An Analysis on How Pre-Trained Language Models Learn Different Aspects

Ejdis Gjinika[1,*], Nicola Arici[1,*], Luca Putelli[1], Alfonso E. Gerevini[1] and Ivan Serina[1]

Università degli Studi di Brescia, Via Branze 38, Brescia (IT)

Abstract

By now, it is widely known that pre-trained Neural Language Models (NLM) and Large Language Models (LLM) possess remarkable capabilities and they are able to solve many Natural Language Processing Tasks. However, not as much is understood regarding *how* Transformer-based models acquire this ability during their complex training process. In this context, an interesting line of work surfaced in the last few years: the study of the so-called *learning trajectories*. Several studies tested the knowledge acquired by a model not only when it was fully trained, but also in its checkpoints, i.e. intermediate versions of the model at different stages during its training. Nonetheless, most of these works focused on simple tasks, often analysing single grammatical aspects (such as part-of-speech tags, transitive verbs, etc.) without a proper comparison with more complex tasks and with semantics-based aspects. In this paper, we consider two additional tasks to study the learning trajectory of NLMs and to compare different aspects. The first one consists on classifying a sentence as correct or wrong, from the grammatical point of view, from a novel dataset which can contain several types of errors. The second one is a totally semantic-based task revolving understanding whether a sentence is funny or not. In our experimental evaluation, we compare the learning trajectories on these two tasks with three simpler grammatical aspects. Thus, we highlight the most important similarities and divergences in terms of how these types of knowledge are learned by three GPT-NeoX models. Moreover, we analyse the behaviour of each layers of the models considered, verifying whether there are significant differences among them.

Keywords

Natural Language Processing, Explainability, Interpretability, Learning Trajectory

1. Introduction

With the rise of the Transformer architecture [2] and the release of the first Large Language Models (LLMs), the race to create the biggest, the most powerful and the most accurate LLM began. New generative capabilities, such as few-shot learning, have been explored and new state-of-the-art results have been obtained in many NLP tasks [3].

Currently, we can use a countless variety of models, spanning from the smallest ones, which can efficiently and swiftly tackle basic tasks [4], to the larger models that can handle multiple intricate tasks with excellent performance. However, our comprehension of the language understanding mechanisms behind these models, assuming that they understand [5], is limited, as well as the mechanisms behind their predictions. In recent years, different research has started to focus on the interpretability of Neural Language Models (NLMs) [6] from different angles: by analyzing self-attention weights to find relations among words [7], by determining whether NLMs have acquired specific world knowledge [8], or by investigating their linguistic capabilities [9].

A widely widespread approach for evaluating the capabilities of a NLM is through one or more probing tasks, i.e. training a simple classifier to verify if a particular language property is contained in the embedded representation of words and sentences calculated by the model [10, 11]. This technique obtains good results and valuable insights, showing how NLMs possess knowledge related to syntax and

XAI.it - 5th Italian Workshop on Explainable Artificial Intelligence, co-located with the 23rd International Conference of the Italian Association for Artificial Intelligence, Bolzano, Italy, November 25-28, 2024 [1]

*Corresponding author.

e.gjinika@studenti.unibs.it (E. Gjinika); nicola.arici@unibs.it (N. Arici); luca.putelli@unibs.it (L. Putelli); alfonso.gerevini@unibs.it (A. E. Gerevini); ivan.serina@unibs.it (I. Serina)

0009-0000-9713-6630 (N. Arici); 0009-0008-5055-6812 (L. Putelli); 0000-0001-9008-6386 (A. E. Gerevini); 0000-0002-7785-9492 (I. Serina)

grammar [9], temporal relations [12], and more semantic related aspects like the presence of metaphors [13].

However, one flaw of these works is that they mostly take into account single aspects of the language (such as syntactic correctness, parts of speech, etc.) or specific models, such as BERT [14], without creating a proper comparison between different tasks or models. Moreover, most of these studies evaluated the performance of probing tasks simply in terms of accuracy, a technique that has been demonstrated to have some shortcomings [15, 16]. Moreover, it would be quite interesting not only to understand the capabilities of a fully-trained NLM (as it has been done by the aforementioned works), but also to investigate how these capabilities are achieved and how these concepts are acquired by a NLM during their training procedure.

In this work, we put to the test and compare different size NLMs and their capabilities across five different language properties, ranging from grammar, (with classical tasks involving transitive verbs, passive forms and concordance among verbs and nouns) to semantics, such as asking the model to detect simple humorous sentences. Moreover, we introduce a task into which the probe consists on classifying a sentence as correct or wrong, from the grammatical point of view, from a novel dataset which "mixes" several types of grammatical errors.

To evaluate the performance of NLMs in these tasks, we implement the state-of-the-art Minimum Description Length (MDL) method [17]. Another important aspect of our work is that we are interested in *when* and *how* a property is learned by a model. Following a relatively new way of analyzing NLMs [18, 19, 20], we decided to evaluate the so-called *learning trajectories* of the models, by studying their performance (using our probing tasks) across their training. In order to do that, we execute our probing tasks in different checkpoints (i.e. points during training), acquiring information regarding how quickly a property is learned, when it reaches its best performance, and how it evolves across time. Finally, we are interested in "where" these language properties are encoded by NLMs. Therefore, we perform an in-depth analysis of the learning trajectories of different layers, evaluating their similarities and differences.

