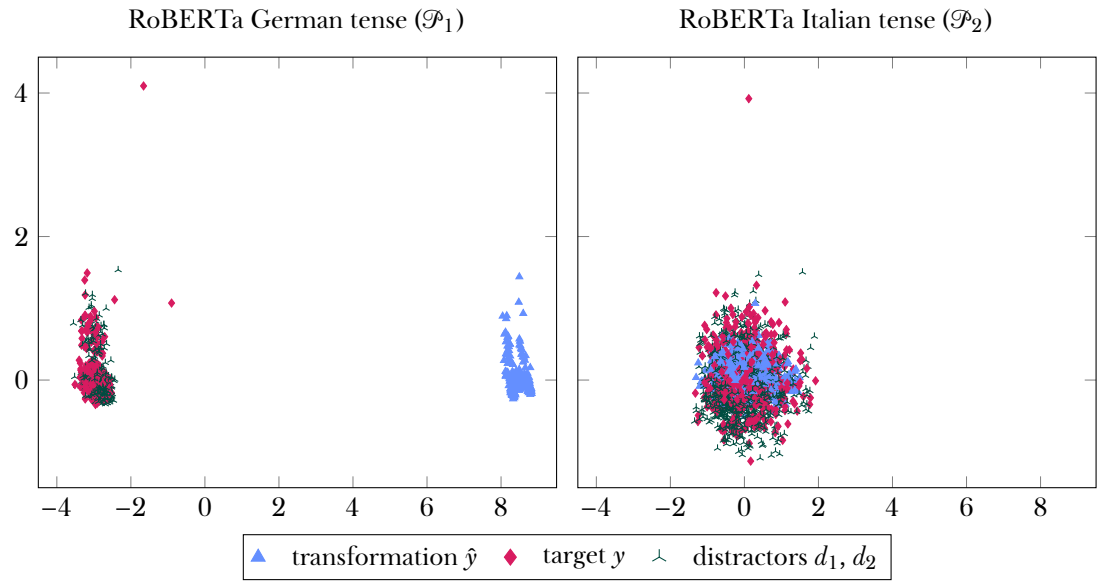

Figure 22: Plots of the German and Italian tense embeddings obtained using RoBERTa together with the first ($\mathcal{P}_1$) and second ($\mathcal{P}_2$) probe, respectively. Distractors d_1 and d_2 are denoted by the same marker for clarity.

represented by the second scatter plot, where the $\hat{y}$ transformations generated by the non-linear probe form three tight groups that sit at the center of the cluster. This pattern is in line with the lower similarity scores exhibited by $\hat{y}$-y, $\hat{y}$-d_1 and $\hat{y}$-d_2, compared to y-d_1 and y-d_2.

It is difficult to draw definitive conclusions about the extent to which architectural choices and pre-training strategies have an impact on the linguistic information encoded in PLMs. Nonetheless, while the results observed for BERT and the Sentence Transformer show a clear and somewhat promising trend, this is not the case for RoBERTa. Although $\hat{y}$ is generally closer to the target than the distractors in most experiments, the advantage is hardly significant. Additionally, the two comparisons y-d_1 and y-d_2 tend to yield considerably higher similarity scores, suggesting that both probes struggle to learn a successful mapping to get the representation of the first sentence in each quadruple as close as possible to that of the target one. This is evident from the left plot in Figure 22. Experiments yielding consistently high similarity scores across all comparisons are equally inconclusive, indicating no significant differences between the four representations.

Across the three architectures investigated in this thesis, RoBERTa is the only one relying solely on the (dynamic) Masked Language Modeling (MLM) pretraining objective. BERT uses both MLM and Next Sentence Prediction, while the Sentence Transformer model relies on Masked and Permuted Language Modeling. This may indicate a possible effect of the pretraining strategy on a PLM's ability to encode linguistic information. Although this effect is not directly proven, a second indication in its favor may come from a comparison of the agreement results registered for the English and Italian RoBERTa models, with respect to the second probe. This is interesting due to the slightly different pretraining strategies adopted by both models. As detailed in section 4.2, while the English one relies on MLM, the Italian RoBERTa-like UmBERTo uses Whole Word Masking (WWM). Looking at the

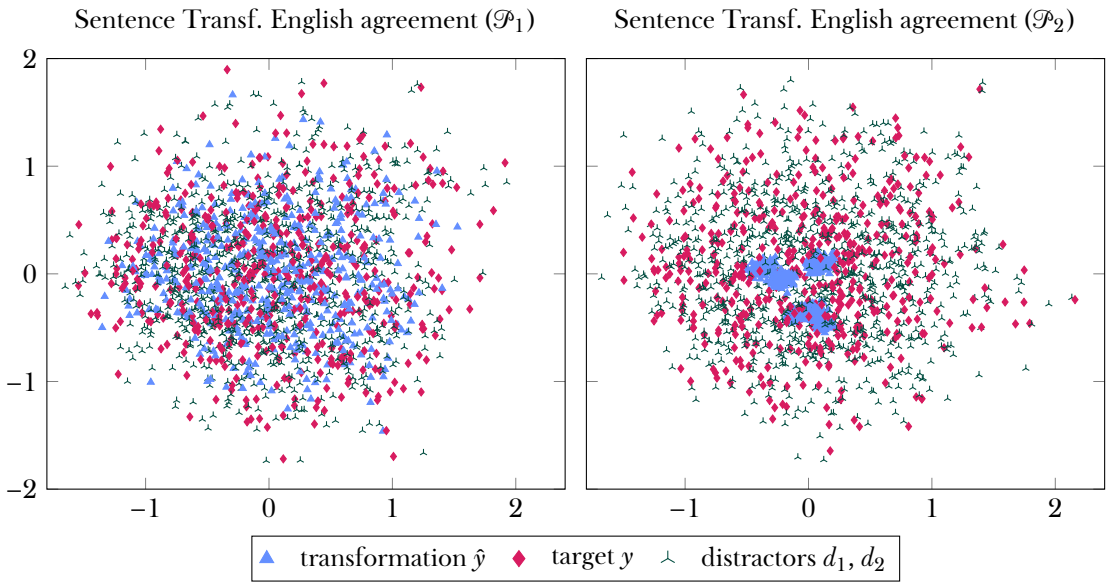

Figure 23: Plots of the English agreement embeddings obtained using the multilingual Sentence Transformer model together with both probes, $\mathcal{P}_1$ and $\mathcal{P}_2$. Distractors d_1 and d_2 are denoted by the same marker for clarity.

results, across both languages the five comparisons register significantly high and tight scores (see subsection 5.1.5, subsection 5.2.5). As illustrated in Figure 24, this results in the four representations being clustered together. However, there is a clear difference between the two scatter plots when it comes to $\hat{y}$. In the case of English, the transformations are arranged according to a peculiar pattern, while this is not the case for Italian. Although not portrayed in the picture, the same pattern is observed for the German RoBERTa results as well. This suggests that MLM and WWM could potentially have different effects on the information stored in the embeddings.

These clues provide an opportunity for further improvement and exploration. However, alongside the analysis of architectures and pretraining methods, it is also sensible to consider the results from the perspective of the two neural probes. In this sense, neither the affine transformation carried out by $\mathcal{P}_1$ nor the non-linear one carried out by $\mathcal{P}_2$ stands out as the best solution. This could indicate that the underlying embedding space for each PLM is linear. If not, the second probe would consistently obtain better results and outperform the first one as a universal approximator. Nevertheless, the limited size of the datasets used to train the probes should also be taken into account as one of the potential reasons for their sub-optimal performance. Therefore, exploring new probe architectures with different layer sizes and activation functions seems to be an interesting avenue for further research.

The final research question of this thesis considers whether PLMs encode certain morphosyntactic relations better others. Again, the results do not yield a clear and definite answer. However, across all languages, the best scores are observed mainly on the agreement datasets. One plausible explanation for this trend may lie in the higher variation across the sentences in each sample. While for the tense dataset sentences in each quadruple only vary with respect to the verb, in the case of agreement there are at least two elements that need to change, namely the subject and the verb. It should be noted, however, that the English agreement results are likely to be

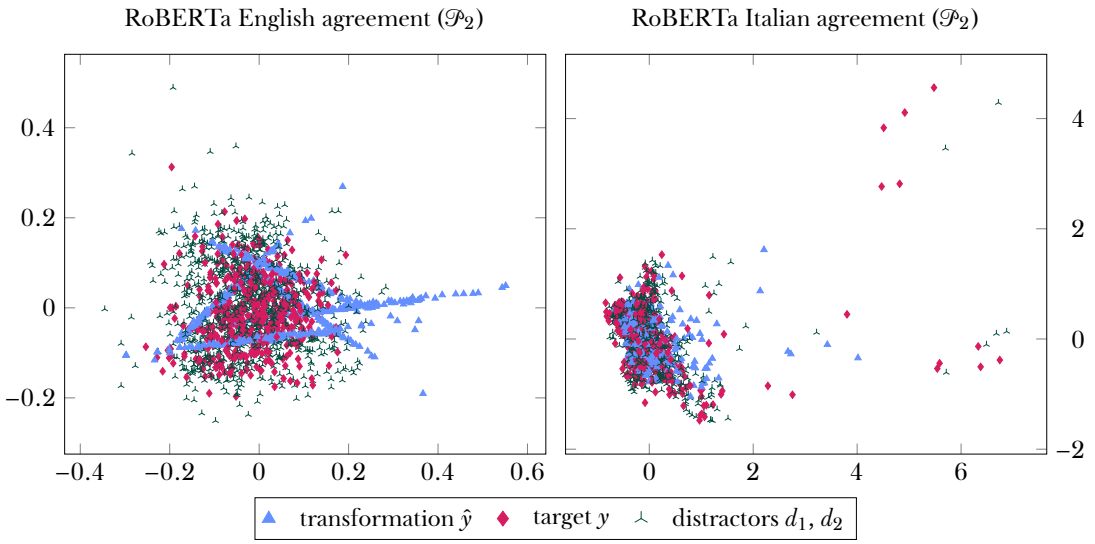

Figure 24: Plots of the English and Italian agreement embeddings obtained using RoBERTa and the second probe $\mathcal{P}_2$. Distractors d_1 and d_2 are denoted by the same marker for clarity.

somewhat biased due to the syncretism between second person singular and plural. In other words, should the first sentence in a quadruple be in the second person singular, it will match the target plural sentence exactly e.g. *You go to school* → *You go to school*. This highlights a methodological shortcoming that leaves room for further improvement in the investigation of this research question. First, it may be beneficial to refine and expand the existing datasets to introduce more variation. At the same time, additional morphosyntactic features should be considered that leave as little room as possible for syncretic cases. Lastly, it may be interesting to expand the analysis to non-Indo-European languages to increase the range of morphosyntactic phenomena that PLMs are tested for.

In summary, the approach outlined in this thesis is promising and warrants additional exploration. Notably, it diverges from earlier approaches such as diagnostic and behavioral probing. The former relies on a classifier to predict a property from a given representation. However, high accuracy cannot be safely interpreted as an indication that the representation encodes any linguistic structure, as the probe may have just learned the task. Behavioral probing, on the other hand, evaluates a model's predictions relying on carefully crafted evaluation datasets. This may provide useful insights about the model as a whole, but fails to explain its inner structure. The solution adopted in this thesis attempts to overcome these limitations, while retaining the strong points of the previous approaches. Contrary to diagnostic probing, the proposed method relies on a direct manipulation of contextual representations. Embeddings are mapped to the target linguistic features through neural probes and each mapping is evaluated both against the target and two distractors. This gives a better perspective on the results. Each mapping experiment allows to draw conclusions about the PLM as a whole, however inspecting the trained probes may also offer interesting insights about the internal structure of the representations and how they store information. Specifically, a particular linguistic feature may be stored either locally or in a distributed manner. Observing and intervening on the individual

neurons of a probe may help locate elements of a representation that encode the feature under investigation.

Chapter 7

Conclusion

The emergence of Pretrained Language Models (PLMs) has marked a major paradigm shift in the field of Natural Language Processing (NLP). By leveraging large amounts of text data, these models can learn contextual representations that enhance the performance of many downstream tasks. However, these models are also very hard to understand, raising issues about their lack of interpretability. This thesis sought to shed some light on their inner workings by investigating their linguistic knowledge. Specifically, this analysis focused on their ability to encode morphosyntactic information for English, German, and Italian, with respect to two phenomena: tense and subject-verb agreement. In this respect, this thesis addressed three research questions: Do PLMs encode morphosyntactic information? How does a PLM's implementation and pretraining strategy affect its ability to encode such information? Is one morphosyntactic phenomenon captured better than the other?

In answering these questions, the thesis makes two key contributions. It presents a new probing method that uses neural probes to manipulate PLMs representations and map them to target linguistic features, as described in chapter 4. It offers curated tense and agreement datasets for the three languages under study, as outlined in chapter 3.

Results presented in chapter 5 indicate that PLMs are likely able to encode information about tense and agreement. However, as discussed in chapter 6, there are only a few cases where a transformation generated by one of the probes scores the closest similarity with the corresponding target representation. These results are promising, but not entirely conclusive. With respect to the second question, BERT and Sentence Transformer exhibit overall comparable behaviors, while RoBERTa stands out as the architecture displaying the most peculiar one. This suggests that a PLM implementation and pretraining strategy may actually affect its ability to encode information. When considering the last question, results appear not to yield a clear and definite answer. Nonetheless, across all three languages, there is a tendency to observe better scores on the agreement datasets. Overall, these findings motivate the need for further research on the interpretability of PLMs, as well as on the refinement and validation of the proposed probing method.

