Learning Trajectories of Figurative Language for Pre-Trained Language Models

Nicola Arici, Luca Putelli, Ejdis Gjinika, Ivan Serina, Alfonso Emilio Gerevini
Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Brescia, Italy
Correspondence: nicola.arici@unibs.it, luca.putelli@unibs.it

Abstract

Figurative language and figures of speech, such as metaphors and hyperboles, are used every day in written and oral communication among human beings. Nonetheless, this imaginative use of words in a non literal way requires a solid understanding of semantics and a deep real-world knowledge. In the longstanding debate about whether Neural Language Models (NLMs) really have a full understanding of text, analysing how they can recognise figurative language can provide some intuition of their functioning, their capabilities and their limits. Therefore, in this paper, we exploit probing tasks to study how several NLMs of different sizes recognise four different figures of speech: hyperboles, metaphors, oxymorons and pleonasms. We analyse whether this information is learned and how it is acquired during the training of the model, describing its learning trajectory. Moreover, we analyse which layers have a better comprehension of figurative language and the influence of pre-training data. Datasets and code are available at https://github.com/nicolarici/learning-trajectories.

1 Introduction

Investigating the capabilities of Neural Language Models (NLMs) is a well established research area in Natural Language Processing (NLP) (Belinkov, 2022). Major studies have been conducted on how these models understand grammar (Miaschi et al., 2020), how attention weights encode dependency relations (Vig and Belinkov, 2019) and on world knowledge contained in such models (Roberts et al., 2020; Heinzerling and Inui, 2021). This is often done by probing tasks, i.e. by training a classifier to verify if the embedded representation of words (or sentences) produced by the models contain some specific information, such as the part-of-speech or whether a word describes the subject of the sentence (Köhn, 2015).

However, analysing semantic-related aspects is quite more complex (van Dijk et al., 2023). In fact, a word can have different meanings and nuances depending on the context in which it appears. Moreover, it can be used literally, metaphorically, or even ironically. With respect to grammar and syntax, into which words have definite and recognisable characteristics which can be easily annotated, these nuances increase the difficulty of creating datasets and designing simple tasks for this kind of evaluation.

In this paper, we tackle the problem on how NLMs deal with semantics by analysing figurative language and, in particular, figures of speech. Understanding figurative language is a challenging task in the overall context of text comprehension (Shutova, 2011). As an example, consider the sentence "The Real Madrid players go 3,000 kilometers per hour". In order to understand that it contains a hyperbole, a lot of implicit semantic and real-world knowledge is needed: that Real Madrid players are humans, that kilometers per hour is a unit of measure of speed, and that humans do not reach that speed. Since figures of speech rely on using words in a nuanced, rhetorical and imaginative way, investigating them can provide interesting insight into the semantic knowledge of NLMs, their inner workings and some of their limits. From a practical perspective, understanding how this knowledge is acquired during training (in terms of quantity and types of data required) could provide some insights for optimizing the training process.

We structure the recognition of the presence of a figure of speech (in particular, we analyse metaphors, oxymorons, hyperboles and pleonasms for the English language) in a sentence as a probing task, and we evaluate it through the state-of-the-art method of Minimum Description Length (MDL) by Voita and Titov (2020). With this configuration, we can also study the performance of the different

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 14440–14461
November 4-9, 2025

layers across the NLMs architecture. Most importantly, we are not only interested in the capabilities of the fully trained models. Following the works in Chiang et al. (2020) and Liu et al. (2021), we also perform this analysis through the entire training process of an NLM, analysing how this semantic-based knowledge is acquired and how it evolves during training. By this procedure, we can also analyse how different data (a small general purpose Wikipedia dump, the literary corpus Project Gutenberg and the extensive The Pile dataset (Gao et al., 2021)) influence the learning of figurative language information. In summary, through this work we aim to address these research questions:

Q1. How do NLMs learn figurative language? Is there a common pattern or different figures of speech are learned in different ways?

Q2. Do larger models have a better understanding of figurative language?

Q3. Which layers are mostly associated with figurative language-related knowledge?

Q4. How is such knowledge acquired during training? How is it influenced by different data?

2 Related Work

Alongside the success and diffusion of Neural Language Models, one of the top research priorities became their explanation and interpretation (Rogers et al., 2020; Zhao et al., 2024a). In this line of work, many approaches have been proposed, such as studying the behaviour of self-attention mechanisms (Clark et al., 2019; Vig, 2019); assessing whether some forms of real-world knowledge are learned by the models (Petroni et al., 2019; Jiang et al., 2020), or analysing lexical and grammatical capabilities through probing tasks (Belinkov and Glass, 2019; Belinkov, 2022).