The remainder of the paper is organized as follows. In Section 2, we describe the related work; in Section 3 we present our probing tasks and the datasets we exploited; in Section 4 we explain the methodology we followed; in Section 5 we show the experimental results obtained and finally, we draw the conclusions and possible future works.

2. Related Work

The explanation of how Neural Language Models work and their knowledge has been the subject of many works, following different points of view and methodologies [21, 22]. First, several white-box approaches studied the self-attention weights of the model's heads analysing whether they encode meaningful relationships among words [7]. This line of work includes visualization techniques [23], clustering [24] and categorization [25, 26]. Second, an interesting analysis has been conducted on whether these models learned specific world knowledge in subjects like history, geography, etc. [8], introducing benchmarks [27] and standard tests [28]. However, one of the most important lines of work focused on syntactic and grammatical capabilities using probing tasks [11].

A probe is a small feed-forward neural network that receives in input the embedded word (or sentence) representations, generated by a NLM, and it is trained to solve a specific supervised task [29] such, for instance, whether a sentence contains a specific grammatical error or not. Probing has been exploited to study different forms of syntactic properties [9, 30], whether the NLMs encode some forms of dependency parsing [31] and temporal relations [12]. An interesting work more concerned with semantic-based aspects is the one in [13], which focuses on how metaphors are recognised by several pre-trained Neural Language Models across different datasets and languages. Another interesting application is presented in [32, 33], into which the authors exploit probing tasks in order to visualize gender bias in BERT word representations.

Although in most studies probing tasks have been only applied to fully trained NLMs [9, 12, 30],

another line of work exploited them in order to understand *how* and *when* such models acquire these capabilities in their training process. More specifically, Saphra and Lopez [18] analysed a LSTM-based language model and discovered that several syntactic features (such as parts of speech) are learned in the first stages of the training, whereas learning most complex aspects (such as topic-related knowledge) need more training steps. The work in [19] obtained similar results for the AlBERT model [34]. They also performed a comparison among the learning processes of grammar and basic semantics (in particular, coreference and semantic role labeling) and reported that they are very similar. In fact, these types of knowledge are learned quite early in the training and they do not improve after the first steps. Focusing on factual knowledge and common sense, the work in [20] analysed RoBERTa in terms of its learning trajectories and found that this type of information is learned more in depth as the training progresses.

In this paper, we perform a similar analysis. However, we compare simple grammar tasks with a more refined grammar test, which combines several types of possible errors, and with a semantic-only case study which is understanding whether a sentence contains some humorous content.

3. Case Studies and Datasets

The case studies approached in this work regard syntax, grammar and a purely semantic task. More specifically, we created five different tasks, called *Causative*, *Coordinate structures*, *Passive*, *Mix* and *Humor* and we collected the respective datasets. Apart from the last one, our datasets were taken and adapted from the BLiMP benchmark [35]. Please note that the *Mix* dataset includes four different types of grammar concepts assembled in a single task. In the Table 1 we show a positive and negative sentence for each task we considered.

In the following, we explain more in-depth each task and the respective dataset.

Causative dataset This dataset is made by sentences that may contain syntactic errors in terms of verb-object concordance. In particular, some verbs may be intransitive and therefore their use associated with an object complement leads to a syntactic error. This dataset consists of 2000 sentences and the labels are equally distributed (1000 correct and 1000 wrong).

Coordinate structures dataset This dataset is made by sentences that may contain structure errors, undermining the coherence of the sentence. Although the sentence structure comprehends several

Task		Positive	Negative
Causative		Chad slows the cart.	Chad **disappears** the cart.
Coordinate Structures		What paintings has Melissa looked like and Erin revealed?	What has Melissa looked like **paintings** and Erin revealed?
Passive		Deborah isn't disturbed by this hospital.	Deborah isn't **laughed** by this hospital.
Mix	Determiners-noun	Winston Churchill boycotts these public parks.	Winston Churchill boycotts these public **park**.
	Verbs-noun	A report about the Impressionists confuses Stacey.	A report about the Impressionists **confuse** Stacey.
	Irregular plural subject-verb	That mouse hasn't distracted a patient.	That mouse **haven't** distracted a patient.
	Regular plural subject-verb	That teacher tours the hills.	That teacher **tour** the hills.
Humor		Is it OK to use my AM radio after noon?	Britain could use sanctions to pressure Maldives government.

Table 1
Examples of the sentences contained in each dataset. For the first four tasks, the column *Positive* contains the correct sentences and the column *Negative* contains the wrong ones. For *Mix*, we show the four different type of errors contained in the dataset, with a positive and negative example each. The mistaken word is marked in bold. For the *Humor* task, the Positive column refers to the humorous sentence, whereas the Negative refers to the not humorous one.

components, all sentences in this data contain at most one error. This dataset contains 4000 sentences, equally distributed between correct and wrong.

Passive dataset This dataset is made by sentences that may contain an error related to the use of verbs in their (possibly not existent) passive form, leading to compatibility errors between verb and subject. Therefore, the goal of the Passive task is to identify whether a verb supports or not the passive form. This dataset contains 4000 sentences with the labels equally distributed.