Appendix A

Appendix

A.1 Hyperparameters

A.1.1 English

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 091	0.007 272	0.593 772	0.539 807	8
11	0.000 054	0.001 741	0.581 206	0.573 784	8
10	0.000 095	0.007 962	0.720 886	0.688 626	32
9	0.000 035	0.004 187	0.691 137	0.649 323	8
8	0.000 059	0.003 527	0.637 658	0.676 215	32
7	0.000 039	0.007 920	0.690 397	0.883 847	8
6	0.000 087	0.001 097	0.522 372	0.803 553	32
5	0.000 072	0.002 401	0.514 407	0.569 118	8
4	0.000 086	0.006 253	0.620 025	0.848 849	8
3	0.000 059	0.004 728	0.562 881	0.853 474	32
2	0.000 091	0.003 729	0.743 073	0.607 158	16
1	0.000 042	0.001 577	0.681 739	0.656 351	8

Table A.1: Tuned hyperparameters for each layer of BERT with respect to the first probe $\mathcal{P}_1$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 096	0.005 026	0.793 187	0.659 003	32
11	0.000 029	0.006 759	0.773 053	0.844 755	8
10	0.000 010	0.007 569	0.647 045	0.857 273	8
9	0.000 077	0.001 522	0.562 377	0.618 645	8
8	0.000 060	0.007 515	0.501 071	0.773 349	16
7	0.000 058	0.002 985	0.611 530	0.623 646	8
6	0.000 051	0.002 023	0.582 299	0.506 117	8
5	0.000 034	0.007 402	0.511 565	0.613 264	8
4	0.000 069	0.005 074	0.787 189	0.740 424	8
3	0.000 046	0.008 366	0.666 536	0.836 369	8
2	0.000 077	0.001 300	0.803 675	0.592 088	16
1	0.000 054	0.002 485	0.556 941	0.764 444	32

Table A.2: Tuned hyperparameters for each layer of BERT with respect to the second probe $\mathcal{P}_2$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 037	0.001 582	0.671 110	0.839 359	8
11	0.000 071	0.001 481	0.543 230	0.575 635	16
10	0.000 087	0.004 897	0.553 696	0.594 286	8
9	0.000 048	0.002 172	0.588 084	0.664 285	8
8	0.000 050	0.001 209	0.760 636	0.558 625	8
7	0.000 064	0.003 694	0.750 118	0.724 225	32
6	0.000 021	0.007 267	0.855 483	0.565 690	8
5	0.000 028	0.002 801	0.592 712	0.504 855	8
4	0.000 077	0.004 946	0.747 312	0.806 673	16
3	0.000 001	0.001 507	0.737 195	0.665 983	16
2	0.000 006	0.007 628	0.546 813	0.754 221	8
1	0.000 058	0.001 133	0.624 462	0.811 164	16

Table A.3: Tuned hyperparameters for each layer of RoBERTa with respect to the first probe $\mathcal{P}_1$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 027	0.009 192	0.502 493	0.524 140	16
11	0.000 074	0.003 990	0.640 847	0.769 638	8
10	0.000 042	0.003 869	0.765 543	0.670 158	8
9	0.000 047	0.001 225	0.532 837	0.767 204	8
8	0.000 086	0.009 339	0.751 937	0.808 643	16
7	0.000 068	0.002 907	0.562 451	0.816 296	8
6	0.000 061	0.001 823	0.682 521	0.645 343	16
5	0.000 036	0.001 031	0.881 273	0.795 859	8
4	0.000 009	0.001 277	0.655 876	0.551 272	8
3	0.000 003	0.003 241	0.530 657	0.874 262	8
2	0.000 086	0.006 666	0.762 279	0.596 944	8
1	0.000 006	0.001 688	0.602 194	0.761 771	8

Table A.4: Tuned hyperparameters for each layer of RoBERTa with respect to the second probe $\mathcal{P}_2$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
1	0.000 075	0.003 559	0.612 730	0.854 786	16

Table A.5: Tuned hyperparameters for the Sentence Transformer model with respect to the first probe $\mathscr{P}_1$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
1	0.000 044	0.002 114	0.728 031	0.707 573	16

Table A.6: Tuned hyperparameters for the Sentence Transformer model with respect to the second probe $\mathscr{P}_2$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 016	0.003 606	0.671 726	0.731 098	8
11	0.000 048	0.002 257	0.533 572	0.811 937	8
10	0.000 026	0.001 862	0.697 950	0.677 921	8
9	0.000 019	0.005 434	0.508 591	0.660 277	16
8	0.000 079	0.007 784	0.865 795	0.633 297	16
7	0.000 062	0.002 582	0.598 838	0.752 554	16
6	0.000 041	0.002 607	0.826 524	0.719 545	8
5	0.000 076	0.002 135	0.689 817	0.609 937	16
4	0.000 088	0.007 681	0.862 593	0.669 919	16
3	0.000 088	0.008 915	0.583 087	0.725 204	32
2	0.000 017	0.001 792	0.544 775	0.517 098	8
1	0.000 074	0.007 779	0.682 155	0.616 283	8

Table A.7: Tuned hyperparameters for each layer of BERT with respect to the first probe $\mathscr{P}_1$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 072	0.009 836	0.862 618	0.763 955	8
11	0.000 074	0.001 509	0.557 215	0.596 256	16
10	0.000 051	0.001 981	0.517 841	0.702 064	8
9	0.000 050	0.002 998	0.577 547	0.863 787	16
8	0.000 057	0.008 977	0.528 030	0.696 412	16
7	0.000 095	0.007 053	0.776 368	0.714 904	8
6	0.000 007	0.003 503	0.836 881	0.888 000	8
5	0.000 057	0.001 973	0.712 908	0.618 038	32
4	0.000 053	0.002 193	0.643 845	0.752 366	8
3	0.000 070	0.009 515	0.523 487	0.597 348	32
2	0.000 049	0.003 928	0.548 980	0.686 065	8
1	0.000 026	0.008 351	0.715 783	0.600 012	8

Table A.8: Tuned hyperparameters for each layer of BERT with respect to the second probe $\mathscr{P}_2$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 067	0.001 383	0.895 293	0.634 367	32
11	0.000 031	0.004 401	0.827 082	0.782 740	8
10	0.000 045	0.002 994	0.741 171	0.881 704	8
9	0.000 002	0.008 302	0.691 667	0.621 970	8
8	0.000 092	0.008 004	0.716 318	0.763 750	8
7	0.000 020	0.001 113	0.512 831	0.696 819	8
6	0.000 071	0.003 768	0.598 635	0.559 288	16
5	0.000 022	0.001 024	0.530 725	0.815 115	8
4	0.000 042	0.008 535	0.512 053	0.842 592	8
3	0.000 046	0.001 176	0.541 980	0.748 537	8
2	0.000 008	0.001 418	0.876 567	0.855 500	8
1	0.000 020	0.001 471	0.727 401	0.859 711	8

Table A.9: Tuned hyperparameters for each layer of RoBERTa with respect to the first probe $\mathcal{P}_1$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 032	0.005 918	0.543 915	0.798 766	8
11	0.000 047	0.002 491	0.527 400	0.665 976	8
10	0.000 091	0.001 142	0.800 159	0.636 677	8
9	0.000 086	0.004 862	0.789 295	0.871 002	16
8	0.000 081	0.004 546	0.708 159	0.730 404	8
7	0.000 040	0.002 114	0.590 384	0.554 607	8
6	0.000 024	0.003 338	0.598 385	0.753 695	8
5	0.000 022	0.003 767	0.591 135	0.507 690	8
4	0.000 032	0.001 044	0.826 856	0.791 745	8
3	0.000 093	0.001 917	0.556 788	0.710 564	32
2	0.000 023	0.005 377	0.873 445	0.854 931	8
1	0.000 045	0.008 477	0.596 451	0.661 246	8

Table A.10: Tuned hyperparameters for each layer of RoBERTa with respect to the second probe $\mathcal{P}_2$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
1	0.000 089	0.001 686	0.540 129	0.815 443	16

Table A.11: Tuned hyperparameters for the Sentence Transformer model with respect to the first probe $\mathcal{P}_1$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
1	0.000 003	0.003 231	0.868 145	0.580 845	8

Table A.12: Tuned hyperparameters for the Sentence Transformer model with respect to the second probe $\mathcal{P}_2$ and the agreement dataset.

A.1.2 Italian

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 064	0.001 469	0.680 872	0.855 507	16
11	0.000 074	0.002 097	0.545 516	0.640 821	32
10	0.000 034	0.001 310	0.682 789	0.692 164	8
9	0.000 062	0.001 765	0.833 542	0.627 333	8
8	0.000 037	0.001 074	0.741 945	0.732 564	32
7	0.000 096	0.005 344	0.722 254	0.778 103	8
6	0.000 009	0.002 247	0.516 985	0.556 840	16
5	0.000 093	0.002 100	0.526 647	0.652 925	16
4	0.000 033	0.001 671	0.593 908	0.715 951	16
3	0.000 084	0.001 181	0.610 071	0.624 991	8
2	0.000 053	0.004 577	0.569 993	0.563 257	8
1	0.000 029	0.001 990	0.653 579	0.798 035	8

Table A.13: Tuned hyperparameters for each layer of BERT with respect to the first probe $\mathcal{P}_1$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 065	0.001 305	0.588 098	0.799 757	16
11	0.000 017	0.006 203	0.554 829	0.810 560	8
10	0.000 048	0.007 595	0.802 825	0.665 262	16
9	0.000 097	0.003 008	0.722 858	0.791 859	8
8	0.000 076	0.009 325	0.759 479	0.787 910	32
7	0.000 018	0.001 193	0.738 144	0.827 640	8
6	0.000 076	0.002 471	0.636 475	0.810 186	8
5	0.000 060	0.002 313	0.591 711	0.659 242	8
4	0.000 074	0.006 217	0.524 776	0.514 729	8
3	0.000 084	0.007 366	0.651 427	0.501 991	8
2	0.000 026	0.004 781	0.521 634	0.733 059	8
1	0.000 026	0.005 195	0.736 976	0.880 972	8

Table A.14: Tuned hyperparameters for each layer of BERT with respect to the second probe $\mathcal{P}_2$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 016	0.007 550	0.799 036	0.614 123	32
11	0.000 032	0.003 504	0.552 475	0.790 328	16
10	0.000 033	0.001 451	0.509 336	0.846 167	8
9	0.000 018	0.001 300	0.888 236	0.814 849	16
8	0.000 052	0.005 598	0.602 819	0.579 160	16
7	0.000 082	0.005 280	0.765 357	0.583 336	16
6	0.000 047	0.001 449	0.892 793	0.623 053	16
5	0.000 038	0.003 902	0.714 221	0.527 437	8
4	0.000 020	0.006 377	0.862 457	0.802 566	8
3	0.000 096	0.001 120	0.780 497	0.651 253	8
2	0.000 083	0.002 974	0.586 644	0.877 904	16
1	0.000 091	0.009 060	0.865 868	0.797 364	16

Table A.15: Tuned hyperparameters for each layer of RoBERTa with respect to the first probe $\mathcal{P}_1$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 095	0.001 803	0.809 945	0.677 281	16
11	0.000 063	0.002 346	0.626 869	0.653 671	8
10	0.000 035	0.005 425	0.673 847	0.679 545	16
9	0.000 072	0.001 366	0.782 307	0.814 265	16
8	0.000 025	0.007 652	0.870 996	0.779 529	8
7	0.000 052	0.002 805	0.898 044	0.630 342	16
6	0.000 010	0.003 025	0.777 058	0.855 220	8
5	0.000 035	0.004 491	0.759 689	0.590 146	8
4	0.000 054	0.005 683	0.744 988	0.872 405	8
3	0.000 083	0.004 741	0.680 059	0.545 901	8
2	0.000 046	0.001 070	0.745 692	0.788 105	8
1	0.000 006	0.003 520	0.619 071	0.592 543	8

Table A.16: Tuned hyperparameters for each layer of RoBERTa with respect to the second probe $\mathcal{P}_2$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
1	0.000 069	0.004 736	0.844 726	0.573 875	16

Table A.17: Tuned hyperparameters for the Sentence Transformer model with respect to the first probe $\mathcal{P}_1$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
1	0.000 084	0.002 124	0.698 680	0.621 749	16

Table A.18: Tuned hyperparameters for the Sentence Transformer model with respect to the second probe $\mathcal{P}_2$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 047	0.002 967	0.765 198	0.655 379	16
11	0.000 009	0.002 389	0.579 901	0.520 098	32
10	0.000 046	0.004 943	0.654 766	0.848 976	8
9	0.000 046	0.001 640	0.674 746	0.735 477	16
8	0.000 017	0.002 901	0.660 134	0.801 538	32
7	0.000 081	0.001 393	0.733 662	0.855 830	16
6	0.000 070	0.006 943	0.550 402	0.553 161	8
5	0.000 077	0.003 123	0.603 939	0.615 867	8
4	0.000 059	0.004 105	0.817 505	0.849 698	8
3	0.000 010	0.003 268	0.503 255	0.556 001	8
2	0.000 026	0.005 931	0.631 983	0.831 605	8
1	0.000 090	0.006 880	0.634 684	0.830 062	8

Table A.19: Tuned hyperparameters for each layer of BERT with respect to the first probe $\mathscr{P}_1$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 085	0.006 564	0.751 058	0.766 678	16
11	0.000 083	0.005 113	0.895 362	0.820 897	16
10	0.000 053	0.004 801	0.577 332	0.861 253	8
9	0.000 044	0.001 811	0.548 754	0.872 980	16
8	0.000 050	0.009 374	0.601 921	0.837 130	16
7	0.000 036	0.005 492	0.617 569	0.593 106	16
6	0.000 068	0.003 688	0.838 883	0.775 062	8
5	0.000 072	0.006 496	0.500 870	0.662 669	32
4	0.000 050	0.002 689	0.558 241	0.560 962	8
3	0.000 082	0.007 655	0.859 572	0.561 772	32
2	0.000 090	0.003 482	0.718 337	0.736 381	8
1	0.000 012	0.002 197	0.585 792	0.544 377	8

Table A.20: Tuned hyperparameters for each layer of BERT with respect to the second probe $\mathscr{P}_2$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 087	0.003 721	0.762 604	0.809 310	16
11	0.000 097	0.002 604	0.585 917	0.547 006	8
10	0.000 033	0.008 409	0.652 765	0.857 479	8
9	0.000 090	0.001 922	0.849 731	0.771 485	16
8	0.000 094	0.002 330	0.661 220	0.736 878	16
7	0.000 053	0.001 830	0.679 277	0.566 358	16
6	0.000 067	0.001 796	0.568 407	0.723 511	8
5	0.000 046	0.001 095	0.743 457	0.671 335	32
4	0.000 097	0.002 057	0.584 006	0.833 849	16
3	0.000 044	0.002 168	0.615 704	0.897 357	8
2	0.000 069	0.003 392	0.834 403	0.532 399	8
1	0.000 045	0.001 250	0.602 363	0.771 287	32

Table A.21: Tuned hyperparameters for each layer of RoBERTa with respect to the first probe $\mathscr{P}_1$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 065	0.004 650	0.522 753	0.698 578	8
11	0.000 048	0.007 437	0.625 740	0.859 326	16
10	0.000 086	0.002 360	0.550 207	0.636 608	8
9	0.000 098	0.007 841	0.812 058	0.520 547	8
8	0.000 043	0.002 476	0.743 573	0.512 191	8
7	0.000 099	0.001 898	0.549 848	0.512 903	16
6	0.000 050	0.009 069	0.885 567	0.508 187	8
5	0.000 029	0.001 865	0.678 608	0.854 403	16
4	0.000 020	0.006 100	0.599 861	0.747 018	8
3	0.000 056	0.005 849	0.510 465	0.637 915	8
2	0.000 013	0.008 259	0.535 840	0.513 804	8
1	0.000 063	0.002 888	0.542 108	0.705 858	16

Table A.22: Tuned hyperparameters for each layer of RoBERTa with respect to the second probe $\mathcal{P}_2$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
1	0.000 031	0.003 624	0.600 858	0.511 751	16

Table A.23: Tuned hyperparameters for the Sentence Transformer model with respect to the first probe $\mathcal{P}_1$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
1	0.000 066	0.002 314	0.574 215	0.650 380	8

Table A.24: Tuned hyperparameters for the Sentence Transformer model with respect to the second probe $\mathcal{P}_2$ and the agreement dataset.