A probe is a simple neural network model that uses the embedded representation of words or sentences generated by a pre-trained language model and that it is trained to solve a specific supervised task (Köhn, 2015; Gupta et al., 2015). This technique has been exploited to study syntactic properties (Jawahar et al., 2019; Miaschi et al., 2020) including dependency parsing (Hewitt and Manning, 2019) and temporal relations (Caselli et al., 2022). Differently from these works, our focus is not on syntax or grammar, but on semantics and, in particular, on figurative language. Recent methods such as Sparse Autoencoders (Huben et al., 2024) and Sparse Probing (Gurnee et al., 2023) work in an unsupervised framework focusing on identifying the function of individual neurons (for instance, whether a neuron activates in the presence of words beginning with the letter W (Huben et al., 2024)). This is substantially different from our study, which analyses whether the whole word representation encodes a high level concept in a supervised manner.

Probing for figurative language has been studied by some works focusing on concepts like compositionality (Liu and Neubig, 2022; Dankers et al., 2022) and idiomaticity (Garcia et al., 2021; Tan and Jiang, 2021). Regarding figures of speech, which are the focus of our study, Aghazadeh et al. (2022) analyse how metaphors are recognised by several NLMs across different datasets and languages. Similarly, the work in Schneidermann et al. (2023) analyses hyperboles on three different Transformer-based encoder models. Unlike these works, we do not focus on a single detection task; instead, we expand the scope to multiple figures of speech, including also oxymorons and pleonasms. Moreover, they focus only on the fully trained model. Instead, we analyse how these capabilities are learned by NLMs during their training.

Understanding how and when a property is learned by an NLM is a relatively new and under investigated research field. Saphra and Lopez (2019) investigated how linguistic properties are encoded in the hidden representations of a LSTM language model. They discovered that syntactic features (such as the part-of-speech) are learned early in the training, whereas the learning of topic-related information is acquired later. Similar results were obtained by Chiang et al. (2020) for the ALBERT model (Lan et al., 2020). Liu et al. (2021) conduct a similar analysis of RoBERTa (Liu et al., 2019) learning trajectories focusing more on factual and common sense knowledge, finding that this type of information is learned more in depth as the training progresses. To the best of our knowledge, our work is the first study on how figurative language is learned during the model training.

Evaluating the effect of the pre-training on the behaviour of NLMs is another underexplored field. Primarily, the works by Longpre et al. (2024) and Zhao et al. (2024b) evaluate how different sources (books, code repositories, web pages) influence the performance on several NLP downstream tasks.

Differently from these works, we analyse how different datasets change the learning trajectory of specific semantic knowledge through probing tasks.

3 Case Studies and Datasets

We analyse four types of figures of speech: Hyperbole, Metaphor, Oxymoron and Pleonasm. In the following, we explain them in detail and introduce the datasets used for our analysis. All datasets are in English, publicly available and released freely for research purposes.

Hyperbole A hyperbole is an exaggeration to amplify or reduce the representation of the connotations of what is being communicated over a qualitative or quantitative scale (Burgers et al., 2016). As a dataset, we used the one introduced by Troiano et al. (2018) in the slightly modified version released in Schneidermann et al. (2023). It consists of 1396 sentences collected from the web and annotated by several human annotators. The dataset is divided equally between hyperbolic and literal sentences.

Metaphor Following Strapparava (2018), a metaphor can be defined as the replacement of one word by another one whose literal sense bears a resemblance to the literal sense of the word replaced. Unlike a comparison, where affinities and divergences between the compared entities are explicitly shown, in a metaphor the two entities are merged into one. A well-known benchmark of manually annotated metaphors is the Language Computer Corporation (LCC) by Mohler et al. (2016). More specifically, we adopt the binary version of LCC released in Aghazadeh et al. (2022). From the $40K$ sentences contained in Binary LCC, we randomly selected 1600 of them almost equally distributed (785 literals and 815 metaphors) to perform a better comparison with the other smaller datasets.

Oxymoron An oxymoron is a juxtaposition of two semantically opposite terms; the terms may be morphologically connected (as in *happily unhappy*) or not (as in *screaming silence*) (Bolognesi et al., 2024). For our analysis, we created a new dataset starting from the 287 oxymorons collected and annotated by Xu et al. (2023). While their dataset contains only positive instances (i.e. sentences containing an oxymoron), we generated the negative ones ourselves. The generation process exploits the GPT-4 model through the official OpenAI API. For each oxymoron, we generated a sentence which contains the same words that compose the oxymoron but used in a literal meaning. Then, the generated sentences were validated by the authors. More details about the prompt and the generation process are given in Appendix A.1. The final dataset contains 1564 examples evenly distributed between positive and negative.