Mix dataset This dataset has been made specifically for this paper and combines four different types of grammatical errors together, taken from the BLiMP benchmark [35]. Specifically, we consider (i) sentences that may contain errors in the use of determiners paired with nouns (*determiners-noun agreement*); (ii) sentences into which the verbs may not agree with nouns (*verbs-noun agreement*); (iii) sentence into which verbs that may not agree with subjects with irregular plurals (*irregular plural subject-verb agreement*); (iv) sentence which may contain errors in the agreement between subjects and verbs with regular plurals (*regular plural subject-verb agreement*). The goal of this dataset is to test the capability of a NLM to simply identify whether a sentence contains a grammatical error *in general*, without a strong regularity among the positive and negative examples provided to the probe. Therefore, the overall task should be more challenging. This dataset has been built with 4000 instances, with equally distributed labels. The error types are sampled randomly.

Humor dataset This datasets consists of sentences that contain simple forms of humor (such as puns and jokes), and sentences that were taken from titles of newspaper articles of extracted from Wikipedia and therefore with no humorous content. This task is purely semantic, and all sentences are grammatically and syntactically correct. This dataset is taken from a from a Kaggle competition of humor detection and contains 4000 sentences.[1] The labels are equally distributed between humorous or not.

4. Methodology

In this section, we describe the overall procedure we employed to calculate and analyse the learning trajectories of NLMs for the considered tasks.

4.1. Structuring and Evaluating the Probing Tasks

Each of our probe consists of a feed-forward neural network which has the goal to learn, starting from the embedded representations provided by a pre-trained NLM, one of the tasks described in Section 3. All tasks we considered are binary, i.e. they have only two classes. For the grammatical ones (*Mix* dataset included), we assign label 1 to the correct sentences and 0 to the wrong ones. Similarly, for the Humor task we assign label 1 to those sentences which contain some forms of humor and 0 to those which do not.

The probing is designed as follows. Considering a task, its dataset and a NLM, the probe receives in input the N-dimensional embedded representation of a sentence s and has to correctly classify s. The embedded representation of s is calculated by averaging all embedding vectors for each token of the sentence, following the procedure in [36]. For each task, we exploited the same neural network structure. In particular, we used two hidden layers with $\frac{N}{4}$ neurons (using ReLu as activation function) and two output neurons with softmax activation function.

Given that a probe is basically a neural network classifier, its performance could evaluated in terms of accuracy. A high accuracy, considering that the input are just the embedding vector provided by the pre-trained NLM, should mean that those representation correctly encodes information regarding the task we analysed. However, studies such as [15, 16] demonstrated that accuracy is not a feasible

[1] https://www.kaggle.com/competitions/humor-detection/data

metric for these analyses. In fact, the authors of [15] show how probe classifiers achieve very high performance, if trained with high quantities of examples, even using totally random data in input. This is due to the strong capability of the neural network to find some patterns even in random data. For the same reason, perturbating the labels for creating meaningless control task lead to good results with probes trained with an adequate number of examples [16]. In both cases, a significant decrease in performance over these random control tasks can be seen only by training the probes with small datasets.

In order to solve these evaluation issues, Voita and Titov [17] introduced MDL (Minimum Description Length), a method based on information theory for measuring knowledge and capabilities of NLM through probing tasks. The authors of [17] demonstrated that MDL is very robust with respect to control tasks, random seeds, datasets and probe characteristics. The main idea behind MDL is measuring both the performance of the probe network but also its *effort*, in terms of the quantity of data necessary to obtain such a performance.

Inspired by the work by Aghazadeh et al. [13], we used the *online coding* version of MDL which works as follows. Instead of training a probe just once using the entire training set, first the method divides the training set into M portions of increasing size. Next, $M-1$ neural networks (all with the same hyperparameters, and all starting from the same weights initiated randomly) are trained, each one with a different portion. The first is trained with the first portion, the second with the second portion (which includes also the instances of the first one), and so on. The evaluation is conducted in terms of cross-entropy over a validation set, which basically consists of the "new instances" from the next portion. Therefore, a neural network trained on the i-th portion is evaluated by calculating the cross-entropy over the instances in the next portion excluding the ones used for training. Following [17], the portions consist of the 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.25, 12.5, 25, 50, and 100 % of the data.

This process is evaluated in terms of a metric called *codelength*, which intuitively can be seen as the sum of all the cross-entropy obtained during the process. More formally, it is defined as:

$$codelength = |M_0| \cdot log_2(K) + \sum_{i=0}^{M-1} (CE_i)$$

into which $|M_0|$ is the size of the first portion, K is the number of classes of the probing task, and CE_i is the cross-entropy calculated over the next portion instances.

Given that the codelength metric depends on the size of the training set, in [17] the authors propose another, more general, metric called *compression*, which is defined as:

$$compression = \frac{L \cdot log_2(K)}{codelength}$$

into which L is the size of the training set. Since in our experiments $K = 2$, we can finally define the compression metric as:

$$compression = \frac{L}{|M_0| + \sum_{i=0}^{M-1}(CE_i)}$$

In Section 5, almost all the results we obtained will be shown and explained in terms of this metric.

4.2. Trajectory and Layer Analysis

The probing task technique described above is often used only on a fully trained NLM to test its capabilities [9, 12, 30]. However, we extend this analysis and we aim to gain insight into how each language property is acquired by the NLMs, we implemented this probing approach at different stages of the model's training [19, 20]. By examining different training steps of the model, it is possible to trace its *learning trajectory* over a particular aspect, such as the tasks we described in Section 3.