A.1.3 German

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 072	0.003 946	0.877 847	0.638 426	8
11	0.000 047	0.001 771	0.567 346	0.547 993	16
10	0.000 042	0.005 299	0.774 485	0.573 207	16
9	0.000 039	0.007 813	0.591 045	0.710 768	32
8	0.000 062	0.003 681	0.864 931	0.758 408	16
7	0.000 035	0.004 411	0.806 657	0.748 197	8
6	0.000 056	0.003 359	0.751 225	0.500 252	8
5	0.000 066	0.003 927	0.785 101	0.522 140	8
4	0.000 080	0.008 307	0.605 821	0.564 611	16
3	0.000 034	0.005 911	0.538 863	0.534 653	8
2	0.000 069	0.008 208	0.717 745	0.718 350	8
1	0.000 050	0.001 473	0.503 801	0.766 124	16

Table A.25: Tuned hyperparameters for each layer of BERT with respect to the first probe $\mathcal{P}_1$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 029	0.008 926	0.745 931	0.518 142	8
11	0.000 018	0.005 585	0.622 300	0.640 805	32
10	0.000 031	0.002 132	0.891 276	0.832 990	8
9	0.000 084	0.002 414	0.528 297	0.807 704	8
8	0.000 012	0.002 821	0.574 384	0.501 007	16
7	0.000 084	0.002 117	0.753 831	0.838 238	16
6	0.000 051	0.002 364	0.764 824	0.507 795	8
5	0.000 064	0.004 865	0.802 200	0.751 462	16
4	0.000 048	0.004 691	0.521 380	0.610 080	8
3	0.000 068	0.006 126	0.607 608	0.523 672	16
2	0.000 066	0.004 154	0.865 752	0.876 046	16
1	0.000 087	0.002 364	0.549 279	0.877 168	8

Table A.26: Tuned hyperparameters for each layer of BERT with respect to the second probe $\mathcal{P}_2$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 036	0.001 511	0.581 724	0.836 765	8
11	0.000 049	0.001 455	0.766 105	0.541 696	8
10	0.000 055	0.007 995	0.863 488	0.865 589	8
9	0.000 034	0.003 451	0.656 826	0.696 539	8
8	0.000 061	0.002 122	0.833 865	0.676 346	8
7	0.000 087	0.001 219	0.585 650	0.558 503	8
6	0.000 083	0.009 636	0.552 683	0.682 854	16
5	0.000 044	0.001 516	0.512 057	0.575 836	8
4	0.000 038	0.006 586	0.551 535	0.708 060	8
3	0.000 016	0.002 199	0.588 317	0.592 075	8
2	0.000 030	0.001 273	0.636 626	0.829 532	8
1	0.000 008	0.001 855	0.597 006	0.552 880	16

Table A.27: Tuned hyperparameters for each layer of RoBERTa with respect to the first probe $\mathcal{P}_1$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 032	0.005 312	0.580 263	0.569 622	32
11	0.000 047	0.001 989	0.671 279	0.763 523	16
10	0.000 037	0.002 164	0.677 528	0.525 225	8
9	0.000 079	0.008 101	0.649 116	0.777 919	8
8	0.000 019	0.005 274	0.796 287	0.740 297	16
7	0.000 031	0.002 054	0.612 419	0.754 690	16
6	0.000 092	0.005 224	0.802 483	0.730 975	8
5	0.000 034	0.006 437	0.577 532	0.649 292	32
4	0.000 021	0.003 456	0.719 871	0.632 123	16
3	0.000 097	0.003 561	0.770 100	0.761 627	16
2	0.000 073	0.001 271	0.501 939	0.516 906	16
1	0.000 088	0.001 299	0.576 095	0.517 527	32

Table A.28: Tuned hyperparameters for each layer of RoBERTa with respect to the second probe $\mathcal{P}_2$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
1	0.000 085	0.009 566	0.870 337	0.787 498	16

Table A.29: Tuned hyperparameters for the Sentence Transformer model with respect to the first probe $\mathcal{P}_1$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
1	0.000 058	0.001 604	0.820 541	0.546 169	8

Table A.30: Tuned hyperparameters for the Sentence Transformer model with respect to the second probe $\mathcal{P}_2$ and the tense dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 076	0.004 917	0.547 990	0.723 906	8
11	0.000 062	0.001 068	0.605 207	0.660 190	8
10	0.000 033	0.002 783	0.578 513	0.735 764	16
9	0.000 094	0.002 110	0.678 640	0.631 065	32
8	0.000 067	0.005 779	0.686 200	0.534 121	8
7	0.000 017	0.002 501	0.653 068	0.675 000	8
6	0.000 086	0.001 866	0.754 398	0.678 858	8
5	0.000 018	0.007 851	0.680 995	0.827 740	16
4	0.000 098	0.001 102	0.753 270	0.526 382	8
3	0.000 050	0.002 323	0.619 651	0.502 697	8
2	0.000 054	0.001 542	0.762 563	0.794 125	8
1	0.000 100	0.006 501	0.865 614	0.648 605	8

Table A.31: Tuned hyperparameters for each layer of BERT with respect to the first probe $\mathscr{P}_1$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 032	0.002 390	0.655 325	0.551 394	8
11	0.000 052	0.005 940	0.693 325	0.741 197	8
10	0.000 057	0.001 071	0.594 780	0.796 633	8
9	0.000 072	0.006 160	0.547 306	0.821 819	8
8	0.000 050	0.002 018	0.544 957	0.517 989	32
7	0.000 016	0.001 228	0.734 978	0.506 161	16
6	0.000 023	0.001 550	0.716 297	0.823 159	8
5	0.000 034	0.001 545	0.699 312	0.826 222	8
4	0.000 037	0.005 355	0.598 291	0.622 065	8
3	0.000 084	0.001 647	0.665 045	0.574 416	8
2	0.000 060	0.002 747	0.851 609	0.649 975	32
1	0.000 021	0.006 920	0.642 807	0.632 184	8

Table A.32: Tuned hyperparameters for each layer of BERT with respect to the second probe $\mathscr{P}_2$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 034	0.005 403	0.710 949	0.577 290	8
11	0.000 082	0.001 070	0.704 971	0.538 718	8
10	0.000 019	0.001 409	0.789 939	0.822 679	8
9	0.000 048	0.001 384	0.545 921	0.540 389	8
8	0.000 030	0.002 949	0.597 869	0.648 538	8
7	0.000 023	0.003 778	0.636 391	0.583 554	32
6	0.000 043	0.002 661	0.648 826	0.800 331	8
5	0.000 008	0.003 230	0.793 535	0.871 448	16
4	0.000 045	0.007 879	0.593 180	0.658 802	16
3	0.000 033	0.001 391	0.603 939	0.751 553	16
2	0.000 036	0.001 640	0.526 755	0.533 165	8
1	0.000 069	0.008 912	0.639 051	0.734 355	16

Table A.33: Tuned hyperparameters for each layer of RoBERTa with respect to the first probe $\mathscr{P}_1$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
12	0.000 063	0.005 453	0.724 183	0.660 330	16
11	0.000 009	0.004 650	0.665 994	0.519 234	8
10	0.000 049	0.002 989	0.634 040	0.802 098	8
9	0.000 063	0.001 437	0.672 350	0.741 287	16
8	0.000 086	0.009 716	0.877 933	0.578 990	32
7	0.000 072	0.001 967	0.537 806	0.703 903	32
6	0.000 063	0.001 430	0.543 921	0.718 842	16
5	0.000 045	0.002 922	0.504 168	0.745 900	8
4	0.000 044	0.003 065	0.598 335	0.508 119	8
3	0.000 053	0.006 568	0.549 693	0.838 477	8
2	0.000 029	0.005 986	0.841 583	0.591 205	8
1	0.000 016	0.002 002	0.753 219	0.867 836	16

Table A.34: Tuned hyperparameters for each layer of RoBERTa with respect to the second probe $\mathcal{P}_2$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
1	0.000 058	0.002 727	0.702 849	0.592 311	8

Table A.35: Tuned hyperparameters for the Sentence Transformer model with respect to the first probe $\mathcal{P}_1$ and the agreement dataset.

	learning rate	weight decay	beta 1	beta 2	batch size
1	0.000 034	0.003 621	0.736 549	0.614 886	16

Table A.36: Tuned hyperparameters for the Sentence Transformer model with respect to the second probe $\mathcal{P}_2$ and the agreement dataset.

A.2 Results

For each language, experiment and probe, this section reports the full results covering all of the twelve layers of BERT and RoBERTa. Specifically, for each of the two models, eight tables are provided:

- two tables (one per probe) covering the cosine similarity results for the tense dataset;
- two tables (one per probe) covering the Euclidean distance results for the tense dataset;
- two tables (one per probe) covering the cosine similarity results for the agreement dataset;
- two tables (one per probe) covering the Euclidean distance results for the agreement dataset.

A.2.1 English

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.9745±0.0082	0.9694±0.0071	0.9729±0.0066	0.9794±0.0087	0.9847±0.0079
11	0.8553±0.0294	0.8391±0.0334	0.8445±0.0316	0.9682±0.0171	0.9754±0.0144
10	0.7816±0.0306	0.7733±0.0315	0.7742±0.031	0.9719±0.0176	0.9767±0.0151
9	0.8942±0.0162	0.8863±0.017	0.887±0.0182	0.9805±0.0143	0.9843±0.013
8	0.8129±0.0189	0.8062±0.0187	0.806±0.0209	0.982±0.0144	0.9868±0.0117
7	0.9385±0.0111	0.9308±0.0124	0.9341±0.0125	0.9809±0.0149	0.9868±0.0106
6	0.847±0.0184	0.8394±0.0153	0.8449±0.0194	0.9846±0.0116	0.9895±0.0088
5	0.9817±0.0054	0.9738±0.0095	0.9781±0.0081	0.9818±0.0133	0.9885±0.0087
4	0.9892±0.0033	0.9867±0.0038	0.9876±0.005	0.9923±0.0053	0.9935±0.0051
3	0.7635±0.0231	0.753±0.0218	0.7596±0.023	0.9939±0.0041	0.9931±0.005
2	0.9047±0.0121	0.9023±0.0118	0.9032±0.0119	0.9991±0.0004	0.9994±0.0002
1	0.8615±0.0094	0.86±0.0093	0.8609±0.0094	0.9996±0.0001	0.9998±0.0001

Table A.37: Average cosine similarity results ± standard deviation for the English tense dataset with respect to $\mathcal{P}_1$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.9657±0.0099	0.9647±0.0082	0.9685±0.0072	0.9794±0.0087	0.9847±0.0079
11	0.9511±0.0129	0.9469±0.0112	0.9513±0.0107	0.9682±0.0171	0.9754±0.0144
10	0.9436±0.0164	0.9436±0.0127	0.9453±0.0123	0.9719±0.0176	0.9767±0.0151
9	0.9767±0.008	0.9728±0.0088	0.974±0.0091	0.9805±0.0143	0.9843±0.013
8	0.9752±0.0089	0.9721±0.0094	0.9746±0.0086	0.982±0.0144	0.9868±0.0117
7	0.9772±0.0071	0.9721±0.01	0.9756±0.0078	0.9809±0.0149	0.9868±0.0106
6	0.9809±0.0061	0.9768±0.0075	0.9801±0.0067	0.9846±0.0116	0.9895±0.0088
5	0.9764±0.0071	0.9718±0.0085	0.9754±0.0075	0.9818±0.0133	0.9885±0.0087
4	0.9856±0.0042	0.9837±0.0044	0.9852±0.0049	0.9923±0.0053	0.9935±0.0051
3	0.9873±0.0039	0.9858±0.0042	0.9861±0.0044	0.9939±0.0041	0.9931±0.005
2	0.9972±0.0007	0.9971±0.0008	0.9974±0.0007	0.9991±0.0004	0.9994±0.0002
1	0.9883±0.0036	0.9881±0.0036	0.9883±0.0036	0.9996±0.0001	0.9998±0.0001

Table A.38: Average cosine similarity results ± standard deviation for the English tense dataset with respect to $\mathcal{P}_2$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	3.6241±0.5266	4.0187±0.4347	3.7958±0.4359	3.2541±0.6666	2.7867±0.6893
11	17.4481±1.5613	18.0876±1.5541	17.721±1.5463	7.1787±1.8717	6.255±1.7645
10	18.2955±1.0504	18.5039±0.9988	18.3304±1.0231	6.3679±1.908	5.7623±1.7898
9	13.0301±0.8084	13.341±0.7596	13.2917±0.8445	4.9618±1.7142	4.4115±1.6503
8	14.3435±0.564	14.5201±0.5347	14.5144±0.6041	4.2615±1.6139	3.6287±1.4353
7	9.0442±0.7229	9.4135±0.668	9.2299±0.7504	4.384±1.6386	3.6439±1.3283
6	13.3656±0.6623	13.5916±0.5347	13.4291±0.6952	3.9793±1.4896	3.2753±1.2438
5	4.3555±0.5922	5.1119±0.8457	4.7047±0.7657	4.0122±1.4598	3.1997±1.1371
4	3.0083±0.4371	3.3254±0.4734	3.2009±0.5779	2.408±0.8164	2.1886±0.7884
3	11.7075±0.3711	11.8087±0.3542	11.7669±0.3763	1.867±0.6226	1.9836±0.6706
2	8.6316±0.3654	8.6655±0.3578	8.6587±0.3606	0.6678±0.1363	0.5284±0.095
1	8.7969±0.2186	8.8434±0.212	8.8204±0.2152	0.3821±0.0764	0.2974±0.0526

Table A.39: Average Euclidean distance results ± standard deviation for the English tense dataset with respect to $\mathcal{P}_1$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	4.1892±0.5404	4.3081±0.4609	4.084±0.4289	3.2541±0.6666	2.7867±0.6893
11	9.031±1.1918	9.5048±1.0111	9.0422±0.9818	7.1787±1.8717	6.255±1.7645
10	9.7646±1.2598	9.7965±1.0708	9.5706±1.0501	6.3679±1.908	5.7623±1.7898
9	5.6518±0.8898	6.1188±0.9296	5.9701±0.9671	4.9618±1.7142	4.4115±1.6503
8	5.2623±0.8919	5.5825±0.9021	5.3273±0.8618	4.2615±1.6139	3.6287±1.4353
7	5.0331±0.7532	5.5407±0.9491	5.1922±0.7946	4.384±1.6386	3.6439±1.3283
6	4.6681±0.7079	5.1395±0.8212	4.7551±0.7571	3.9793±1.4896	3.2753±1.2438
5	4.7853±0.6931	5.2291±0.7978	4.8848±0.7182	4.0122±1.4598	3.1997±1.1371
4	3.4248±0.4809	3.6354±0.4774	3.4597±0.5347	2.408±0.8164	2.1886±0.7884
3	2.7978±0.4168	2.9523±0.4241	2.9177±0.4504	1.867±0.6226	1.9836±0.6706
2	1.2104±0.1541	1.2332±0.1547	1.1632±0.151	0.6678±0.1363	0.5284±0.095
1	3.5903±0.4577	3.6264±0.4524	3.6043±0.4542	0.3821±0.0764	0.2974±0.0526

Table A.40: Average Euclidean distance results ± standard deviation for the English tense dataset with respect to $\mathcal{P}_2$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.65±0.0401	0.6493±0.0401	0.6497±0.0401	0.9998±0.0001	0.9998±0.0001
11	0.6515±0.0968	0.653±0.0966	0.6528±0.0966	1.0±0.0	1.0±0.0
10	0.8471±0.0238	0.8474±0.0238	0.8474±0.0238	1.0±0.0	1.0±0.0
9	0.7249±0.0493	0.7249±0.0494	0.7248±0.0494	1.0±0.0	1.0±0.0
8	0.7245±0.05	0.7239±0.05	0.724±0.05	1.0±0.0	1.0±0.0
7	0.5035±0.1476	0.5028±0.1478	0.503±0.1477	1.0±0.0	1.0±0.0
6	0.4832±0.0994	0.4827±0.0994	0.4829±0.0994	1.0±0.0	1.0±0.0
5	0.5644±0.0887	0.5642±0.0887	0.5642±0.0887	1.0±0.0	1.0±0.0
4	0.6592±0.0275	0.6594±0.0275	0.6595±0.0275	1.0±0.0	1.0±0.0
3	−0.0285±0.0315	−0.0285±0.0315	−0.0285±0.0315	1.0±0.0	1.0±0.0
2	0.0133±0.0505	0.0133±0.0506	0.0133±0.0505	1.0±0.0	1.0±0.0
1	0.7022±0.0969	0.7018±0.0969	0.7017±0.0969	0.9999±0.0001	0.9999±0.0001