Pleonasm A pleonasm is an overabundant expression formed by the addition of one or more words that are not necessary from a grammatical or conceptual point of view. It is often used to provide the sentence with emphasis, confidence or verbosity (Lehmann, 2005). For our analysis, we exploited a dataset based on the Semantic Pleonasm Corpus benchmark (SPC) by Kashefi et al. (2018). This dataset contains 3019 sentences with different pleonasms made by a pair of consecutive words. Each instance has been labelled by human annotators that evaluate whether *one* of the two words is redundant, *both* words are redundant or *none* of them are. To adapt this dataset to our binary task, we discard the sentences labelled as *both*. Our final dataset contains 3002 sentences, of which 1720 with a pleonasm and 1282 without.

4 Methodology

In this section, we describe the method we use to analyse the learning trajectories of NLMs applied to figurative language.

4.1 Figurative Language as a Probing Task

Our probes aim to determine how much is known by an NLM in terms of figurative language. More specifically, the goal is to identify whether a sentence contains a specific figure of speech or not.

The task is designed as follows. Considering one figure of speech, its relative dataset and a pre-trained NLM, we create a feed-forward neural network classifier that receives as input the N-dimensional embedded representation of a sentence s and should output 1 if s contains the figure of speech and 0 otherwise. Following the approach by Reimers and Gurevych (2019), we calculate the embedded representation of s as the average of the representation of each token in the sentence. The neural network has two hidden layers with $\frac{N}{4}$ neurons with ReLU as activation function and an output layer with 2 neurons using softmax activation function. For training the classifier to recognise a figure of speech, we use 80% of the corresponding dataset as a training set. During the training phase,

the weights of the pre-trained NLM are frozen; thus, the model is not fine-tuned for the task. The dimension of the hidden layers and the other hyperparameters of the classifier were obtained by a preliminary random search on a validation set (10% of the dataset).

In principle, the performance of the classifier could be evaluated through standard machine learning metrics, such as accuracy or F-Score. Since the NLM is frozen, good metric values should mean that the embedded representation correctly encodes specific knowledge relative to the figure of speech. However, this assertion has been disputed by several studies. For instance, Zhang and Bowman (2018) show how the same probe classifier can achieve the same performance in a linguistic task if fed with a pre-trained representation or a random one; a significant difference in terms of accuracy would only be seen with a small set of training data. Similar results were obtained by Hewitt and Liang (2019), who tested the method with several control tasks, for which they show how a probe classifier provides very similar results both predicting a real linguistic phenomenon and a random label.

A solution to these issues is the **Minimum Description Length** (MDL) probing method by Voita and Titov (2020). MDL is based on information theory, and it has been proven to be robust with respect to changes in the probe settings, the choice of random seed and the aforementioned issues with random representations and control tasks. Intuitively, MDL measures not only the classifier performance, but also the effort required to achieve such performance through the *codelength* metric. Voita and Titov (2020) propose two alternative approaches for MDL: the *variational coding* and the *online coding*. The results obtained by these two methods are consistent with each other. Following Aghazadeh et al. (2022), for our analysis we used the latter method which works as follows. First the training set for our probing task is divided into M portions of increasing dimension.[1] Next, for each of these portions, a classifier is trained and evaluated through the cross-entropy metric. More precisely, a classifier trained on the i-th portion is evaluated by calculating the cross-entropy over the instances in the $(i+1)$-th portion excluding those used for training. The *codelength* is defined as:

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

where $|M_0|$ is the size of the first training portion, K is the number of classes of the probing task, and CE_i is the cross-entropy. In our case, we consider only binary tasks, so $K = 2$.

Since the codelength is related to the size of the dataset $|D|$, in order to draw a better comparison, our analysis is performed in terms of *compression*. This metric is defined as:

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

Since $K = 2$, the compression can be seen as the ratio between the dataset size and the codelength.[2]

4.2 Learning Trajectory and Layer Analysis

In the literature, probing tasks have been applied to NLMs to assess the capabilities of a model which has been fully trained and released to the public (Jawahar et al., 2019; Miaschi et al., 2020; Caselli et al., 2022). However, the same techniques can be used for studying how a model acquires these capabilities during its different training steps (Chiang et al., 2020; Liu et al., 2021). By probing the model through these steps, it is possible to derive what has been called the *learning trajectory* of the model with respect to some specific aspect.

Therefore, we do not perform the probing task only on a final, fully trained NLM but also on its *checkpoints*. A checkpoint is an intermediate version of the model (and its weights) saved during its training. For NLMs, a checkpoint is saved every time the model has received a certain number of tokens in input. Considering a specific figure of speech and given a set of J checkpoints of a pre-trained model, we execute the probing task and measure repeatedly its performance in terms of compression with the MDL method over all the J checkpoints.

Moreover, we extend our analysis to the different layers that compose the NLMs considered. By this experiment, we aim to verify if there are significant differences across layers and their knowledge related to figurative language. Similarly to what

[1] As in Voita and Titov (2020) the portions are 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.25, 12.5, 25, 50, 100 %.