Therefore, in our analysis, we perform the probing task on all the models' *checkpoints*. A checkpoint is an intermediate version of the model saved at a particular time during the training process. Although

many strategies to save a checkpoint can be adopted (such as saving every time a fixed amount of time has passed, every epoch, etc.) the checkpoints we consider are saved after the model has processed a specific number of tokens. More information regarding this aspect is provided in Section 5. Probing over the different checkpoints do not require particular expedients. In fact, it consists of training the probing tasks repeatedly over each checkpoint and evaluating its performance in terms of compression. By measuring how this metric changes over the checkpoints, it is possible to obtain the learning trajectory for that particular task.

Another fundamental aspect we consider in our study is analysing probing tasks in different parts of the NLM architecture and, in particular, among its layers. Thus, not only we execute and evaluate probing task considering all model's checkpoints, but also to the different layers of each checkpoint. In order to do that, we simply repeat the procedure explained above changing the probe input: for testing the last layer, we provide to the probe the embedded representation calculated by the last layer of the architecture, for testing the penultimate layer we provide the one calculated by the penultimate, etc. Therefore, considering a NLM with L layers and from which J checkpoints were saved, we execute the probing task $J \times L$ times. This way, we obtain the learning trajectory for a specific task for each layer.

5. Experimental Results

The experimental evaluation we conducted considers **three GPT-NeoX models** belonging to the Pythia benchmark suite [37]. Pythia is a collection of publicly released model of various sizes based on the GPTNeoX architecture [38]. All the models are trained over the same dataset (The Pile [39]) for almost $300B$ tokens. For each model, Pythia provides 154 checkpoints. The checkpoints are saved after 0 (i.e. with the model weights initialized randomly), 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1000 training steps and, after the 1000th step, every subsequent 1000 steps. For all three models, each step consists on training the model with $2M$ additional tokens. From this collection we consider the models with $70M$, $160M$ and $410M$ parameters. These models have a different number of layers, in fact they have respectively 6, 12 and 24 layers each.

For our layer analysis, we did not consider the $70M$ model due to the low number of layers of this model. Taking $160M$ and $410M$, in order to conduct a clearer and more meaningful comparison between two models with a different number of layers, we decided to group the layers by zones based on their order in the GPT stack. More specifically, starting from the layers closer to the input, we select three groups:

- **Bottom Layers** which consist of the first third of the layers (e.g., the first 4 layers for the 12 layers model), excluding the embedding layer;
- the **Middle Layers**, which consist of the middle third (e.g., from layer 5 to 7 included for the 12 layers model);
- the **Top Layers**, which consist of the last third, excluding the last one which we treat and visualise separately (e.g., from layer 8 to 11 for the 12 layers model).

As explained in Section 4.1, all the experiments reported in this section are in terms of the compression metric calculated by the MDL method [17]. This choice is not only coherent with the literature [15, 16] but it also helps understanding learning trajectories and differences among layers. In fact, accuracy may not show significant changes across different configurations. For instance, considering the $160M$ model and the *Coordinate structures* task in their final checkpoint, the layers have all a very high accuracy (97.55 on average, with a standard deviation of 0.02) but a compression that ranges from 2.1 to 10. Similarly, accuracy can reach very high values even in the very early stages of the trainin. Considering the same model for the *Coordinate structures* task, after 512 steps the probe trained on the last layer obtains an accuracy of 95.25. However, at the same step the compression is quite low (3.77), indicating that the concept is not totally contained in the representation.

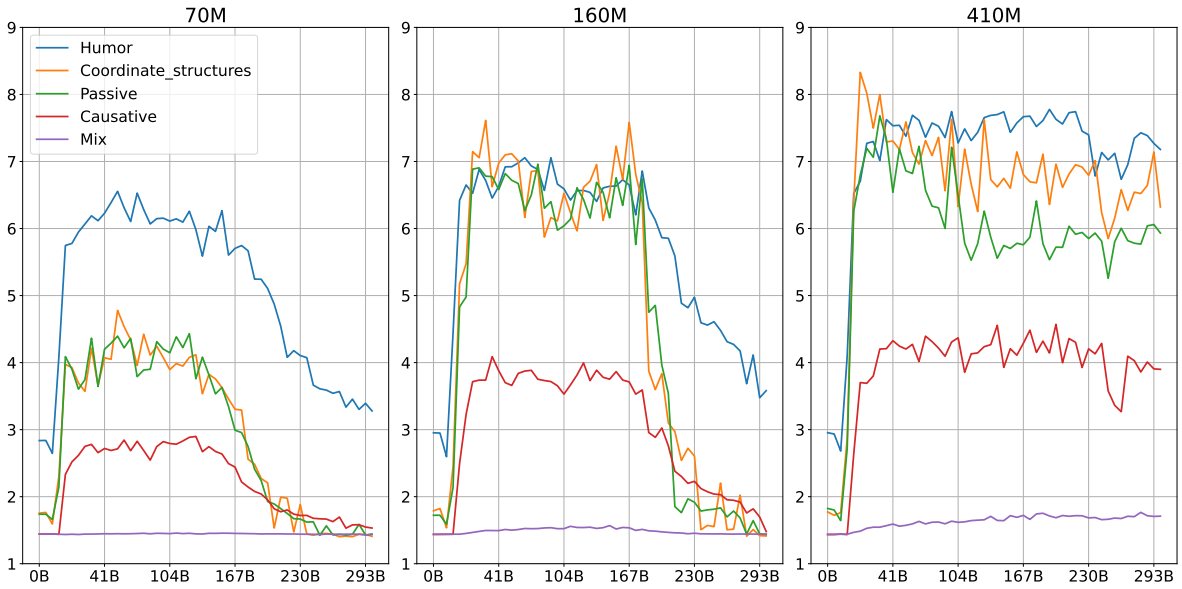

Figure 1: Comparison of the learning trajectories (in terms of compression) of the last layer of the GPT-NeoX models with $70M$, $160M$ and $410M$. Each line represents a different task (Humor in blue, Coordinate Structures in orange, Passive in green, Causative in red and Mix in purple).