Table A.41: Average cosine similarity results ± standard deviation for the English tense dataset with respect to $\mathcal{P}_1$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.9489±0.0045	0.9486±0.0046	0.9488±0.0045	0.9998±0.0001	0.9998±0.0001
11	0.9999±0.0	0.9999±0.0	0.9999±0.0	1.0±0.0	1.0±0.0
10	0.9999±0.0	0.9999±0.0	0.9999±0.0	1.0±0.0	1.0±0.0
9	0.9999±0.0	0.9999±0.0	0.9999±0.0	1.0±0.0	1.0±0.0
8	0.9999±0.0	0.9999±0.0	0.9999±0.0	1.0±0.0	1.0±0.0
7	0.9999±0.0	0.9999±0.0	0.9999±0.0	1.0±0.0	1.0±0.0
6	0.9978±0.0005	0.9977±0.0005	0.9977±0.0005	1.0±0.0	1.0±0.0
5	0.9999±0.0	0.9999±0.0001	0.9998±0.0001	1.0±0.0	1.0±0.0
4	0.8852±0.0105	0.8853±0.0105	0.8854±0.0105	1.0±0.0	1.0±0.0
3	0.6196±0.0866	0.6197±0.0866	0.6198±0.0866	1.0±0.0	1.0±0.0
2	0.9999±0.0	0.9999±0.0	0.9999±0.0	1.0±0.0	1.0±0.0
1	0.7993±0.0146	0.7992±0.0147	0.7991±0.0147	0.9999±0.0001	0.9999±0.0001

Table A.42: Average cosine similarity results ± standard deviation for the English tense dataset with respect to $\mathcal{P}_2$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	9.6367±0.1417	9.6632±0.1419	9.6299±0.1391	0.2408±0.0542	0.2158±0.0506
11	23.7087±0.4934	23.7002±0.4931	23.7017±0.493	0.1712±0.0456	0.1712±0.0422
10	20.5381±0.5893	20.5334±0.5893	20.5329±0.5895	0.1254±0.0317	0.1257±0.0315
9	23.2197±0.4623	23.2193±0.4626	23.2196±0.4628	0.1001±0.0278	0.1041±0.0281
8	23.1259±0.564	23.1302±0.5638	23.13±0.5636	0.1315±0.0396	0.1348±0.0411
7	25.5871±0.6397	25.5886±0.6401	25.5885±0.6399	0.1397±0.0279	0.1343±0.0281
6	24.7799±0.5197	24.7805±0.5193	24.7801±0.5195	0.1542±0.0326	0.1524±0.0309
5	23.8156±0.5287	23.8154±0.5286	23.8155±0.5287	0.1879±0.0382	0.1889±0.0378
4	20.4574±0.1662	20.4563±0.1662	20.4563±0.1662	0.0801±0.0154	0.082±0.0167
3	27.7628±0.3464	27.7625±0.3465	27.763±0.346	0.0858±0.0195	0.0874±0.0212
2	28.1137±0.4478	28.1124±0.4485	28.1129±0.4482	0.1107±0.0285	0.1078±0.0282
1	17.3039±0.5911	17.3027±0.5908	17.303±0.5908	0.3069±0.0806	0.3014±0.0802

Table A.43: Average Euclidean distance results ± standard deviation for the English tense dataset with respect to $\mathcal{P}_1$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	7.2405±0.1322	7.2673±0.1336	7.2336±0.1312	0.2408±0.0542	0.2158±0.0506
11	0.4361±0.0898	0.4479±0.0911	0.4493±0.0903	0.1712±0.0456	0.1712±0.0422
10	0.2972±0.043	0.3036±0.0412	0.3027±0.0412	0.1254±0.0317	0.1257±0.0315
9	0.2862±0.044	0.2871±0.0433	0.2898±0.043	0.1001±0.0278	0.1041±0.0281
8	0.3854±0.0711	0.378±0.0697	0.3822±0.0691	0.1315±0.0396	0.1348±0.0411
7	0.3785±0.0626	0.3895±0.0691	0.3859±0.0687	0.1397±0.0279	0.1343±0.0281
6	4.159±0.3958	4.1692±0.3944	4.165±0.3951	0.1542±0.0326	0.1524±0.0309
5	0.504±0.1055	0.5209±0.1084	0.5218±0.108	0.1879±0.0382	0.1889±0.0378
4	18.3287±0.2069	18.3278±0.2069	18.328±0.2069	0.0801±0.0154	0.082±0.0167
3	22.664±0.2463	22.6634±0.2463	22.6633±0.2463	0.0858±0.0195	0.0874±0.0212
2	0.3064±0.0285	0.3075±0.0261	0.3084±0.0256	0.1107±0.0285	0.1078±0.0282
1	17.0214±0.1238	17.0186±0.1244	17.0187±0.1244	0.3069±0.0806	0.3014±0.0802

Table A.44: Average Euclidean distance results ± standard deviation for the English tense dataset with respect to $\mathcal{P}_2$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.8572±0.0201	0.8523±0.0208	0.8517±0.0209	0.9784±0.0116	0.9815±0.0109
11	0.8451±0.0253	0.8205±0.0336	0.8224±0.0314	0.9671±0.018	0.9732±0.0163
10	0.7771±0.0283	0.7562±0.0353	0.7556±0.0343	0.9688±0.018	0.9742±0.017
9	0.7071±0.024	0.6971±0.0312	0.6923±0.0298	0.9751±0.0172	0.9787±0.0169
8	0.9429±0.0102	0.9259±0.021	0.9262±0.0204	0.9777±0.0164	0.9801±0.0164
7	0.9159±0.0157	0.9007±0.0228	0.9008±0.022	0.9795±0.0141	0.9816±0.0142
6	0.952±0.0121	0.9421±0.0181	0.9416±0.0178	0.9846±0.0111	0.9862±0.0116
5	0.9317±0.0127	0.9195±0.0179	0.92±0.0174	0.9841±0.0105	0.986±0.0108
4	0.9562±0.0079	0.9503±0.0103	0.9506±0.0099	0.9919±0.0068	0.9924±0.0068
3	0.8515±0.0183	0.8444±0.0195	0.8447±0.0195	0.9921±0.0072	0.9924±0.0073
2	0.685±0.0207	0.6844±0.0203	0.6847±0.0203	0.9992±0.0005	0.9992±0.0006
1	0.9568±0.0041	0.9564±0.004	0.9565±0.0041	0.9997±0.0002	0.9997±0.0002

Table A.45: Average cosine similarity results ± standard deviation for the English agreement dataset with respect to $\mathcal{P}_1$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.9735±0.0079	0.966±0.0105	0.9667±0.0102	0.9784±0.0116	0.9815±0.0109
11	0.9588±0.0113	0.9482±0.0145	0.9494±0.014	0.9671±0.018	0.9732±0.0163
10	0.9666±0.0091	0.953±0.0157	0.9547±0.0145	0.9688±0.018	0.9742±0.017
9	0.9746±0.0076	0.962±0.0162	0.963±0.0154	0.9751±0.0172	0.9787±0.0169
8	0.9768±0.0078	0.9644±0.0162	0.965±0.0157	0.9777±0.0164	0.9801±0.0164
7	0.9818±0.0057	0.9689±0.0136	0.9703±0.0129	0.9795±0.0141	0.9816±0.0142
6	0.9415±0.0133	0.9422±0.0124	0.9418±0.0127	0.9846±0.0111	0.9862±0.0116
5	0.9755±0.0073	0.9687±0.0114	0.9692±0.0107	0.9841±0.0105	0.986±0.0108
4	0.986±0.004	0.981±0.0077	0.9817±0.0072	0.9919±0.0068	0.9924±0.0068
3	0.9843±0.0047	0.9803±0.0076	0.98±0.0075	0.9921±0.0072	0.9924±0.0073
2	0.9978±0.0007	0.9975±0.0007	0.9975±0.0008	0.9992±0.0005	0.9992±0.0006
1	0.9987±0.0005	0.9986±0.0005	0.9986±0.0005	0.9997±0.0002	0.9997±0.0002

Table A.46: Average cosine similarity results ± standard deviation for the English agreement dataset with respect to $\mathcal{P}_2$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	8.9439±0.4801	8.9564±0.4768	8.998±0.4616	3.2639±0.8321	3.006±0.8219
11	18.1011±1.4689	18.5669±1.538	18.6028±1.5319	7.3518±2.0526	6.6035±2.0019
10	18.3663±1.0242	18.8928±1.1208	18.9082±1.1065	6.7498±1.9836	6.0839±1.9723
9	19.0106±0.6191	19.22±0.7536	19.334±0.7176	5.6137±1.9514	5.1087±2.0172
8	8.7561±0.6881	9.5644±1.0621	9.5537±1.0438	4.7528±1.7691	4.432±1.8334
7	10.2936±0.8316	10.9031±1.0191	10.8976±0.9913	4.5776±1.5965	4.2867±1.6566
6	8.2283±0.9548	8.7673±1.1555	8.7866±1.1447	4.0013±1.4549	3.7281±1.5523
5	9.0811±0.7152	9.5758±0.8253	9.547±0.8202	3.7919±1.2832	3.5146±1.3327
4	6.9179±0.5611	7.2108±0.6093	7.1982±0.5906	2.4171±0.9867	2.3241±0.9983
3	9.93±0.4337	10.0894±0.452	10.0927±0.4483	2.065±0.8925	2.0141±0.9205
2	12.1447±0.2371	12.1671±0.2373	12.1612±0.2406	0.6003±0.1817	0.621±0.1841
1	6.1074±0.2187	6.1239±0.216	6.1157±0.2173	0.3227±0.1044	0.3302±0.1001

Table A.47: Average Euclidean distance results ± standard deviation for the English agreement dataset with respect to $\mathcal{P}_1$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	3.6841±0.5218	4.1414±0.5907	4.1085±0.5913	3.2639±0.8321	3.006±0.8219
11	8.422±1.117	9.3633±1.2967	9.2795±1.2683	7.3518±2.0526	6.6035±2.0019
10	7.1804±0.9416	8.449±1.3595	8.3055±1.2888	6.7498±1.9836	6.0839±1.9723
9	5.9328±0.8563	7.1507±1.4513	7.0619±1.3978	5.6137±1.9514	5.1087±2.0172
8	5.0989±0.8197	6.234±1.3655	6.1797±1.3396	4.7528±1.7691	4.432±1.8334
7	4.5025±0.6827	5.8185±1.2398	5.6878±1.1985	4.5776±1.5965	4.2867±1.6566
6	8.8509±0.9186	8.8518±0.8246	8.8584±0.8461	4.0013±1.4549	3.7281±1.5523
5	4.9066±0.7171	5.5116±0.9717	5.4692±0.9258	3.7919±1.2832	3.5146±1.3327
4	3.3914±0.4748	3.9138±0.7459	3.8523±0.7138	2.4171±0.9867	2.3241±0.9983
3	3.1355±0.4647	3.4892±0.6389	3.526±0.6297	2.065±0.8925	2.0141±0.9205
2	1.0807±0.1638	1.1468±0.1654	1.1416±0.1745	0.6003±0.1817	0.621±0.1841
1	0.7402±0.1214	0.7669±0.1214	0.7554±0.1251	0.3227±0.1044	0.3302±0.1001

Table A.48: Average Euclidean distance results ± standard deviation for the English agreement dataset with respect to $\mathcal{P}_2$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.4725±0.0962	0.4727±0.0963	0.4724±0.0964	0.9996±0.0002	0.9997±0.0002
11	0.4902±0.0887	0.4905±0.0886	0.4903±0.0886	1.0±0.0	1.0±0.0
10	0.6493±0.0596	0.6494±0.0595	0.6495±0.0595	1.0±0.0	1.0±0.0
9	0.0351±0.0445	0.0351±0.0445	0.0351±0.0445	1.0±0.0	1.0±0.0
8	0.8381±0.0238	0.8384±0.0238	0.8382±0.0238	1.0±0.0	1.0±0.0
7	0.5176±0.129	0.5178±0.129	0.5176±0.129	1.0±0.0	1.0±0.0
6	0.7113±0.0357	0.7115±0.0357	0.7113±0.0356	1.0±0.0	1.0±0.0
5	0.6012±0.0646	0.6015±0.0646	0.6013±0.0646	1.0±0.0	1.0±0.0
4	0.6694±0.0754	0.6694±0.0754	0.6694±0.0754	1.0±0.0	1.0±0.0
3	0.6748±0.0505	0.6748±0.0505	0.6748±0.0505	1.0±0.0	1.0±0.0
2	0.0859±0.0676	0.0859±0.0676	0.0859±0.0676	1.0±0.0	1.0±0.0
1	0.6113±0.08	0.611±0.08	0.6112±0.08	0.9999±0.0001	0.9999±0.0001

Table A.49: Average cosine similarity results ± standard deviation for the English agreement dataset with respect to $\mathcal{P}_1$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.9993±0.0003	0.9992±0.0003	0.9993±0.0003	0.9996±0.0002	0.9997±0.0002
11	0.9999±0.0	0.9999±0.0	0.9999±0.0	1.0±0.0	1.0±0.0
10	0.9999±0.0	0.9999±0.0	0.9999±0.0	1.0±0.0	1.0±0.0
9	1.0±0.0	1.0±0.0	1.0±0.0	1.0±0.0	1.0±0.0
8	0.9999±0.0	0.9999±0.0	0.9999±0.0	1.0±0.0	1.0±0.0
7	0.9999±0.0	0.9999±0.0	0.9999±0.0	1.0±0.0	1.0±0.0
6	0.9771±0.0025	0.9773±0.0025	0.9772±0.0025	1.0±0.0	1.0±0.0
5	0.9687±0.008	0.9689±0.008	0.9688±0.0081	1.0±0.0	1.0±0.0
4	0.9981±0.0014	0.9981±0.0014	0.9981±0.0014	1.0±0.0	1.0±0.0
3	0.9645±0.0037	0.9645±0.0037	0.9645±0.0037	1.0±0.0	1.0±0.0
2	0.962±0.0088	0.9619±0.0088	0.962±0.0088	1.0±0.0	1.0±0.0
1	0.9996±0.0001	0.9995±0.0001	0.9995±0.0001	0.9999±0.0001	0.9999±0.0001

Table A.50: Average cosine similarity results ± standard deviation for the English agreement dataset with respect to $\mathscr{P}_2$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	10.1998±0.2247	10.216±0.2196	10.1972±0.2218	0.2992±0.064	0.2829±0.0669
11	24.6885±0.3751	24.6866±0.3748	24.688±0.3748	0.1886±0.0415	0.177±0.0434
10	23.6591±0.5631	23.6586±0.5628	23.6575±0.5626	0.1385±0.0326	0.1275±0.0348
9	29.8087±0.5432	29.8095±0.5436	29.8089±0.5433	0.1069±0.0341	0.1027±0.0338
8	19.8419±0.3702	19.8389±0.3704	19.8401±0.3701	0.1407±0.0479	0.1352±0.0461
7	25.487±0.6505	25.4862±0.6507	25.4868±0.6506	0.1394±0.0325	0.1371±0.0293
6	22.9451±0.3874	22.9445±0.3876	22.9453±0.3874	0.1737±0.0389	0.1724±0.0366
5	23.6932±0.3584	23.693±0.3585	23.6937±0.3584	0.225±0.0455	0.2273±0.045
4	20.2683±0.5523	20.2687±0.5522	20.2685±0.5522	0.0686±0.0197	0.0715±0.0199
3	21.4935±0.4781	21.4934±0.4781	21.4934±0.4781	0.0629±0.022	0.0639±0.0218
2	26.9656±0.5071	26.9659±0.5073	26.9657±0.5071	0.0837±0.0309	0.0832±0.0305
1	17.9696±0.3931	17.9749±0.3924	17.9707±0.393	0.2392±0.0896	0.2289±0.0884

Table A.51: Average Euclidean distance results ± standard deviation for the English agreement dataset with respect to $\mathscr{P}_1$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.43±0.0846	0.4652±0.0829	0.4437±0.0876	0.2992±0.064	0.2829±0.0669
11	0.4192±0.0976	0.4272±0.0953	0.4253±0.0967	0.1886±0.0415	0.177±0.0434
10	0.3401±0.0363	0.3435±0.0364	0.344±0.0392	0.1385±0.0326	0.1275±0.0348
9	0.2668±0.0383	0.2744±0.0396	0.2718±0.0423	0.1069±0.0341	0.1027±0.0338
8	0.3786±0.0596	0.3891±0.0614	0.3873±0.063	0.1407±0.0479	0.1352±0.0461
7	0.3299±0.0636	0.3359±0.0622	0.3335±0.0643	0.1394±0.0325	0.1371±0.0293
6	10.365±0.3532	10.3589±0.3533	10.3637±0.3533	0.1737±0.0389	0.1724±0.0366
5	11.5292±0.9299	11.5221±0.9304	11.5273±0.9303	0.225±0.0455	0.2273±0.045
4	3.414±1.2622	3.4147±1.262	3.4138±1.2622	0.0686±0.0197	0.0715±0.0199
3	11.6076±0.4287	11.6072±0.4287	11.6073±0.4287	0.0629±0.022	0.0639±0.0218
2	14.2107±0.9555	14.2114±0.9554	14.2109±0.9554	0.0837±0.0309	0.0832±0.0305
1	0.6363±0.1011	0.6404±0.1027	0.6442±0.1019	0.2392±0.0896	0.2289±0.0884

Table A.52: Average Euclidean distance results ± standard deviation for the English agreement dataset with respect to $\mathscr{P}_2$ and RoBERTa. Each row represents a different layer starting from the last one (12).