[2] For more details regarding the MDL method, the reader can refer to the original paper (Voita and Titov, 2020) and to this blog post: https://lena-voita.github.io/posts/mdl_probes.html.

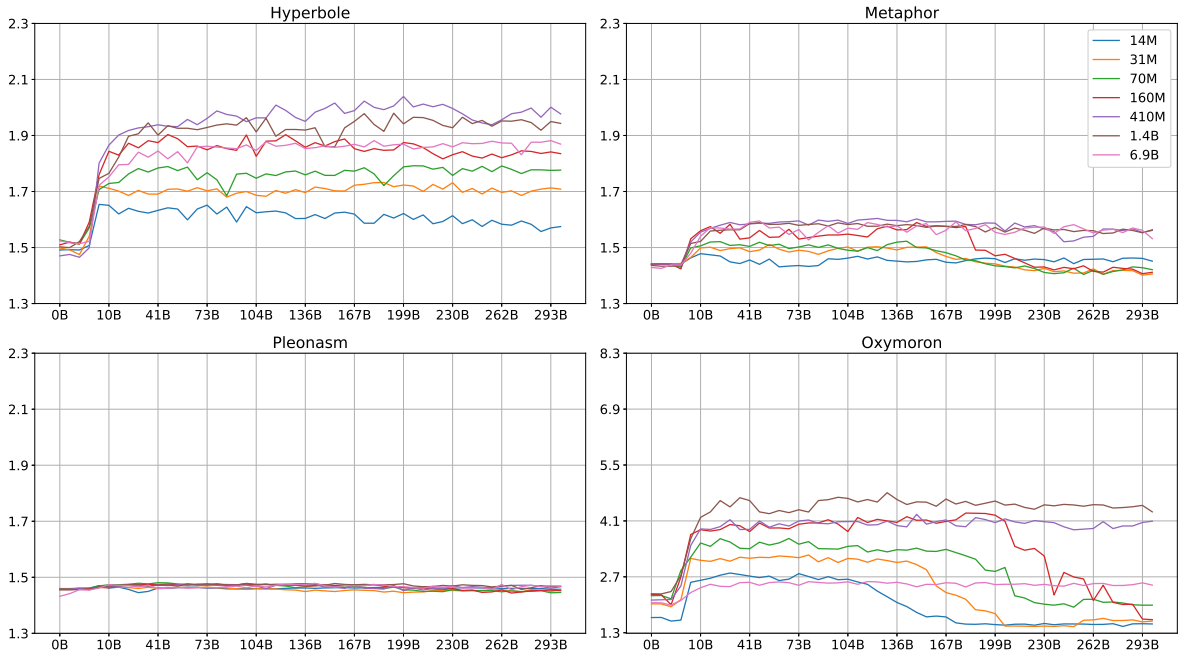

Figure 1: Learning trajectories of the GPT-NeoX models considering their last layer. Each plot regards a different figure of speech. On the x-axis, we report the number of tokens used in training; on the y-axis we report the compression calculated using MDL. Each line represents a model: in blue, the one with $14M$ parameters, in orange $31M$, in green $70M$, in red $160M$, in purple $410M$, in brown $1.4B$, and in pink $6.9B$.

has been previously described, given a checkpoint (final or intermediate) made by L layers, we repeat the probing task and its evaluation for the embedded representations provided by each layer.

5 Experimental Evaluation

For our experimental evaluation, we considered several NLMs (for the English language) and their available checkpoints, based on the GPT-NeoX architecture (Black et al., 2022).

We evaluate each model in terms of compression through the MDL method over all the available checkpoints. From a theoretical point of view, as described in Section 4, MDL is the state-of-the-art method for evaluating probing tasks, given its robustness versus randomness and control tasks. Moreover, from a practical point of view, using accuracy leads to a definitely less clear evaluation. For instance, Hyperbole accuracy ranges from 70% to 80% for almost the entire trajectory, with no real distinctions among models despite their much different size and number of parameters. Instead, compressions lead to more stable and understandable results. For the engaged reader, results in terms of accuracy are reported in Appendix C.

We considered the Pythia benchmark suite (Biderman et al., 2023), i.e. **GPT-NeoX models** (Black et al., 2022), a variant of GPT very similar to GPT-3 (Brown et al., 2020) and GPT-J (Wang and Komatsuzaki, 2021). All these models were trained over The Pile dataset (Gao et al., 2021), for a total of almost $300B$ tokens. For each model, a checkpoint is saved after 0 (i.e. with the model weights randomly initialised), 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1000 and every subsequent 1000 training steps. Each step consists of $2M$ tokens. The models considered differ in terms of number of layers, heads, embedding dimension and therefore for their overall number of parameters. We selected the models with $14M$, $31M$, $70M$, $160M$, $410M$, $1.4B$ and $6.9B$ parameters.[3]

5.1 Results for the Last Layer (Q1, Q2)

In this section, we conduct a detailed comparison of the performance of GPT models, providing answers to **Q1** and **Q2**.