5.1. Learning Trajectories and Task Comparison

The learning trajectories of the GPT-NeoX models, considering their last layer, for all the five probing tasks we considered are available in Figure 1. As expected, $410M$ is the best performing model, with higher compression with respect to $160M$ and $70M$, which is the smallest and worst performing model.

Comparing the different tasks we analysed, the highest values of compression are generally obtained by the Humor task. For $70M$, there is in fact a highly notable difference across all the learning trajectory (a maximum of 7.5 for *Humor* versus 4.8 for *Coordinate structures*, 4.4 for *Passive*, 2.9 for *Causative* and 1.5 for *Mix*). Considering $160M$, the performance obtained for Humor are very similar to those obtained for *Coordinate structures* and *Passive* until the last part of the training process. The *Causative* task has intermediate results in all the models we considered, reaching a maximum value of 4.4 of the compression for $410M$. This is probably due to its smaller dataset, which has only 2000 instances with respect to the 4000 instances of the other tasks.

The *Mix* task exhibits a completely different behaviour. In fact, it obtains very low values in all the models, not even reaching a value of 2 of compression for the most powerful model ($410M$), whereas *Causative* exceeds 4 and the other tasks can even exceed 7. Moreover, the learning trajectory is basically flat, showing no visible improvement during the learning process. This is probably due to the complexity of the dataset. In fact, the *Mix* task is composed by four different simple tasks and all models struggle to identify the sentence that contains an error, without focusing on a single aspect and not knowing exactly which the error is. For all the learning trajectories, we can see that most of the knowledge is acquired in the very early stage of the training (before the threshold of $41B$ tokens), without no noteworthy improvement afterwards.

In Figure 1, looking at the trajectories of the $70M$ and $160M$ models we can see that, at some point (about 150-180B tokens), the performance of all tasks except *Mix* decreases significantly. A possible explanation of this phenomenon is that, after extensive training, the last layer mostly focuses on Masked Language Modeling task which is typically used for training the NLM. Therefore, the layer probably "forgets" relevant linguistic information. Although this phenomenon has previously been observed in [21, 40], a more in-depth analysis of this aspect is required. In particular, it is important to note that the $410M$ model does not show performance decay in none of the tasks analyzed and instead the learning trajectories are mostly stable. We speculate that this may be due to its size (nearly 2.5 times the parameters of the $160M$ model) and a better management of the knowledge among his levels. However,

Figure 2: Learning trajectory (in terms of compression) for our considered tasks of the GPT-NeoX model with $160M$ parameters. The first subfigure refers to the first 4 layers (Bottom Layers), the second subfigure refers to the subsequent 3 layers (Middle Layers), the third one refers to the subsequent 4 (Top Layers). Each line represents the median result of the considered layers, across our five different tasks.

we cannot exclude that other tasks may have a performance decay or that continuing its lead would lead to a similar decrease in terms of compression.

5.2. Layer Comparison

The next experiment we conducted regards the study of the learning trajectory considering different layers of the NLMs. As for the previous experiments, we considered the same five probing tasks.

The results are available in Figure 2. For brevity's sake, we report only the results obtained for the $160M$ model but similar observations could be done for $70M$ and $410M$. As explained in Section 4.2, we divided the 12 layers of the model in three groups: **Bottom** (on the left), which refers to the lowest 4 layers of the architecture, **Middle** (in the center), which refers to the subsequent 3 layers and **Top** (on the right), which refers to the highest 4 layers, excluding the last one. In Figure 2 we report the median compression of the three groups across all the considered tasks.

Generally, the best results are obtained by the Middle layers, with just two exceptions: the *Causative* task, which has a slightly highest compression considering the Top layers, and the *Mix* task which, again, obtains very poor results with no notable differences among the three groups. The Top layers, however, obtain definitely better compression with respect to the Bottom ones, especially for *Coordinate structures* and *Passive* (with values ranging from 7 to 9 for Top and ranging mostly from 3 to 4 for Bottom). Instead, *Humor* reaches very high values of compressions in all three cases.

A more detailed look of the *Passive* and *Humor* tasks is given in Figure 3 for both the $160M$ and $410M$ models. We did not include $70M$ in this analysis due to its low number of layers, which limits the significance of this analysis. In particular, the plots show the performance of the *Bottom* (in blue), *Middle* (in orange) and *Top* (in green) layers and the last layer of the model. Besides representing the median value of each group layers (the line at the center of each area), we represent also the area between the 1st and 3rd quartile of the compression distribution obtained by the layers. This way, we show the variability of the performance among the different layers in each group.