A.2.2 Italian

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.7958±0.1248	0.7322±0.1477	0.7343±0.1452	0.8446±0.1244	0.8523±0.1271
11	0.779±0.1722	0.7537±0.1743	0.7302±0.1878	0.9097±0.1262	0.904±0.1334
10	0.8945±0.1018	0.8888±0.1019	0.8813±0.1208	0.9452±0.0807	0.9468±0.0823
9	0.959±0.0362	0.9566±0.0364	0.9542±0.0411	0.9722±0.038	0.9735±0.036
8	0.8023±0.054	0.7999±0.0524	0.7938±0.0587	0.9823±0.0202	0.9831±0.0203
7	0.9808±0.0093	0.9787±0.01	0.9788±0.0117	0.9861±0.0112	0.987±0.0119
6	0.5552±0.0532	0.5457±0.0533	0.5446±0.0526	0.9886±0.0071	0.9897±0.0074
5	0.9275±0.0144	0.9222±0.0159	0.9221±0.0152	0.9882±0.0069	0.9897±0.0069
4	0.6992±0.0438	0.6955±0.044	0.692±0.0439	0.9903±0.0055	0.9914±0.0058
3	0.9806±0.0051	0.9774±0.0049	0.9785±0.0053	0.9863±0.0084	0.9878±0.0087
2	0.9782±0.0065	0.9751±0.0062	0.9763±0.0062	0.9868±0.0088	0.988±0.0087
1	0.9633±0.0069	0.9609±0.007	0.9619±0.0067	0.9907±0.006	0.9916±0.0059

Table A.53: Average cosine similarity results ± standard deviation for the Italian tense dataset with respect to $\mathcal{P}_1$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.8508±0.0903	0.8037±0.1114	0.8016±0.1154	0.8446±0.1244	0.8523±0.1271
11	0.853±0.1474	0.8376±0.1541	0.8111±0.1745	0.9097±0.1262	0.904±0.1334
10	0.9245±0.0718	0.9189±0.076	0.912±0.0897	0.9452±0.0807	0.9468±0.0823
9	0.9675±0.0264	0.9646±0.0283	0.9641±0.0293	0.9722±0.038	0.9735±0.036
8	0.9718±0.0175	0.9701±0.0179	0.9694±0.0204	0.9823±0.0202	0.9831±0.0203
7	0.9742±0.0111	0.9729±0.0117	0.9726±0.0142	0.9861±0.0112	0.987±0.0119
6	0.9824±0.0054	0.981±0.0053	0.9819±0.006	0.9886±0.0071	0.9897±0.0074
5	0.9811±0.0047	0.979±0.0045	0.9803±0.0045	0.9882±0.0069	0.9897±0.0069
4	0.984±0.0036	0.9827±0.0032	0.9833±0.0034	0.9903±0.0055	0.9914±0.0058
3	0.9787±0.0053	0.9761±0.0047	0.9772±0.0051	0.9863±0.0084	0.9878±0.0087
2	0.9749±0.0078	0.9729±0.007	0.9738±0.007	0.9868±0.0088	0.988±0.0087
1	0.9803±0.005	0.9787±0.005	0.9794±0.0046	0.9907±0.006	0.9916±0.0059

Table A.54: Average cosine similarity results ± standard deviation for the Italian tense dataset with respect to $\mathcal{P}_2$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	5.4168±1.4592	5.6789±1.3979	5.5829±1.3917	4.812±2.114	4.5978±2.1659
11	7.2082±2.7873	7.8377±2.9475	7.6342±2.9112	5.1406±3.3048	5.0311±3.4894
10	5.2719±1.388	5.4845±1.5902	5.2991±1.4432	3.7164±2.1877	3.5675±2.1905
9	4.1672±1.2578	4.2995±1.3259	4.2742±1.2911	3.1038±1.7334	2.9969±1.7004
8	9.3852±0.5514	9.4179±0.5597	9.4265±0.548	2.6724±1.256	2.5663±1.2842
7	2.9648±0.6001	3.1085±0.6077	3.0893±0.697	2.386±0.869	2.2807±0.9409
6	12.7543±0.402	12.7585±0.4	12.7768±0.4045	2.2043±0.6735	2.0776±0.7317
5	5.6614±0.4391	5.7486±0.4526	5.7235±0.4368	1.9438±0.5606	1.7948±0.6008
4	10.4804±0.448	10.5052±0.446	10.5173±0.4453	1.9027±0.5454	1.7712±0.5926
3	2.6449±0.3407	2.8432±0.3007	2.7707±0.3387	2.1216±0.6411	1.9768±0.6881
2	2.5205±0.3594	2.6881±0.3258	2.6205±0.333	1.8847±0.6068	1.7749±0.6338
1	3.4168±0.3036	3.5223±0.2933	3.4779±0.2878	1.6067±0.5318	1.5045±0.5441

Table A.55: Average Euclidean distance results ± standard deviation for the Italian tense dataset with respect to $\mathcal{P}_1$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	4.8267±1.5438	5.1376±1.5137	5.1066±1.5778	4.812±2.114	4.5978±2.1659
11	5.9791±2.3666	6.4246±2.5736	6.4585±2.6617	5.1406±3.3048	5.0311±3.4894
10	4.6023±1.3476	4.8044±1.553	4.7105±1.4954	3.7164±2.1877	3.5675±2.1905
9	3.6271±1.1931	3.7799±1.2825	3.7498±1.2486	3.1038±1.7334	2.9969±1.7004
8	3.6473±0.8772	3.7468±0.8929	3.7323±0.9451	2.6724±1.256	2.5663±1.2842
7	3.4234±0.663	3.4924±0.6614	3.4911±0.7666	2.386±0.869	2.2807±0.9409
6	2.8461±0.4377	2.9382±0.3941	2.868±0.433	2.2043±0.6735	2.0776±0.7317
5	2.548±0.3241	2.6809±0.286	2.5933±0.2956	1.9438±0.5606	1.7948±0.6008
4	2.5362±0.29	2.6381±0.2439	2.5856±0.264	1.9027±0.5454	1.7712±0.5926
3	2.7445±0.3302	2.9018±0.2781	2.8294±0.3108	2.1216±0.6411	1.9768±0.6881
2	2.6839±0.3971	2.7894±0.3518	2.7369±0.3575	1.8847±0.6068	1.7749±0.6338
1	2.4173±0.3025	2.5178±0.2926	2.4802±0.2782	1.6067±0.5318	1.5045±0.5441

Table A.56: Average Euclidean distance results ± standard deviation for the Italian tense dataset with respect to $\mathcal{P}_2$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.8609±0.0551	0.8587±0.0459	0.8572±0.0477	0.9749±0.0372	0.9723±0.0378
11	0.5827±0.05	0.5814±0.0492	0.5826±0.0492	0.998±0.0225	0.9979±0.0227
10	0.9174±0.0662	0.9181±0.0613	0.9172±0.0643	0.9786±0.0498	0.9773±0.0613
9	0.7871±0.0602	0.7911±0.0457	0.7851±0.0506	0.9566±0.0663	0.9544±0.0737
8	0.9416±0.0517	0.9408±0.0386	0.9406±0.0412	0.9672±0.0491	0.9672±0.0496
7	0.9445±0.0587	0.9417±0.0596	0.9438±0.0606	0.9656±0.0589	0.9663±0.0589
6	0.9249±0.0401	0.9171±0.0474	0.921±0.0426	0.9608±0.0478	0.9633±0.0434
5	0.7174±0.0471	0.7168±0.0469	0.7171±0.0471	0.9977±0.004	0.9979±0.0038
4	0.5494±0.056	0.5504±0.0557	0.5508±0.056	0.9987±0.0018	0.9988±0.0016
3	0.9503±0.0195	0.9493±0.0191	0.9495±0.0195	0.9883±0.0129	0.9884±0.0124
2	0.8216±0.0151	0.8218±0.0151	0.8208±0.0151	0.9983±0.0017	0.9981±0.0019
1	0.9744±0.0084	0.9726±0.0091	0.9718±0.0106	0.9859±0.0129	0.9837±0.0154

Table A.57: Average cosine similarity results ± standard deviation for the Italian tense dataset with respect to $\mathcal{P}_1$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.9545±0.0348	0.9532±0.0165	0.9506±0.0216	0.9749±0.0372	0.9723±0.0378
11	0.9959±0.043	0.9968±0.0364	0.9969±0.0359	0.998±0.0225	0.9979±0.0227
10	0.9703±0.0551	0.9673±0.0471	0.9667±0.0566	0.9786±0.0498	0.9773±0.0613
9	0.9334±0.0637	0.9341±0.0487	0.9298±0.061	0.9566±0.0663	0.9544±0.0737
8	0.9465±0.0431	0.9457±0.0287	0.9457±0.0344	0.9672±0.0491	0.9672±0.0496
7	0.949±0.0522	0.9462±0.052	0.9487±0.0535	0.9656±0.0589	0.9663±0.0589
6	0.9212±0.0411	0.9144±0.046	0.9187±0.0418	0.9608±0.0478	0.9633±0.0434
5	0.9955±0.0039	0.9952±0.0036	0.9953±0.0036	0.9977±0.004	0.9979±0.0038
4	0.9971±0.0018	0.997±0.0019	0.997±0.0021	0.9987±0.0018	0.9988±0.0016
3	0.98±0.01	0.9795±0.0104	0.9795±0.0108	0.9883±0.0129	0.9884±0.0124
2	0.996±0.0018	0.9957±0.002	0.996±0.0019	0.9983±0.0017	0.9981±0.0019
1	0.9335±0.018	0.9321±0.0172	0.9324±0.0187	0.9859±0.0129	0.9837±0.0154

Table A.58: Average cosine similarity results ± standard deviation for the Italian tense dataset with respect to $\mathcal{P}_2$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	2.9717±0.3568	3.0018±0.2914	3.0102±0.3027	1.2108±0.519	1.2772±0.5304
11	2.6562±0.2401	2.6505±0.1915	2.6502±0.1959	0.1186±0.2507	0.1178±0.2549
10	2.2452±0.6451	2.2706±0.5331	2.2851±0.6965	0.9343±0.8281	0.9487±0.946
9	5.0851±0.8669	5.0902±0.7831	5.1515±0.923	2.2237±1.57	2.2732±1.6759
8	3.5661±1.0936	3.6091±0.9508	3.6251±1.0327	2.5022±1.4076	2.5041±1.4279
7	3.5264±1.5498	3.6173±1.5882	3.547±1.5764	2.4837±1.8433	2.4625±1.8197
6	3.5836±0.9815	3.7514±1.0908	3.6755±1.0218	2.3462±1.4256	2.2879±1.347
5	5.0221±0.1475	5.0233±0.1461	5.0235±0.1462	0.3391±0.2465	0.3255±0.2369
4	6.6767±0.1265	6.6754±0.1269	6.6772±0.1269	0.3207±0.195	0.3186±0.1842
3	2.718±0.555	2.7234±0.5515	2.7274±0.5461	0.8802±0.4845	0.8753±0.4783
2	3.4977±0.0756	3.499±0.0766	3.5001±0.076	0.2749±0.1259	0.284±0.135
1	1.1833±0.207	1.2121±0.2214	1.2377±0.2488	0.8057±0.3818	0.8615±0.4343

Table A.59: Average Euclidean distance results ± standard deviation for the Italian tense dataset with respect to $\mathcal{P}_1$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	1.7113±0.3772	1.7568±0.2534	1.7939±0.2923	1.2108±0.519	1.2772±0.5304
11	0.1736±0.2987	0.166±0.1915	0.1636±0.196	0.1186±0.2507	0.1178±0.2549
10	1.1888±0.7504	1.2635±0.6558	1.2692±0.8115	0.9343±0.8281	0.9487±0.946
9	2.8565±1.342	2.9037±1.2852	2.9767±1.4299	2.2237±1.57	2.2732±1.6759
8	3.4296±1.0062	3.4735±0.8553	3.4786±0.9489	2.5022±1.4076	2.5041±1.4279
7	3.3919±1.4563	3.4907±1.4878	3.4012±1.4726	2.4837±1.8433	2.4625±1.8197
6	3.6774±0.9371	3.8237±1.0102	3.7389±0.9488	2.3462±1.4256	2.2879±1.347
5	0.6043±0.1878	0.6251±0.1924	0.6158±0.1921	0.3391±0.2465	0.3255±0.2369
4	0.5748±0.1448	0.5817±0.1499	0.5797±0.1578	0.3207±0.195	0.3186±0.1842
3	1.2574±0.3377	1.2701±0.3405	1.2706±0.3502	0.8802±0.4845	0.8753±0.4783
2	0.4524±0.0947	0.468±0.1031	0.4525±0.0962	0.2749±0.1259	0.284±0.135
1	1.9424±0.2828	1.9308±0.2689	1.9573±0.2955	0.8057±0.3818	0.8615±0.4343