Results are available in Figure 1, which shows the learning trajectories of the last layer for each model. Considering the values of compression, on the y-axis of each plot, **we can see notable differences among the figures of speech considered**. Those which are mostly recognisable by all the

[3]More details are available at https://github.com/EleutherAI/pythia and in Appendix B

Model	L	Hyperbole		Metaphor		Oxymoron		Pleonasm	
		BL	↑ %	BL	↑ %	BL	↑ %	BL	↑ %
14M	6	5	9.7	6	0.0	5	28.0	5	0.4
31M	6	5	21.4	2	6.3	3	91.0	1	1.5
70M	6	4	24.2	4	7.0	2	87.8	0	2.5
160M	12	8	25.1	9	11.2	6	206.8	10	2.1
410M	24	14	9.2	15	3.5	12	50.3	12	2.3
1.4B	24	13	9.2	11	1.9	11	72.9	10	2.8
6.9B	32	18	4.8	17	5.1	11	13.7	9	3.6

Table 1: Results of the layer analysis conducted on the GPT-NeoX models. Column L gives the total number of layers; the columns of the considered figures of speech give the best performing layer (column BL) and the improvement of the compression metric in percentage with respect to the last layer (column ↑ %).

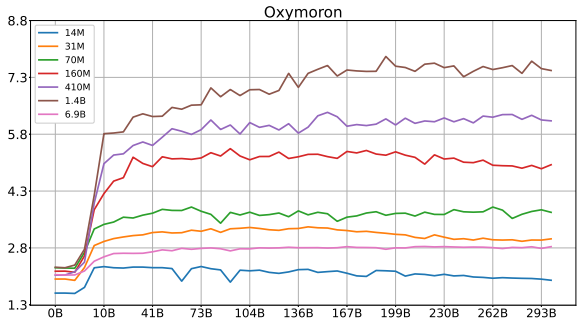

Figure 2: Learning trajectories of the GPT-NeoX models for Oxymoron, considering their best layer. On the x-axis, we report the number of tokens used in training; on the y-axis we report the compression. Each line represents a model: in blue, the one with $14M$ parameters, in orange $31M$, in green $70M$, in red $160M$, in purple $410M$, in brown $1.4B$, and in pink $6.9B$.

GPT models are Oxymoron, which reaches a compression higher than 4.5 with the $1.4B$ model, and Hyperbole, which reaches a compression higher than 2 with the $410M$ and $1.4B$ models. Some difficulties can be observed for Metaphor. Even if the three biggest models ($410M$, $1.4B$ and $6.9B$) reach a compression of about 1.6, this value is notably lower than the ones obtained for Oxymoron and Hyperbole. For Pleonasm, all models obtain similar performance and do not improve with respect to the randomly initialised model, which obtains a compression of 1.46. In order to improve performance for Metaphor and Pleonasm, we have tried tuning the hyperparameters of the neural classifier, checking 100 further configurations, obtaining no improvement. For Metaphor, even doubling and quadrupling the training data did not lead to improvements. This suggests that recognising Metaphor and Pleonasm is particularly difficult for the GPT models we considered.

For Hyperbole, Oxymoron and Metaphor, we can see that, **on average, bigger models obtain a better performance**. Taking Hyperbole as an example, $31M$ obtains a maximum value of 1.73 whereas $160M$ obtains 1.85 and $410M$ obtains 2.04. Notable exceptions are $1.4B$ and $6.9B$ for Hyperbole (which do not perform better than $410M$) and $410M$, $1.4B$ and $6.9B$ for Metaphor, which reach similar compression.

The learning trajectories of Hyperbole, Oxymoron and Metaphor share some similarities. As can be seen in Figure 1, **most of the models start increasing their compression value in the initial phases of the training** (before $10B$ tokens) **and they do not have remarkable improvements afterwards**. This suggests that, in order to acquire the capability of recognising these figures of speech, the considered GPT models do not require a massive dataset. This behaviour is consistent with the one observed by Liu et al. (2021) for linguistic tasks, whereas factual knowledge and reasoning have a more gradual learning process.

In some cases, we can see a worsening of compression during the training process. Whereas the $14M$ model for Hyperbole obtains only a slight worsening (from 1.65 to 1.58), this behaviour is more evident for Metaphor and Oxymoron. For Metaphor, the $160M$ model after receiving $180B$ tokens in training rapidly decreases its performance, passing from 1.58 to 1.49 of compression. This behaviour is even more drastic for Oxymoron: all models apart from $410M$ and $1.4B$ progressively worsen their performance in the latter stages of training. An interesting aspect that can be noted is that smaller models worsen earlier: $14M$ starts decreasing its compression after about $100B$ tokens, $31M$ after $142B$, $70M$ after $167B$ and $160M$ after $199B$. We do not observe this behaviour for $410M$ and $1.4B$.