In Figure 3, we can see that the $160M$ model has a low variability for all the layer groups, and especially for Top and Middle considering the *Humor* task. The highest variability can be seen for the Bottom layers in the *Passive* task. Instead, $410M$ presents a higher variability especially for the *Bottom* layers. Specifically, the *Passive* area for the Bottom layers ranges approximately from 2.5 to 7.

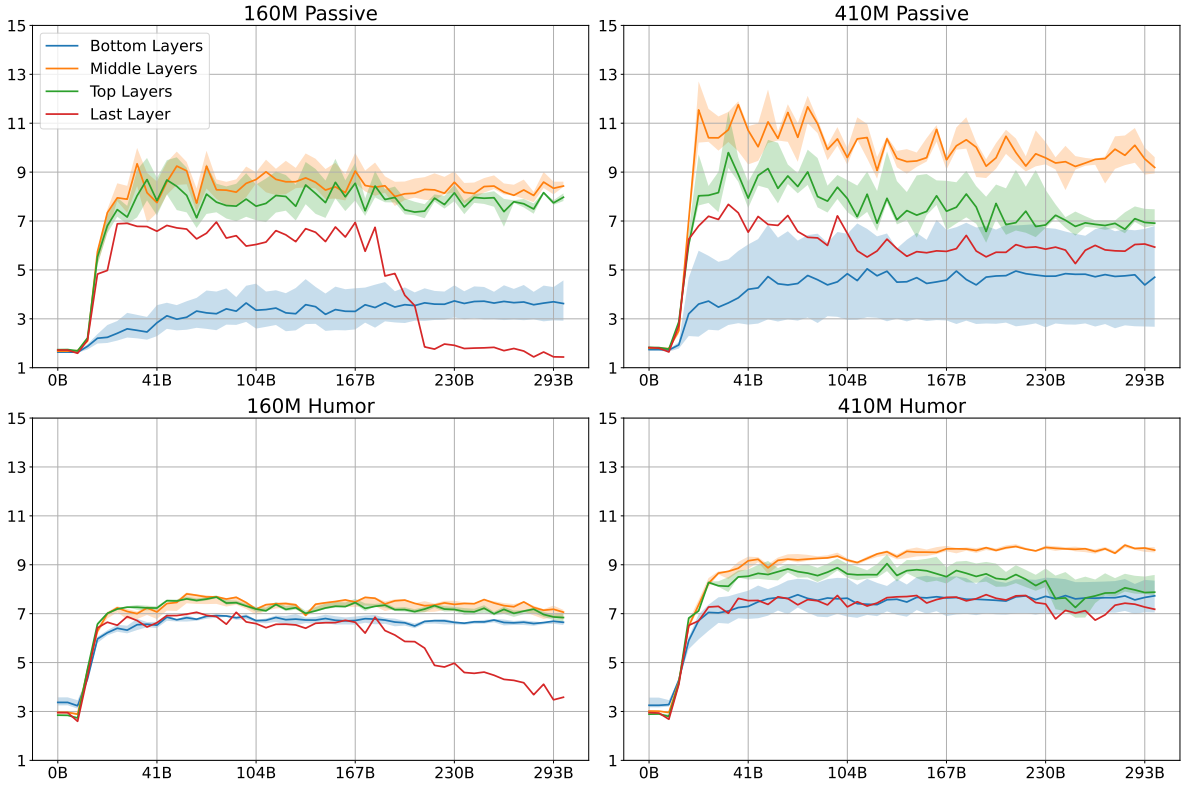

Figure 3: Analysis of the behaviour of the layers and of the Bottom, Middle and Top Layers for the Passive and Humor task, considering the $160M$ and $410M$ models. Each area represents the area between the 1st quartile and 3rd quartile of the compression obtained by the layers, the line in the area represents the median.

The lowest results are obtained by the first layer and the highest are obtained by layer 7. Nonetheless, it is important to point out that despite a higher number of layers, with respect to $160M$, the Top layers of 410 performs very well in the Humor task with a very high compressions and a very slow variability. This indicates that basically all layers encode very efficiently and very similarly this type of knowledge. Analysing the trajectories, we can observe a slight decay of performance for the *Passive* task, considering Middle and Top layers, and for *Humor* considering only the Top layers. These decays are not present in $160M$. The last layer (in red in Figure 3) presents the same behaviour we observed in Figure 1 and described in Section 5.1, with a very evident decrease of performance after about $167B$ tokens for the $160M$ model.

6. Conclusion and Future Work

We have investigated and compared the learning trajectories for three GPT-NeoX models of different size considering different probing tasks: three specific grammar tasks (*Causative*, *Coordinate structures* and *Passive*), a generic grammatical correctness task (*Mix*) and a purely semantic one (*Humor*).

Our evaluation, conducted in terms of compression exploiting the MDL method, shows that the considered models acquire knowledge both related to grammar and related to a specific aspect of semantics, with very high performance for *Coordinate structures*, *Passive* and *Humor*. However, the *Mix* task shows a very low compression which denotes a limited capability of a NLM to discern whether a sentence is correct or not in general and without focusing on a single, specific aspect. The learning trajectories we analysed showed that most of this knowledge is acquired early and, for the most part, the compression is stable or (only for the last layer) even decreasing. The performance decay phenomenon is probably due to the specialisation of the last layer in the Masked Language Modeling task for which

it is trained [21, 40].

Moreover, we have analysed the different layers of the considered models grouping them in three groups: the Bottom layers (the third closest to the input), the Middle layers and the Top ones. Generally, the best results are obtained by the Middle layers, whereas the Bottom layers provide the worst results, especially for the grammar tasks.