Table A.60: Average Euclidean distance results ± standard deviation for the Italian tense dataset with respect to $\mathcal{P}_2$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.7601±0.1161	0.7033±0.1351	0.6843±0.1491	0.7739±0.1682	0.7563±0.1793
11	0.3673±0.2345	0.357±0.2234	0.3668±0.2238	0.8245±0.2165	0.7938±0.238
10	0.858±0.1192	0.8542±0.1182	0.8642±0.1099	0.882±0.1375	0.8708±0.1392
9	0.8556±0.087	0.8447±0.0886	0.8562±0.0788	0.9446±0.0609	0.9429±0.0578
8	0.6305±0.0927	0.6229±0.0925	0.633±0.0828	0.9629±0.0378	0.9623±0.0357
7	0.9633±0.0154	0.9541±0.019	0.9558±0.0136	0.9707±0.0219	0.9703±0.0212
6	0.9783±0.0078	0.9699±0.008	0.9699±0.009	0.9792±0.0106	0.9789±0.0105
5	0.9754±0.0066	0.9634±0.0092	0.9647±0.0087	0.9781±0.0097	0.9776±0.009
4	0.9386±0.0106	0.9259±0.0127	0.9273±0.0134	0.9818±0.0077	0.9811±0.0073
3	0.7323±0.0235	0.7149±0.0242	0.7247±0.025	0.9722±0.0121	0.9731±0.0111
2	0.9227±0.013	0.906±0.0142	0.9075±0.0135	0.9723±0.0132	0.9741±0.0111
1	0.9872±0.0046	0.9698±0.009	0.9719±0.0088	0.9744±0.0116	0.9759±0.0103

Table A.61: Average cosine similarity results ± standard deviation for the Italian agreement dataset with respect to $\mathcal{P}_1$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.8367±0.0941	0.7845±0.1221	0.7611±0.1347	0.7739±0.1682	0.7563±0.1793
11	0.8797±0.1099	0.8555±0.1411	0.8334±0.1531	0.8245±0.2165	0.7938±0.238
10	0.9124±0.0713	0.9±0.0787	0.8953±0.0825	0.882±0.1375	0.8708±0.1392
9	0.952±0.0352	0.9497±0.0357	0.9502±0.0329	0.9446±0.0609	0.9429±0.0578
8	0.9673±0.0236	0.9631±0.0268	0.9651±0.0208	0.9629±0.0378	0.9623±0.0357
7	0.9727±0.0147	0.9677±0.0171	0.9697±0.0122	0.9707±0.0219	0.9703±0.0212
6	0.9823±0.0071	0.9767±0.0059	0.9759±0.0078	0.9792±0.0106	0.9789±0.0105
5	0.979±0.006	0.9736±0.0056	0.9746±0.0058	0.9781±0.0097	0.9776±0.009
4	0.9843±0.0044	0.9784±0.0046	0.9778±0.005	0.9818±0.0077	0.9811±0.0073
3	0.9754±0.0076	0.9662±0.0075	0.9687±0.0075	0.9722±0.0121	0.9731±0.0111
2	0.9827±0.0059	0.9661±0.0098	0.9689±0.0084	0.9723±0.0132	0.9741±0.0111
1	0.979±0.0073	0.9694±0.0073	0.9736±0.0075	0.9744±0.0116	0.9759±0.0103

Table A.62: Average cosine similarity results ± standard deviation for the Italian agreement dataset with respect to $\mathcal{P}_2$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	5.4438±1.5626	5.8417±1.6362	6.2832±1.9404	5.5343±2.6616	5.8744±2.7034
11	8.6865±4.0383	8.3916±3.6329	9.5803±4.266	5.8694±4.7592	6.7475±4.9532
10	5.3161±1.6591	5.5915±1.7738	5.9536±1.984	5.152±3.1307	5.4926±3.0373
9	7.3267±1.1039	7.4355±1.0406	7.5213±1.1787	4.3328±2.2459	4.486±2.0744
8	11.6555±0.5727	11.6568±0.5484	11.7161±0.5966	3.8258±1.7144	3.9124±1.5945
7	4.2243±0.7344	4.6052±0.7635	4.5777±0.6718	3.4763±1.1634	3.5189±1.1292
6	3.369±0.5112	3.8092±0.4684	3.8555±0.5248	3.0001±0.737	3.0447±0.709
5	3.303±0.4244	3.76±0.4285	3.7815±0.4424	2.6684±0.5826	2.7277±0.537
4	6.1292±0.4184	6.3872±0.4225	6.3904±0.4361	2.634±0.5494	2.7017±0.5199
3	9.0969±0.2575	9.2828±0.2522	9.2124±0.2525	3.0455±0.6509	3.0107±0.606
2	4.8729±0.358	5.3068±0.3545	5.2351±0.3404	2.7883±0.6696	2.7046±0.5848
1	1.9481±0.3241	2.9987±0.4484	2.8927±0.457	2.735±0.6284	2.6615±0.5792

Table A.63: Average Euclidean distance results ± standard deviation for the Italian agreement dataset with respect to $\mathcal{P}_1$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	4.6092±1.6326	5.1607±1.8439	5.6487±2.1204	5.5343±2.6616	5.8744±2.7034
11	4.7928±3.0809	5.2689±3.5323	6.3815±4.1214	5.8694±4.7592	6.7475±4.9532
10	4.5281±1.8897	4.8848±2.0282	5.2811±2.1313	5.152±3.1307	5.4926±3.0373
9	4.264±1.3661	4.344±1.2512	4.4919±1.2939	4.3328±2.2459	4.486±2.0744
8	3.7993±1.0542	4.0044±0.9795	4.0216±0.9838	3.8258±1.7144	3.9124±1.5945
7	3.4722±0.7895	3.7655±0.7864	3.6819±0.688	3.4763±1.1634	3.5189±1.1292
6	2.8138±0.5241	3.2328±0.4042	3.2938±0.507	3.0001±0.737	3.0447±0.709
5	2.6646±0.3802	2.9778±0.3146	2.9473±0.3417	2.6684±0.5826	2.7277±0.537
4	2.4911±0.3441	2.9199±0.3084	2.9619±0.3372	2.634±0.5494	2.7017±0.5199
3	2.8978±0.4402	3.3951±0.3704	3.283±0.3867	3.0455±0.6509	3.0107±0.606
2	2.2185±0.3646	3.1253±0.455	2.9835±0.4113	2.7883±0.6696	2.7046±0.5848
1	2.4894±0.4206	3.019±0.3624	2.8073±0.4004	2.735±0.6284	2.6615±0.5792

Table A.64: Average Euclidean distance results ± standard deviation for the Italian agreement dataset with respect to $\mathcal{P}_2$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.9347±0.123	0.9361±0.0876	0.9387±0.0982	0.9474±0.13	0.949±0.1381
11	0.9307±0.1191	0.9401±0.0836	0.9374±0.0967	0.9781±0.1265	0.9752±0.1363
10	0.8882±0.126	0.8901±0.1063	0.8886±0.1161	0.9473±0.1395	0.9506±0.1382
9	0.9105±0.1041	0.9012±0.1053	0.9034±0.1053	0.9212±0.1271	0.9289±0.1217
8	0.9327±0.0863	0.9231±0.0984	0.926±0.0915	0.9384±0.1098	0.9433±0.1073
7	0.9173±0.0916	0.91±0.1048	0.9116±0.1028	0.9357±0.1075	0.9401±0.1058
6	0.9262±0.0435	0.9196±0.0468	0.9198±0.0476	0.9478±0.0608	0.9529±0.0548
5	0.395±0.0614	0.3936±0.0616	0.3921±0.0615	0.9958±0.0076	0.996±0.0069
4	0.7665±0.0406	0.7664±0.0405	0.766±0.0409	0.9973±0.0027	0.9975±0.0024
3	0.8026±0.0442	0.794±0.042	0.7969±0.0448	0.9782±0.0172	0.9784±0.0167
2	0.9497±0.0061	0.948±0.006	0.9484±0.006	0.9968±0.0019	0.9969±0.0017
1	0.8602±0.0282	0.8444±0.0302	0.8557±0.0281	0.9742±0.0175	0.9756±0.0168

Table A.65: Average cosine similarity results ± standard deviation for the Italian agreement dataset with respect to $\mathcal{P}_1$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.9403±0.1126	0.9438±0.0548	0.9445±0.0883	0.9474±0.13	0.949±0.1381
11	0.9765±0.121	0.9902±0.0565	0.9858±0.0858	0.9781±0.1265	0.9752±0.1363
10	0.9486±0.1165	0.9513±0.0961	0.9501±0.1083	0.9473±0.1395	0.9506±0.1382
9	0.9253±0.0852	0.9113±0.1007	0.9136±0.1037	0.9212±0.1271	0.9289±0.1217
8	0.9366±0.0717	0.9284±0.0804	0.9306±0.0816	0.9384±0.1098	0.9433±0.1073
7	0.9345±0.0728	0.9234±0.0947	0.9246±0.0934	0.9357±0.1075	0.9401±0.1058
6	0.9304±0.0404	0.925±0.043	0.9253±0.0433	0.9478±0.0608	0.9529±0.0548
5	0.972±0.0067	0.972±0.0062	0.9717±0.0063	0.9958±0.0076	0.996±0.0069
4	0.9927±0.0024	0.9923±0.0028	0.9926±0.0024	0.9973±0.0027	0.9975±0.0024
3	0.9782±0.0102	0.9744±0.011	0.9738±0.0117	0.9782±0.0172	0.9784±0.0167
2	0.9818±0.0048	0.9813±0.0048	0.9815±0.0048	0.9968±0.0019	0.9969±0.0017
1	0.9721±0.0092	0.9653±0.0122	0.9668±0.0115	0.9742±0.0175	0.9756±0.0168

Table A.66: Average cosine similarity results ± standard deviation for the Italian agreement dataset with respect to $\mathcal{P}_2$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	1.8521±0.8929	1.9373±0.6401	1.8724±0.7093	1.5541±1.1323	1.4949±1.1736
11	1.7521±1.325	1.6203±0.6831	1.6705±1.0298	0.3871±1.5456	0.435±1.7639
10	2.6286±1.5625	2.5202±0.939	2.5762±1.3129	1.4858±1.9599	1.4523±2.0587
9	3.5133±2.0057	3.6542±1.8471	3.6205±1.891	3.1718±2.4875	3.0209±2.3999
8	3.7247±1.6152	4.0018±1.6441	3.9185±1.6141	3.3889±2.2169	3.2379±2.1316
7	4.1554±1.9053	4.3083±2.0571	4.2818±2.0372	3.4063±2.4065	3.2852±2.3316
6	3.5655±1.0088	3.7125±1.0687	3.712±1.0604	2.7953±1.5386	2.6632±1.4485
5	5.7154±0.0995	5.7195±0.1004	5.7188±0.1003	0.4537±0.3426	0.4485±0.3281
4	5.9881±0.1985	5.9933±0.1961	5.9907±0.1971	0.5047±0.2213	0.4831±0.2063
3	4.3995±0.3253	4.4513±0.3084	4.436±0.3177	1.276±0.5141	1.2696±0.5195
2	2.4767±0.1026	2.4864±0.0994	2.4868±0.1005	0.3992±0.1127	0.387±0.1098
1	2.7814±0.3014	2.8898±0.328	2.791±0.2968	1.1638±0.4079	1.1261±0.4081

Table A.67: Average Euclidean distance results ± standard deviation for the Italian agreement dataset with respect to $\mathscr{P}_1$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	1.7828±0.839	1.8645±0.5381	1.8002±0.6551	1.5541±1.1323	1.4949±1.1736
11	0.5052±1.4543	0.3437±0.5791	0.4088±1.0708	0.3871±1.5456	0.435±1.7639
10	1.5444±1.6801	1.4642±1.0628	1.5051±1.4342	1.4858±1.9599	1.4523±2.0587
9	3.2034±1.886	3.4227±1.8567	3.3757±1.9143	3.1718±2.4875	3.0209±2.3999
8	3.6508±1.4721	3.8915±1.4995	3.8001±1.5286	3.3889±2.2169	3.2379±2.1316
7	3.7233±1.7083	3.9641±1.9587	3.9414±1.9354	3.4063±2.4065	3.2852±2.3316
6	3.4674±0.9722	3.59±1.0272	3.5876±1.0163	2.7953±1.5386	2.6632±1.4485
5	2.9967±0.2281	2.9976±0.2265	2.9965±0.2278	0.4537±0.3426	0.4485±0.3281
4	2.178±0.4298	2.1932±0.4186	2.184±0.4227	0.5047±0.2213	0.4831±0.2063
3	1.3471±0.2922	1.4616±0.3122	1.4742±0.3253	1.276±0.5141	1.2696±0.5195
2	1.6784±0.1884	1.6822±0.1874	1.6833±0.1878	0.3992±0.1127	0.387±0.1098
1	1.2508±0.2125	1.3891±0.2573	1.3526±0.237	1.1638±0.4079	1.1261±0.4081

Table A.68: Average Euclidean distance results ± standard deviation for the Italian agreement dataset with respect to $\mathscr{P}_2$ and RoBERTa. Each row represents a different layer starting from the last one (12).

A.2.3 German

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.9306±0.0286	0.8948±0.0351	0.9049±0.0341	0.9151±0.0431	0.9218±0.0415
11	0.7306±0.0421	0.7012±0.0474	0.7132±0.0462	0.9436±0.0282	0.9469±0.0283
10	0.7953±0.0383	0.7664±0.0437	0.7778±0.0433	0.9308±0.0371	0.9308±0.0379
9	0.7535±0.0479	0.7383±0.0484	0.7445±0.0463	0.9466±0.0364	0.9454±0.037
8	0.9102±0.0216	0.8991±0.026	0.8994±0.0258	0.9633±0.0266	0.9614±0.027
7	0.9521±0.0151	0.9443±0.0194	0.9449±0.0197	0.9705±0.0228	0.9688±0.0235
6	0.9775±0.0098	0.9723±0.0148	0.9718±0.015	0.9801±0.0169	0.979±0.0172
5	0.9879±0.0068	0.984±0.0123	0.9839±0.0125	0.9879±0.0128	0.9875±0.0131
4	0.9455±0.0152	0.9406±0.0177	0.942±0.0179	0.9956±0.0076	0.9955±0.0075
3	0.8649±0.0129	0.862±0.0133	0.8637±0.0134	0.9982±0.002	0.9983±0.0019
2	0.9665±0.0056	0.966±0.0059	0.9658±0.0059	0.999±0.0013	0.999±0.0013
1	0.8162±0.0221	0.8162±0.0221	0.8163±0.0221	0.9994±0.0005	0.9994±0.0004

Table A.69: Average cosine similarity results ± standard deviation for the German tense dataset with respect to $\mathcal{P}_1$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.921±0.0264	0.8964±0.0308	0.9083±0.0288	0.9151±0.0431	0.9218±0.0415
11	0.7938±0.0279	0.7716±0.0289	0.7883±0.0276	0.9436±0.0282	0.9469±0.0283
10	0.9277±0.0237	0.9094±0.0274	0.9148±0.0263	0.9308±0.0371	0.9308±0.0379
9	0.9529±0.0212	0.9356±0.0308	0.938±0.0305	0.9466±0.0364	0.9454±0.037
8	0.9042±0.02	0.8996±0.0184	0.9041±0.0177	0.9633±0.0266	0.9614±0.027
7	0.9667±0.0133	0.9599±0.0187	0.96±0.0189	0.9705±0.0228	0.9688±0.0235
6	0.9748±0.01	0.9716±0.0145	0.9715±0.0146	0.9801±0.0169	0.979±0.0172
5	0.9854±0.0066	0.9829±0.0109	0.9832±0.0111	0.9879±0.0128	0.9875±0.0131
4	0.995±0.004	0.9932±0.0073	0.9933±0.0071	0.9956±0.0076	0.9955±0.0075
3	0.9962±0.0013	0.9958±0.0019	0.9959±0.0018	0.9982±0.002	0.9983±0.0019
2	0.9973±0.0012	0.9972±0.0014	0.9972±0.0013	0.999±0.0013	0.999±0.0013
1	0.9983±0.0004	0.9982±0.0004	0.9983±0.0004	0.9994±0.0005	0.9994±0.0004