Finally, for Oxymoron the $6.9B$ model exhibits a peculiar behaviour: similarly to smaller models, its performance does not increase much over training. However, as the larger models, its compression is stable and does not worsen after a certain iteration. We conducted further experiments to analyse whether the low compression was due to the probe dimensions. More specifically, we performed a random search of the probe hyperparameters (number of hidden layers from 0 to 4 and number of neurons from $N/2$ to $N/32$, where N is the embedding size). However, we did not see any improvement.

Thus, we claim this is a characteristic of the model.

We have also conducted another experiment for all case studies on $410M$, considering as the probe input only the span of the sentence containing the figure of speech. The complete results are reported in Appendix C. For Metaphor and Pleonasm we did not observe a visible difference in performance; on the contrary, for Oxymoron and Hyperbole, a sharp deterioration in performance is observed, achieving, respectively, 1.64 (vs 1.98) and 1.85 (vs 4.09). This could be caused by the poor quality of the annotated spans for some datasets, or by the loss of contextual information that NLMs distribute over all tokens. Finally, we have performed the same experiments over a BERT (Devlin et al., 2019) and a RoBERTa (Liu et al., 2019) models. The results, available in Appendix C, are very similar to those obtained for the GPT-NeoX models, and they confirm the generalizability of our results.

5.2 Results of the Layer Analysis (Q3)

In this section, we analyse the performance of the different layers of the models we considered, trying to find out which ones are the mostly related to figurative language knowledge (**Q3**).

First, we focus on the last checkpoint for each figure of speech and the results are available in Table 1. Analysing which layers perform best (reported in the BL column), we can see that in most cases **the best layers are in the central part or in the second half of the model**. For instance, the best performing layer of $1.4B$ (which has a total of 24 layers) for Hyperbole is layer 13 and for $6.9B$ is layer 18 over 32. The same can be said for smaller models, such as $160M$ (12 layers). In this case, the best layers are 8 for Hyperbole, 9 for Metaphor and 6 for Oxymoron. These results are consistent with the literature. In fact, there is a general intuition that the lower layers of Transformer-based architectures store more syntactic and grammatical knowledge, the intermediate layers have more semantics-related knowledge, while the last layers are more task oriented (Miaschi et al., 2020; Fayyaz et al., 2021). An evident improvement with respect to the last layer can be seen for Hyperbole, since $31M$, $70M$ and $160M$ have an improvement of more than 20%. For Metaphor, the best performing models ($410M$, $1.4B$ and $6.9B$) show a slighter improvement of their performance, with respectively a 3.5%, 1.9% and 5.1% increase. For Pleonasm there is no particular improvement, and even the best performing layers are not able to recognise a pleonasm in a sentence.

For Oxymoron, we have a more peculiar situation. In fact, for the last layer, as we discussed in Section 5.1 and shown in Figure 1, all models except $410M$, $1.4B$ and $6.9B$ (which still obtains results comparable to $31M$ for the last layer) suffer a strong decrease in performance in the latter stages of the learning trajectory. However, the results in Table 1 indicate that the best performing layers have much better performance compared to the last, with an increase of 91.0% for $31M$, 87.8% for $70M$ and an impressive 206.8% for $160M$. Analysing the complete learning trajectory of the GPT-NeoX models, shown in Figure 2, we can see that there is basically no evident worsening of the performance over the training, and all models except $14M$ (which shows a gradual decrease in terms of compressions) are stable or even show a slight increase. For instance, the $1.4B$ model reaches 7.48 at the final checkpoint with respect to 5.82 at $10B$ tokens and 6.82 at $100B$ tokens. This phenomenon is consistent with recent findings (Rogers et al., 2020; Wallat et al., 2020; Haviv et al., 2023), which demonstrate that Transformer models predominantly retrieve memorised information in the intermediate layers. This has also been confirmed by the work in Tirumala et al. (2022) which, focusing on the last layer of NLMs, analyses how they may forget information during training. They also point out that this effect is more pronounced in smaller models, whereas larger models tend to maintain higher levels of memorisation during training. This is consistent with our results for Oxymoron (Figure 1). Moreover, our layer analysis suggests that information regarding oxymorons may not be forgotten but instead stored in intermediate layers. Therefore, we claim that this phenomenon cannot be categorized as a catastrophic forgetting (Thompson et al., 2019) of the whole model. Instead, as shown by (Wallat et al., 2020), we claim that the last layer becomes more focused on the language modeling training task, rather than storing semantic knowledge, which can be found elsewhere in the model. Further experimental investigation is necessary to better characterise and quantify these effects.