As future work, we want to explore the performance decrease of the last layer with more tasks and verifying possible explanations. Moreover, this analysis requires the access to open source model which provide the embedded representations of words and sentences. An important development would be to devise similar procedures to analyse closed source models.

Acknowledgments

This work has been partly funded by Regione Lombardia through the initiative "Programma degli interventi per la ripresa economica: sviluppo di nuovi accordi di collaborazione con le università per la ricerca, l'innovazione e il trasferimento tecnologico" - DGR n. XI/4445/2021.

References

[1] M. Polignano, C. Musto, R. Pellungrini, E. Purificato, G. Semeraro, M. Setzu, XAI.it 2024: An Overview on the Future of Explainable AI in the era of Large Language Models, in: Proceedings of 5th Italian Workshop on Explainable Artificial Intelligence, co-located with the 23rd International Conference of the Italian Association for Artificial Intelligence, Bolzano, Italy, November 25-28, 2024, CEUR. org, 2024.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[4] N. Arici, A. E. Gerevini, M. Olivato, L. Putelli, L. Sigalini, I. Serina, Real-world implementation and integration of an automatic scoring system for workplace safety courses in italian, Future Internet 15 (2023) 268. URL: https://doi.org/10.3390/fi15080268. doi:10.3390/FI15080268.

[5] E. M. Bender, T. Gebru, A. McMillan-Major, S. Shmitchell, On the dangers of stochastic parrots: Can language models be too big?, in: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 610–623. doi:10.1145/3442188.3445922.

[6] Y. Belinkov, J. Glass, Analysis methods in neural language processing: A survey, Transactions of the Association for Computational Linguistics 7 (2019) 49–72. URL: https://aclanthology.org/Q19-1004. doi:10.1162/tacl_a_00254.

[7] K. Clark, U. Khandelwal, O. Levy, C. D. Manning, What does BERT look at? an analysis of BERT's attention, in: T. Linzen, G. Chrupała, Y. Belinkov, D. Hupkes (Eds.), Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Association for Computational Linguistics, Florence, Italy, 2019, pp. 276–286. URL: https://aclanthology.org/W19-4828. doi:10.18653/v1/W19-4828.

[8] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language models as knowledge bases?, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 2463–2473. URL: https://aclanthology.org/D19-1250. doi:10.18653/v1/D19-1250.

[9] A. Miaschi, D. Brunato, F. Dell'Orletta, G. Venturi, Linguistic profiling of a neural language model, in: D. Scott, N. Bel, C. Zong (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 745–756. URL: https://aclanthology.org/2020.coling-main.65. doi:10.18653/v1/2020.coling-main.65.

[10] A. Köhn, What's in an embedding? analyzing word embeddings through multilingual evaluation, in: L. Màrquez, C. Callison-Burch, J. Su (Eds.), Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 2067–2073. URL: https://aclanthology.org/D15-1246. doi:10.18653/v1/D15-1246.

[11] Y. Belinkov, Probing classifiers: Promises, shortcomings, and advances, Computational Linguistics 48 (2022) 207–219. URL: https://aclanthology.org/2022.cl-1.7. doi:10.1162/coli_a_00422.

[12] T. Caselli, I. Dini, F. Dell'Orletta, How about time? probing a multilingual language model for temporal relations, in: N. Calzolari, C.-R. Huang, H. Kim, J. Pustejovsky, L. Wanner, K.-S. Choi, P.-M. Ryu, H.-H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T. K. Lee, E. Santus, F. Bond, S.-H. Na (Eds.), Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 3197–3209. URL: https://aclanthology.org/2022.coling-1.283.

[13] E. Aghazadeh, M. Fayyaz, Y. Yaghoobzadeh, Metaphors in pre-trained language models: Probing and generalization across datasets and languages, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2037–2050. URL: https://aclanthology.org/2022.acl-long.144. doi:10.18653/v1/2022.acl-long.144.

[14] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[15] K. Zhang, S. Bowman, Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis, in: T. Linzen, G. Chrupała, A. Alishahi (Eds.), Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 359–361. URL: https://aclanthology.org/W18-5448. doi:10.18653/v1/W18-5448.

[16] J. Hewitt, P. Liang, Designing and interpreting probes with control tasks, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 2733–2743. URL: https://aclanthology.org/D19-1275. doi:10.18653/v1/D19-1275.

[17] E. Voita, I. Titov, Information-theoretic probing with minimum description length, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 183–196. URL: https://aclanthology.org/2020.emnlp-main.14. doi:10.18653/v1/2020.emnlp-main.14.

[18] N. Saphra, A. Lopez, Understanding learning dynamics of language models with SVCCA, in: J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 3257–3267. URL: https://aclanthology.org/N19-1329. doi:10.18653/v1/N19-1329.

[19] C.-H. Chiang, S.-F. Huang, H.-y. Lee, Pretrained language model embryology: The birth of ALBERT, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6813–6828. URL: https://aclanthology.org/2020.emnlp-main.553. doi:10.18653/v1/2020.emnlp-main.553.

[20] Z. Liu, Y. Wang, J. Kasai, H. Hajishirzi, N. A. Smith, Probing across time: What does RoBERTa know and when?, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 820–842. URL: https://aclanthology.org/2021.findings-emnlp.71. doi:10.18653/v1/2021.findings-emnlp.71.