Table A.70: Average cosine similarity results ± standard deviation for the German tense dataset with respect to $\mathcal{P}_2$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	7.1654±1.2778	8.7077±1.361	8.2758±1.352	7.864±1.9187	7.5265±1.8929
11	19.1391±1.0502	19.781±1.0842	19.5336±1.078	8.8941±2.1759	8.6071±2.2015
10	13.7187±1.0336	14.4123±1.0719	14.2021±1.0835	8.0743±2.0642	8.0788±2.1184
9	14.207±1.0709	14.5769±1.0667	14.4471±1.0374	6.7724±1.9849	6.839±2.0249
8	9.2938±0.9781	9.7502±1.0584	9.7298±1.0476	5.6537±1.7349	5.7954±1.7647
7	6.6918±0.9339	7.1249±1.0435	7.1055±1.0471	5.001±1.5913	5.1432±1.6227
6	4.6229±0.8381	5.0948±1.085	5.1505±1.0806	4.1871±1.4013	4.3204±1.3957
5	3.2547±0.6566	3.6733±1.0062	3.6857±1.018	3.094±1.1561	3.1591±1.1585
4	7.7645±0.8514	7.9627±0.9178	7.9021±0.9281	1.7093±0.9137	1.7334±0.9004
3	8.9918±0.2859	9.0298±0.2952	9.0106±0.2939	0.8551±0.3593	0.8545±0.3423
2	6.925±0.4279	6.9444±0.4338	6.9489±0.4333	0.7437±0.3444	0.7544±0.3337
1	12.1665±0.4597	12.1662±0.4599	12.1611±0.4596	0.6715±0.214	0.6198±0.2132

Table A.71: Average Euclidean distance results ± standard deviation for the German tense dataset with respect to $\mathcal{P}_1$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	7.6078±1.2105	8.6605±1.263	8.1485±1.2084	7.864±1.9187	7.5265±1.8929
11	17.9615±0.8854	18.468±0.8328	18.1059±0.8363	8.8941±2.1759	8.6071±2.2015
10	8.2704±1.3358	9.2245±1.3905	8.9873±1.3589	8.0743±2.0642	8.0788±2.1184
9	6.4133±1.2353	7.4736±1.5344	7.3329±1.5259	6.7724±1.9849	6.839±2.0249
8	9.8254±0.9249	10.0258±0.8576	9.8505±0.8354	5.6537±1.7349	5.7954±1.7647
7	5.4447±0.9454	5.9351±1.1484	5.9402±1.1517	5.001±1.5913	5.1432±1.6227
6	4.8598±0.8316	5.1321±1.0499	5.1492±1.0494	4.1871±1.4013	4.3204±1.3957
5	3.5644±0.6421	3.813±0.917	3.7812±0.9321	3.094±1.1561	3.1591±1.1585
4	2.0079±0.5116	2.2614±0.8241	2.2423±0.8173	1.7093±0.9137	1.7334±0.9004
3	1.3468±0.1953	1.4092±0.2633	1.3869±0.2524	0.8551±0.3593	0.8545±0.3423
2	1.3031±0.2649	1.3324±0.2914	1.3286±0.2878	0.7437±0.3444	0.7544±0.3337
1	1.1376±0.1427	1.1636±0.1291	1.152±0.1328	0.6715±0.214	0.6198±0.2132

Table A.72: Average Euclidean distance results ± standard deviation for the German tense dataset with respect to $\mathcal{P}_2$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.6116±0.0641	0.6054±0.0649	0.6087±0.0654	0.9964±0.013	0.9963±0.0122
11	0.6011±0.0952	0.6008±0.099	0.6012±0.0994	0.9976±0.0197	0.9977±0.0192
10	0.7389±0.076	0.7382±0.0832	0.7378±0.0846	0.9976±0.0171	0.9978±0.0168
9	0.5545±0.1234	0.5574±0.1221	0.5538±0.1243	0.9981±0.0115	0.9982±0.0113
8	0.8369±0.0566	0.8367±0.059	0.8364±0.0595	0.9993±0.0053	0.9993±0.0055
7	0.7726±0.0421	0.772±0.0439	0.7744±0.0432	0.9986±0.0074	0.9986±0.0078
6	0.6695±0.0734	0.6698±0.0738	0.6692±0.0734	0.9988±0.0033	0.9988±0.0036
5	0.6489±0.0382	0.6515±0.0375	0.6491±0.0375	0.9996±0.0007	0.9996±0.0008
4	0.7198±0.0504	0.7197±0.0504	0.7197±0.0504	1.0±0.0	1.0±0.0
3	0.4412±0.1247	0.4412±0.1247	0.4413±0.1247	1.0±0.0	1.0±0.0
2	0.6635±0.0688	0.6633±0.0688	0.6634±0.0688	1.0±0.0	1.0±0.0
1	0.1485±0.0269	0.1486±0.0268	0.1484±0.0268	0.9999±0.0	0.9999±0.0

Table A.73: Average cosine similarity results ± standard deviation for the German tense dataset with respect to $\mathcal{P}_1$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.8687±0.0251	0.8662±0.0244	0.8684±0.0241	0.9964±0.013	0.9963±0.0122
11	0.9864±0.0233	0.9869±0.0219	0.9875±0.0192	0.9976±0.0197	0.9977±0.0192
10	0.9975±0.017	0.9978±0.016	0.9982±0.0137	0.9976±0.0171	0.9978±0.0168
9	0.998±0.0112	0.9985±0.0084	0.9988±0.0065	0.9981±0.0115	0.9982±0.0113
8	0.9196±0.0234	0.9203±0.0209	0.9204±0.0204	0.9993±0.0053	0.9993±0.0055
7	0.9291±0.0258	0.9301±0.021	0.9322±0.0181	0.9986±0.0074	0.9986±0.0078
6	0.9993±0.002	0.9992±0.0023	0.9992±0.0022	0.9988±0.0033	0.9988±0.0036
5	0.8851±0.0138	0.8864±0.0134	0.8855±0.0133	0.9996±0.0007	0.9996±0.0008
4	0.9136±0.0066	0.9136±0.0066	0.9136±0.0066	1.0±0.0	1.0±0.0
3	1.0±0.0	1.0±0.0	1.0±0.0	1.0±0.0	1.0±0.0
2	0.9997±0.0003	0.9997±0.0003	0.9997±0.0003	1.0±0.0	1.0±0.0
1	0.9995±0.0003	0.9995±0.0003	0.9995±0.0003	0.9999±0.0	0.9999±0.0

Table A.74: Average cosine similarity results ± standard deviation for the German tense dataset with respect to $\mathcal{P}_2$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	11.4892±0.2696	11.4253±0.269	11.4062±0.2696	0.8608±0.6957	0.8954±0.6588
11	20.8563±0.3716	20.8597±0.3857	20.8599±0.3963	0.4556±1.4887	0.455±1.444
10	22.3942±0.5904	22.3999±0.616	22.41±0.6287	0.6895±1.5731	0.6984±1.5156
9	23.1529±0.4976	23.1487±0.4894	23.1612±0.4937	0.8206±1.2746	0.8401±1.2247
8	19.9413±0.4509	19.9416±0.4548	19.9433±0.455	0.3404±0.7864	0.344±0.7667
7	17.6541±0.5347	17.6551±0.5354	17.639±0.5311	0.5368±1.0469	0.5233±1.0475
6	18.7646±0.2594	18.7634±0.2597	18.7551±0.2581	0.6983±0.7545	0.6653±0.7931
5	18.2657±0.2636	18.2628±0.2638	18.2648±0.2624	0.4848±0.3531	0.4578±0.3587
4	16.2968±0.3238	16.2971±0.3238	16.2972±0.3238	0.0391±0.0084	0.0404±0.0085
3	19.0841±0.3949	19.0843±0.3949	19.0842±0.3949	0.0449±0.0119	0.0463±0.0118
2	19.2335±0.2764	19.234±0.2764	19.2337±0.2764	0.0578±0.016	0.059±0.0159
1	14.5102±0.1108	14.5094±0.1103	14.5121±0.1105	0.1539±0.0444	0.1569±0.0443

Table A.75: Average Euclidean distance results ± standard deviation for the German tense dataset with respect to $\mathcal{P}_1$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	10.1257±0.2935	10.0543±0.285	10.0351±0.2912	0.8608±0.6957	0.8954±0.6588
11	13.0285±0.9639	13.0165±0.9486	13.0072±0.9297	0.4556±1.4887	0.455±1.444
10	2.5564±2.5591	2.5051±2.5237	2.4995±2.4255	0.6895±1.5731	0.6984±1.5156
9	1.3446±1.216	1.3028±1.0355	1.2959±0.8558	0.8206±1.2746	0.8401±1.2247
8	18.9084±0.4635	18.9042±0.458	18.9043±0.4581	0.3404±0.7864	0.344±0.7667
7	14.7117±0.5794	14.7007±0.5332	14.68±0.5124	0.5368±1.0469	0.5233±1.0475
6	0.6061±0.5203	0.6518±0.5621	0.6296±0.5666	0.6983±0.7545	0.6653±0.7931
5	16.5509±0.257	16.5521±0.2587	16.548±0.2599	0.4848±0.3531	0.4578±0.3587
4	13.8634±0.1774	13.8637±0.1774	13.8638±0.1774	0.0391±0.0084	0.0404±0.0085
3	0.1654±0.0297	0.1669±0.03	0.1667±0.0301	0.0449±0.0119	0.0463±0.0118
2	1.4709±0.8886	1.4725±0.8878	1.4722±0.8877	0.0578±0.016	0.059±0.0159
1	0.5038±0.1723	0.5087±0.1729	0.5086±0.1726	0.1539±0.0444	0.1569±0.0443

Table A.76: Average Euclidean distance results ± standard deviation for the German tense dataset with respect to $\mathcal{P}_2$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.9318±0.0258	0.8787±0.0371	0.8838±0.0337	0.895±0.0474	0.9036±0.0422
11	0.9204±0.0234	0.8744±0.0383	0.8821±0.0325	0.9268±0.035	0.9349±0.0304
10	0.7773±0.0406	0.7299±0.0534	0.7408±0.0482	0.9066±0.0447	0.9197±0.0382
9	0.8782±0.0253	0.8406±0.036	0.8487±0.0332	0.9333±0.0353	0.9427±0.0315
8	0.9646±0.0125	0.9355±0.0231	0.9424±0.0194	0.9485±0.0289	0.9572±0.0251
7	0.8762±0.0237	0.8583±0.028	0.8656±0.0254	0.9601±0.0235	0.9666±0.0203
6	0.9804±0.0074	0.9661±0.0105	0.9693±0.0095	0.9752±0.0133	0.9793±0.0117
5	0.8084±0.0292	0.8039±0.0282	0.8037±0.0286	0.9876±0.0053	0.9885±0.0057
4	0.9969±0.002	0.9955±0.0022	0.9954±0.0027	0.9972±0.0014	0.9972±0.0023
3	0.9354±0.0079	0.9332±0.0079	0.9327±0.0078	0.9978±0.0011	0.9978±0.0011
2	0.929±0.0069	0.9275±0.007	0.9275±0.0069	0.9986±0.0007	0.9986±0.0008
1	0.999±0.0003	0.9985±0.0004	0.9984±0.0004	0.999±0.0007	0.999±0.0006

Table A.77: Average cosine similarity results ± standard deviation for the German agreement dataset with respect to $\mathcal{P}_1$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.9248±0.0258	0.8814±0.0324	0.8856±0.03	0.895±0.0474	0.9036±0.0422
11	0.9501±0.0172	0.9177±0.0243	0.9227±0.0212	0.9268±0.035	0.9349±0.0304
10	0.9375±0.0207	0.8907±0.0337	0.9007±0.0282	0.9066±0.0447	0.9197±0.0382
9	0.9536±0.0159	0.9194±0.0266	0.9261±0.0236	0.9333±0.0353	0.9427±0.0315
8	0.9527±0.0142	0.9331±0.0195	0.9385±0.0171	0.9485±0.0289	0.9572±0.0251
7	0.9519±0.0147	0.9451±0.0146	0.9476±0.0148	0.9601±0.0235	0.9666±0.0203
6	0.9719±0.009	0.9638±0.009	0.9659±0.0088	0.9752±0.0133	0.9793±0.0117
5	0.9869±0.005	0.9827±0.0045	0.9829±0.0053	0.9876±0.0053	0.9885±0.0057
4	0.9958±0.002	0.9954±0.002	0.9952±0.0026	0.9972±0.0014	0.9972±0.0023
3	0.9969±0.0008	0.9965±0.0008	0.9963±0.0009	0.9978±0.0011	0.9978±0.0011
2	0.9973±0.0011	0.9972±0.0011	0.997±0.0012	0.9986±0.0007	0.9986±0.0008
1	0.9983±0.0005	0.9982±0.0005	0.9981±0.0005	0.999±0.0007	0.999±0.0006

Table A.78: Average cosine similarity results ± standard deviation for the German agreement dataset with respect to $\mathcal{P}_2$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	7.1467±1.2801	9.4443±1.4381	9.2809±1.3426	8.8372±2.1063	8.4829±1.9814
11	11.7134±1.4301	13.6639±1.7347	13.3724±1.5474	10.1384±2.4916	9.5679±2.3008
10	14.2794±1.0476	15.4581±1.2171	15.1919±1.1077	9.4834±2.3877	8.7955±2.168
9	10.5171±0.9584	11.7576±1.1383	11.5145±1.0757	7.6282±2.0756	7.0659±1.9499
8	5.7252±0.9455	7.6877±1.3478	7.2959±1.1914	6.7716±1.9056	6.1754±1.7548
7	10.7278±0.8634	11.329±0.9081	11.1405±0.8671	5.9089±1.6541	5.4175±1.5234
6	4.3414±0.7218	5.7037±0.8576	5.4408±0.795	4.8051±1.2255	4.3896±1.1252
5	12.7843±0.7594	12.8906±0.7314	12.8863±0.7438	3.289±0.699	3.1609±0.6976
4	1.6235±0.3065	1.9426±0.3124	1.9487±0.3679	1.4984±0.3745	1.4783±0.4272
3	7.0266±0.3094	7.0551±0.3084	7.073±0.3048	1.0135±0.2468	1.02±0.2377
2	8.9116±0.2886	8.943±0.2881	8.9461±0.286	0.9254±0.2348	0.9259±0.2364
1	0.8904±0.1208	1.087±0.1408	1.1003±0.1474	0.8289±0.2748	0.8231±0.2549

Table A.79: Average Euclidean distance results ± standard deviation for the German agreement dataset with respect to $\mathcal{P}_1$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	7.4671±1.2514	9.3278±1.2924	9.1903±1.2255	8.8372±2.1063	8.4829±1.9814
11	8.445±1.3335	10.7883±1.567	10.4801±1.3804	10.1384±2.4916	9.5679±2.3008
10	7.7517±1.224	10.2243±1.5956	9.7631±1.3742	9.4834±2.3877	8.7955±2.168
9	6.4114±1.0485	8.4182±1.393	8.0791±1.2729	7.6282±2.0756	7.0659±1.9499
8	6.5949±0.9572	7.8313±1.134	7.5298±1.0321	6.7716±1.9056	6.1754±1.7548
7	6.5881±0.9585	7.0572±0.9097	6.9163±0.9325	5.9089±1.6541	5.4175±1.5234
6	5.1959±0.7624	5.9031±0.7124	5.7339±0.7029	4.8051±1.2255	4.3896±1.1252
5	3.4039±0.5503	3.9287±0.4753	3.9023±0.5164	3.289±0.699	3.1609±0.6976
4	1.867±0.3351	1.9648±0.3231	1.983±0.3702	1.4984±0.3745	1.4783±0.4272
3	1.2249±0.1477	1.3111±0.1359	1.3411±0.1461	1.0135±0.2468	1.02±0.2377
2	1.4557±0.32	1.4786±0.3198	1.513±0.3242	0.9254±0.2348	0.9259±0.2364
1	1.1264±0.1763	1.1819±0.1638	1.1961±0.162	0.8289±0.2748	0.8231±0.2549