5.3 Impact of Training Data (Q4)

To analyse the impact of pre-training data on the model's ability to detect figures of speech (**Q4**), we chose to train two GPT-NeoX models from scratch, with $70M$ and $160M$ parameters respec-

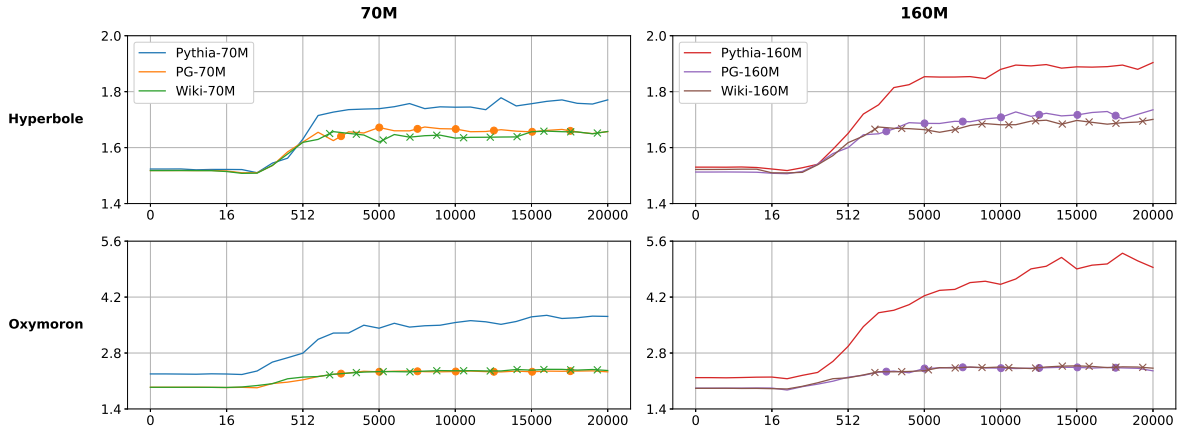

Figure 3: Learning trajectories of the best layer for the GPT-NeoX models trained from scratch. For each figure of speech, we report the compression value for the first $20K$ iterations. Each line represents a different model: in blue Pythia-$70M$, in orange PG-$70M$, in green Wiki-$70M$, in red Pythia-$160M$, in purple PG-$160M$, in brown Wiki-$160M$; dots for PG models and crosses for Wiki models represent the termination of an epoch on the dataset.

tively, on two small English datasets, provided by Dolma (Soldaini et al., 2024). The first is *Project Gutenberg*, a corpus containing approximately $55K$ books (13GB of text, for a total of $5.3B$ tokens), which we hypothesised should contain a more literary language and, therefore, more figures of speech. The second one is a generic corpus, i.e. a Wikipedia dump made by 6 million pages (11GB of text, for a total of $3.8B$ tokens).

To train the models, we follow the procedure described in Biderman et al. (2023), without changing any hyperparameters. We train the architecture with $41B$ tokens ($20K$ iterations of $2M$ tokens) considering both the smaller datasets involved and the performance reported in Figure 1, which does not show any visible improvement after that threshold. Given that both datasets contain a lower number of tokens, the training was continued for several epochs (7 for Project Gutenberg and 11 for Wikipedia). After training, we performed the same linguistic evaluation presented in Biderman et al. (2023). Results are reported in Appendix D.

In Figure 3, we report the learning trajectories of the $70M$ and $160M$ models trained on the original dataset (*Pythia-70* and *Pythia-160*), the Project Gutenberg collection (*PG-70* and *PG-160)* and Wikipedia (*Wiki-70* and *Wiki-160)*. With respect to the previous experiments, we report only the results for the two best performing figures of speech, Hyperbole and Oxymoron. The complete results are available in Appendix C. Focusing on the comparison between the PG and Wiki models, we observe that the PG-70 and PG-160 trained on books, presumably richer in figures of speech, do not outperform the respective Wiki models. For Hyperbole, PG and Wiki models stabilise around very similar compression values, 1.66 for the PG-70 and Wiki-70 models, and 1.72 for PG-160 and Wiki-160. Similarly, for Oxymoron, both PG and Wiki models reach 2.30 compression for $70M$ and $160M$ parameters models. These results suggest that **the more literary language expected from the Project Gutenberg corpus does not confer a measurable advantage over a more generic dataset, in terms of learning figures of speech**.

However, it is evident that the Pythia models consistently achieve superior performance, especially for Hyperbole, 1.77 for $70M$ and 1.90 for $160M$, and Oxymoron, 3.72 for $70M$ and 4.94 for $160M$. A plausible explanation is the dataset size and epoch structure. In fact, the original The Pile dataset is definitely larger w.r.t. PG and Wiki. Therefore, whereas The Pile provides continuous novel data throughout the training process, PG and Wiki force the training to loop over the same data multiple times. This contributes to the stabilisation of the learning trajectories after the first epoch. These findings are also supported by the linguistic evaluations in the Appendix D, which confirms the consistent advantage of training with The Pile. For Pleonasm and Metaphor, no notable performance differences were observed between PG and Wiki.