[21] A. Rogers, O. Kovaleva, A. Rumshisky, A primer in BERTology: What we know about how BERT works, Transactions of the Association for Computational Linguistics 8 (2020) 842–866. URL: https://aclanthology.org/2020.tacl-1.54. doi:10.1162/tacl_a_00349.

[22] H. Zhao, H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, M. Du, Explainability for large language models: A survey, ACM Trans. Intell. Syst. Technol. 15 (2024). URL: https://doi.org/10.1145/3639372. doi:10.1145/3639372.

[23] J. Vig, A multiscale visualization of attention in the transformer model, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations, Association for Computational Linguistics, 2019, pp. 37–42.

[24] Y. Guan, J. Leng, C. Li, Q. Chen, M. Guo, How far does BERT look at: Distance-based clustering and analysis of BERT's attention, in: Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020, International Committee on Computational Linguistics, 2020, pp. 3853–3860.

[25] L. Serina, L. Putelli, A. E. Gerevini, I. Serina, Synonyms, antonyms and factual knowledge in BERT heads, Future Internet 15 (2023) 230.

[26] L. Putelli, A. E. Gerevini, A. Lavelli, T. Mehmood, I. Serina, On the behaviour of bert's attention for the classification of medical reports, in: C. Musto, R. Guidotti, A. Monreale, G. Semeraro (Eds.), Proceedings of the 3rd Italian Workshop on Explainable Artificial Intelligence co-located with 21th International Conference of the Italian Association for Artificial Intelligence(AIxIA 2022), Udine, Italy, November 28 - December 3, 2022, volume 3277 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 16–30.

[27] H. ElSahar, P. Vougiouklis, A. Remaci, C. Gravier, J. S. Hare, F. Laforest, E. Simperl, T-rex: A large scale alignment of natural language with knowledge base triples, in: N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, T. Tokunaga (Eds.), Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018, European Language Resources Association (ELRA), 2018.

[28] Z. Jiang, F. F. Xu, J. Araki, G. Neubig, How can we know what language models know, Trans. Assoc. Comput. Linguistics 8 (2020) 423–438.

[29] A. Gupta, G. Boleda, M. Baroni, S. Padó, Distributional vectors encode referential attributes, in: L. Màrquez, C. Callison-Burch, J. Su (Eds.), Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 12–21. URL: https://aclanthology.org/D15-1002. doi:10.18653/v1/D15-1002.

[30] G. Jawahar, B. Sagot, D. Seddah, What does BERT learn about the structure of language?, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 3651–3657. URL: https://aclanthology.org/P19-1356. doi:10.18653/v1/P19-1356.

[31] J. Hewitt, C. D. Manning, A structural probe for finding syntax in word representations, in: J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4129–4138. URL: https://aclanthology.org/N19-1419. doi:10.18653/v1/N19-1419.

[32] M. Dusi, N. Arici, A. E. Gerevini, L. Putelli, I. Serina, Graphical identification of gender bias in BERT with a weakly supervised approach, in: D. Nozza, L. C. Passaro, M. Polignano (Eds.), Proceedings of the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI 2022) co-located with 21th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2022), Udine, November 30th, 2022, volume 3287 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 164–176.

[33] M. Dusi, N. Arici, A. E. Gerevini, L. Putelli, I. Serina, Discrimination bias detection through categorical association in pre-trained language models, IEEE Access (2024) 1–1. doi:10.1109/ACCESS.2024.3482010.

[34] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A lite BERT for self-supervised learning of language representations, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020. URL: https://openreview.net/forum?id=H1eA7AEtvS.

[35] A. Warstadt, A. Parrish, H. Liu, A. Mohananey, W. Peng, S. Wang, S. R. Bowman, Blimp: The benchmark of linguistic minimal pairs for english, Trans. Assoc. Comput. Linguistics 8 (2020) 377–392. URL: https://doi.org/10.1162/tacl_a_00321. doi:10.1162/TACL_A_00321.

[36] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. URL: https://aclanthology.org/D19-1410. doi:10.18653/v1/D19-1410.

[37] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, O. van der Wal, Pythia: A suite for analyzing large language models across training and scaling, in: A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, J. Scarlett (Eds.), International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, volume 202 of *Proceedings of Machine Learning Research*, PMLR, 2023, pp. 2397–2430.

[38] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang, M. Pieler, U. S. Prashanth, S. Purohit, L. Reynolds, J. Tow, B. Wang, S. Weinbach, GPT-NeoX-20B: An open-source autoregressive language model, in: Proceedings of the ACL Workshop on Challenges & Perspectives in Creating Large Language Models, 2022. URL: https://arxiv.org/abs/2204.06745.

[39] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, C. Leahy, The pile: An 800gb dataset of diverse text for language modeling, CoRR abs/2101.00027 (2021). URL: https://arxiv.org/abs/2101.00027. arXiv:2101.00027.

[40] J. Wallat, J. Singh, A. Anand, BERTnesia: Investigating the capture and forgetting of knowledge in BERT, in: A. Alishahi, Y. Belinkov, G. Chrupała, D. Hupkes, Y. Pinter, H. Sajjad (Eds.), Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, Association for Computational Linguistics, Online, 2020, pp. 174–183. URL: https://aclanthology.org/2020.blackboxnlp-1.17. doi:10.18653/v1/2020.blackboxnlp-1.17.