Table A.80: Average Euclidean distance results ± standard deviation for the German agreement dataset with respect to $\mathcal{P}_2$ and BERT. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.5824±0.0394	0.5798±0.0392	0.5785±0.0399	0.9968±0.0062	0.9972±0.0059
11	0.8137±0.0566	0.8132±0.0575	0.8134±0.0574	0.999±0.0089	0.9992±0.0083
10	0.3834±0.1466	0.3831±0.1463	0.3836±0.1467	0.999±0.0067	0.999±0.008
9	0.7205±0.1224	0.7199±0.1224	0.7211±0.1217	0.9991±0.0035	0.9991±0.0051
8	0.6051±0.1626	0.6051±0.1626	0.6049±0.1629	0.9999±0.0007	0.9998±0.0014
7	0.0566±0.0442	0.0566±0.0444	0.0563±0.0447	0.9998±0.0012	0.9996±0.0024
6	0.6814±0.0884	0.6816±0.0886	0.6817±0.0881	0.9997±0.0015	0.9995±0.002
5	0.0609±0.0393	0.0609±0.0393	0.061±0.0394	0.9998±0.0006	0.9998±0.0006
4	0.5531±0.0607	0.5531±0.0607	0.5531±0.0607	1.0±0.0	1.0±0.0
3	0.491±0.1588	0.491±0.1589	0.491±0.1589	1.0±0.0	1.0±0.0
2	0.6508±0.0911	0.6509±0.0911	0.6508±0.0911	1.0±0.0	1.0±0.0
1	0.7877±0.0326	0.7871±0.0327	0.787±0.0327	0.9999±0.0001	0.9999±0.0001

Table A.81: Average cosine similarity results ± standard deviation for the German agreement dataset with respect to $\mathcal{P}_1$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	0.9957±0.0135	0.9948±0.0137	0.9949±0.015	0.9968±0.0062	0.9972±0.0059
11	0.8831±0.0438	0.8827±0.0446	0.8828±0.0453	0.999±0.0089	0.9992±0.0083
10	0.998±0.0157	0.9977±0.0165	0.9976±0.0183	0.999±0.0067	0.999±0.008
9	0.9973±0.0101	0.9971±0.0109	0.9969±0.0127	0.9991±0.0035	0.9991±0.0051
8	0.9925±0.0061	0.9924±0.0064	0.9923±0.0073	0.9999±0.0007	0.9998±0.0014
7	0.9484±0.0229	0.9483±0.0233	0.947±0.0253	0.9998±0.0012	0.9996±0.0024
6	0.9965±0.0065	0.9965±0.0065	0.9958±0.0077	0.9997±0.0015	0.9995±0.002
5	0.9993±0.0015	0.9994±0.0015	0.9991±0.0019	0.9998±0.0006	0.9998±0.0006
4	1.0±0.0	1.0±0.0	1.0±0.0	1.0±0.0	1.0±0.0
3	1.0±0.0	1.0±0.0	1.0±0.0	1.0±0.0	1.0±0.0
2	0.9911±0.005	0.9911±0.005	0.9911±0.005	1.0±0.0	1.0±0.0
1	0.8911±0.0106	0.8907±0.0107	0.8907±0.0106	0.9999±0.0001	0.9999±0.0001

Table A.82: Average cosine similarity results ± standard deviation for the German agreement dataset with respect to $\mathcal{P}_2$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	11.6617±0.1955	11.6864±0.1913	11.7035±0.1954	0.9264±0.4942	0.8694±0.4498
11	19.5109±0.5276	19.5155±0.532	19.5129±0.5297	0.3852±0.915	0.3444±0.8348
10	24.3291±0.4868	24.3292±0.4857	24.3246±0.4859	0.5917±0.9682	0.543±0.9547
9	22.4855±0.8058	22.4869±0.8061	22.4837±0.8037	0.6868±0.8034	0.65±0.8295
8	21.4986±0.4194	21.4989±0.4197	21.4994±0.4207	0.1976±0.2812	0.2121±0.3842
7	22.7263±0.3367	22.7267±0.3402	22.7273±0.3418	0.2165±0.3553	0.2661±0.544
6	18.6926±0.5226	18.6919±0.5247	18.699±0.5253	0.3202±0.398	0.3779±0.5314
5	21.6945±0.3024	21.6964±0.3017	21.7024±0.3019	0.2852±0.2685	0.3204±0.3208
4	17.4336±0.3266	17.4335±0.3266	17.4336±0.3266	0.0443±0.0151	0.045±0.0166
3	18.8937±0.4819	18.8935±0.4819	18.8935±0.4819	0.0552±0.0198	0.0564±0.0211
2	19.1768±0.389	19.1768±0.389	19.1769±0.3889	0.0723±0.0262	0.0752±0.0284
1	11.5076±0.2969	11.509±0.2969	11.5101±0.2969	0.2006±0.0691	0.2037±0.0724

Table A.83: Average Euclidean distance results ± standard deviation for the German agreement dataset with respect to $\mathcal{P}_1$ and RoBERTa. Each row represents a different layer starting from the last one (12).

	$\hat{y}$-y	$\hat{y}$-d_1	$\hat{y}$-d_2	y-d_1	y-d_2
12	1.3383±0.6582	1.4348±0.6484	1.4233±0.685	0.9264±0.4942	0.8694±0.4498
11	19.0459±0.4252	19.0485±0.4325	19.0464±0.4325	0.3852±0.915	0.3444±0.8348
10	1.7667±1.5502	1.8008±1.6099	1.8024±1.6785	0.5917±0.9682	0.543±0.9547
9	8.2874±2.9032	8.3011±2.8988	8.3046±2.9075	0.6868±0.8034	0.65±0.8295
8	12.6453±0.7405	12.646±0.7428	12.6494±0.7469	0.1976±0.2812	0.2121±0.3842
7	12.8754±0.7878	12.8754±0.7908	12.8948±0.8087	0.2165±0.3553	0.2661±0.544
6	5.189±0.6354	5.1866±0.6384	5.2308±0.6878	0.3202±0.398	0.3779±0.5314
5	0.6136±0.4849	0.6041±0.4833	0.6785±0.5768	0.2852±0.2685	0.3204±0.3208
4	0.1292±0.0231	0.1305±0.0226	0.1302±0.0221	0.0443±0.0151	0.045±0.0166
3	0.1469±0.0257	0.1495±0.0257	0.1494±0.0262	0.0552±0.0198	0.0564±0.0211
2	6.8155±1.4536	6.8151±1.4536	6.8154±1.4536	0.0723±0.0262	0.0752±0.0284
1	10.3082±0.1776	10.3089±0.178	10.3101±0.1776	0.2006±0.0691	0.2037±0.0724

Table A.84: Average Euclidean distance results ± standard deviation for the German agreement dataset with respect to $\mathcal{P}_2$ and RoBERTa. Each row represents a different layer starting from the last one (12).

Bibliography

Altszyler, E., Sigman, M., and Slezak, D. F. (2016). Comparative study of LSA vs word2vec embeddings in small corpora: a case study in dreams database. *arXiv preprint arXiv:1610.01520*.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR*, San Diego, CA, USA.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.

Bastings, J., Belinkov, Y., Elazar, Y., Hupkes, D., Saphra, N., and Wiegreffe, S., editors (2022). *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Belinkov, Y. (2022). Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Belinkov, Y. and Glass, J. (2019). Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

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

Boleda, G. (2020). Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6(1):213–234.

Boleda, G. and Herbelot, A. (2016). Formal distributional semantics: Introduction to the special issue. *Computational Linguistics*, 42(4):619–635.

Camacho-Collados, J. and Pilehvar, M. T. (2018). From word to sense embeddings: A survey on vector representations of meaning. *J. Artif. Int. Res.*, 63(1):743–788.

Çöltekin, c. and Rama, T. (2023). What do complexity measures measure? correlating and validating corpus-based measures of morphological complexity. *Linguistics Vanguard*, 9(s1):27–43.

Christ, O. (1994). A modular and flexible architecture for an integrated corpus query system. In *Proceedings of COMPLEX'94: 3rd Conference on Computational Lexicography and Text Research*, pages 23–32, Budapest, Hungary. tt cmp-lg: tt 9408005.

Clark, S. (2015). *Vector Space Models of Lexical Meaning*, chapter 16, pages 493–522. John Wiley & Sons, Ltd.

Clinciu, M.-A. and Hastie, H. (2019). A survey of explainable AI terminology. In *Proceedings of the 1st Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence (NL4XAI 2019)*, pages 8–13. Association for Computational Linguistics.

Confalonieri, R., Coba, L., Wagner, B., and Besold, T. R. (2021). A historical perspective of explainable artificial intelligence. *WIREs Data Mining and Knowledge Discovery*, 11(1):e1391.

Covington, M. A. and McFall, J. D. (2010). Cutting the gordian knot: The moving-average type–token ratio (mattr). *Journal of Quantitative Linguistics*, 17(2):94–100.

Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., and Sen, P. (2020). A survey of the state of explainable AI for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Suzhou, China. Association for Computational Linguistics.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dhillon, I. S. and Modha, D. S. (2001). Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1–2):143–175.

Diao, S. (2023). mlconjug3. `https://github.com/Ars-Linguistica/mlconjug3`.

Elazar, Y., Ravfogel, S., Jacovi, A., and Goldberg, Y. (2021). Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Firth, J. R. (1957). A synopsis of linguistic theory 1930-55. 1952-59:1–32.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89, Turin, Italy.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

Graziani, M., Dutkiewicz, L., Calvaresi, D., Amorim, J. P., Yordanova, K., Vered, M., Nair, R., Abreu, P. H., Blanke, T., Pulignano, V., Prior, J. O., Lauwaert, L., Reijers, W., Depeursinge, A., Andrearczyk, V., and Müller, H. (2023). A global taxonomy of interpretable ai: unifying the terminology for the technical and social sciences. *Artificial Intelligence Review*, 56:3473–3504.

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., and Baroni, M. (2018). Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Günther, F., Rinaldi, L., and Marelli, M. (2019). Vector-space models of semantic representation from a cognitive perspective: A discussion of common misconceptions. *Perspectives on Psychological Science*, 14(6):1006–1033. PMID: 31505121.

Harris, Z. S. (1954). Distributional structure. *WORD*, 10(2-3):146–162.

Hewitt, J. and Liang, P. (2019). Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

Jacovi, A. and Goldberg, Y. (2020). Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Jakubíček, M., Kilgarriff, A., McCarthy, D., and Rychlý, P. (2010). Fast syntactic searching in very large corpora for many languages. In *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*, pages 741–747, Tohoku University, Sendai, Japan. Institute of Digital Enhancement of Cognitive Processing, Waseda University.

Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., and Suchomel, V. (2013). The tenten corpus family. In *7th international corpus linguistics conference CL*, pages 125–127.

Jurafsky, D. and Martin., J. H. (2023). *Transformers and Pretrained Language Models*, chapter 10. Web publication (draft) at https://web.stanford.edu/ jurafsky/slp3/.

Kanerva, P., Kristofersson, J., and Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, volume 1036.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models.

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., and Suchomel, V. (2014). The sketch engine: ten years on. *Lexicography*, 1:7–36.

Kilgarriff, A., Rychlý, P., Smrž, P., and Tugwell, D. (2004). The sketch engine. In *Proceedings of the 11th EURALEX International Congress*, pages 105–116. Université de Bretagne-Sud, Faculté des lettres et des sciences humaines.

Lakretz, Y., Kruszewski, G., Desbordes, T., Hupkes, D., Dehaene, S., and Baroni, M. (2019). The emergence of number and syntax units in LSTM language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 11–20, Minneapolis, Minnesota. Association for Computational Linguistics.

Landauer, T. K. and Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

Lasri, K., Pimentel, T., Lenci, A., Poibeau, T., and Cotterell, R. (2022). Probing for the usage of grammatical number. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8818–8831, Dublin, Ireland. Association for Computational Linguistics.

Lenci, A. and Littell, J. S. (2008). Distributional semantics in linguistic and cognitive research. *The Italian Journal of Linguistics*, 20:1–32.

Lenci, A., Sahlgren, M., Jeuniaux, P., Gyllensten, A. C., and Miliani, M. (2022). A comparative evaluation and analysis of three generations of distributional semantic models. *Language resources and evaluation*, 56(4):1269–1313.

Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, volume 27, pages 2177—2185. Curran Associates, Inc.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., and Stoica, I. (2018). Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*.

Linardatos, P., Papastefanopoulos, V., and Kotsiantis, S. (2021). Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1).

Linzen, T., Dupoux, E., and Goldberg, Y. (2016). Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Liu, Q., Kusner, M. J., and Blunsom, P. (2020). A survey on contextual embeddings. *arXiv preprint arXiv:2003.07278*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Madsen, A., Reddy, S., and Chandar, S. (2022). Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys*, *55*(8):1–42.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Scoring, term weighting and the vector space model*, chapter 6. Cambridge University Press. Web publication at http://informationretrieval.org/.

Marcinkevičs, R. and Vogt, J. E. (2020). Interpretability and explainability: A machine learning zoo mini-tour. *arXiv preprint arXiv:2012.01805*.

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

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In Bengio, Y. and LeCun, Y., editors, *1st International Conference on Learning Representations, ICLR*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 26, pages 3111–3119.

Mikolov, T., tau Yih, W., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Minixhofer, B., Paischer, F., and Rekabsaz, N. (2022). WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.

Nicolai, G., Cherry, C., and Kondrak, G. (2015). Morpho-syntactic regularities in continuous word representations: A multilingual study. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 129–134, Denver, Colorado. Association for Computational Linguistics.

Osgoodand, C. E., Suci, G. J., and Tannenbaum, P. H. (1957). *The Measurement of meaning*. University of Illinois Press.

Padró, L. and Stanilovsky, E. (2012). Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey. ELRA.

Parisi, L., Francia, S., and Magnani, P. (2020). Umberto: an italian language model trained with whole word masking. `https://github.com/musixmatchresearch/umberto`.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2020). Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

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

Sahlgren, M. (2005). An introduction to random indexing. In *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering*.

Sahlgren, M. and Lenci, A. (2016). The effects of data size and frequency range on distributional semantic models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 975–980, Austin, Texas. Association for Computational Linguistics.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK.

Schmid, H. and Laws, F. (2008). Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 777–784, Manchester, UK. Coling 2008 Organizing Committee.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. (2020). Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

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

Tjoa, E. and Guan, C. (2021). A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE transactions on neural networks and learning systems*, 32(11):4793–4813.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

van der Maaten, L., Postma, E., and van den Herik, J. (2009). Dimensionality reduction: A comparative review. Technical Report TiCC-TR 2009-005, Tilburg centre for Creative Computing.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Westera, M. and Boleda, G. (2019). Don't blame distributional semantics if it can't do entailment. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 120–133, Gothenburg, Sweden. Association for Computational Linguistics.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 5754–5764.