6 Conclusions and Future Work

Considering seven different GPT-NeoX models, we investigated whether and how NLMs are able

to recognise four types of figurative language (hyperboles, metaphors, oxymorons and pleonasms) describing their learning trajectories. Our results show that the most recognisable figures of speech are hyperboles and oxymorons, whereas more difficulties can be found for metaphors and especially for pleonasms. Analysing the learning trajectories, we discovered that such capabilities are acquired in the early stages of the training and usually do not improve in the latter stages (Q1). On the contrary, we observed that for Oxymoron the final layer of several models decreases its results, coherently with Tirumala et al. (2022). Rogers et al. (2020) and Haviv et al. (2023). Apart from the $6.9B$ model, bigger models tend to better understand the figures of speech we analysed (Q2). Analysing the different layers of each model, we have seen that best performing ones are usually in the central part or in the second half of the model, consistently with the literature (Fayyaz et al., 2021) (Q3). By comparing models trained on different datasets, we saw that using a literary corpus does not provide advantages in recognising figures of speech, with respect to a model trained on generic content. Moreover, a larger general dataset drastically improves the results, coherently with Longpre et al. (2024) and Zhao et al. (2024b) (Q4).

As future work, we aim to expand our analysis considering more figures of speech, more datasets and their impact, considering also other language and multilingual models. For instance, it would be interesting to identify whether figures of speech knowledge is language specific or shared by more languages (Zhao et al., 2024c). Finally, we will focus on LLMs and black box models.

Limitations

The study presented in this paper is a first step towards a more detailed comprehension on how NLMs understand figurative language and as such it has some limitations.

First, since our method leverages the model embeddings, not all models can be analysed through our approach. LLMs are released in three main ways: *open source*, where both the original weights and the model technical specification (such as its underlying architecture, the training algorithm, and often the datasets used for training), are publicly released; *open/restricted weight*, where only the original weights are available, in some cases under certain conditions; *closed source*, where nothing is available, except for a way (usually a web interface or a set of APIs) to communicate with the LLM. Our method is applicable to the first type of released models, as long as the checkpoints are available or there are enough computational resources to reproduce them; it is applicable also for the second type but only if the checkpoints are released. Our method is not applicable to closed source models.

The second main limitation is intrinsic to probing tasks and how they interact with generative LLMs. Probing tasks allow us to investigate whether and how a certain type of knowledge is contained in the model embeddings. However, what a probing task cannot capture is how this information is used in the text generation process.

Other limitations are related to the datasets and the models considered in our analysis. For the datasets, we only considered one dataset per figure of speech containing about $1.5K$-$3K$ sentences. A more in-depth study of the phenomenon could be conducted considering several datasets of different sizes. Moreover, all the data we considered are in English. However, apart from some few examples (such as the TroFi dataset for non literal language by Birke and Sarkar (2006)) such resources are quite difficult to find or create, especially for less known types of figurative language and for other languages. For the models, we considered only a single family of GPT-based models (the GPT-NeoX benchmark models released by Biderman et al. (2023)). We are aware that several different open-source LLMs are available for testing, also bigger and more powerful ones.

Ethics and Impact Statement

This paper investigates the capabilities of Neural Language Models in the overall field of semantics and, more specifically, to recognise figurative language with probing tasks. Although we expect that this type of studies might help understanding the behaviour of NLMs, in order to improve their factuality and overall coherence, interpretability studies might also help in understanding the mechanics behind how models generate misleading, abusive or biased language. This knowledge of how models work might be used in order to expand those unwanted aspects of the generated text in a harmful way.

Although the calculation of the learning trajectories is based on simple feed-forward neural network classifiers, the overall combination of retrieving

the embeddings, the multiple training processes required by MDL (Voita and Titov, 2020) and the studies among checkpoints and layers require extensive computational time on a single DGX-A100 server involving a 40GB MIG of a single GPU. Under this setting, probing all the checkpoints with the MDL method could take from 2 hours, for the fastest model, to 51 hours for the slowest one. Given that MDL guarantees stability of the results across multiple seeds (Voita and Titov, 2020), our results were obtained over a single run. Overall, the greatest impact on the environment is definitely from training the two GPT-NeoX models on different datasets. Although the training was truncated at the $20K$ iteration, the resource consumption is considerable and can be measured in about 700 hours. Total emissions, estimated with the Machine Learning CO2 Impact Tool (Lacoste et al., 2019), are 75.6 kgCO_2eq.

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. This research also utilized resources from the AI4WATER project ("Optimizing Water Resources in Coastal Areas using Artificial Intelligence") that is part of the PRIMA Programme supported by the European Union and by the Italian Ministry of University and Research.

References

Ehsan Aghazadeh, Mohsen Fayyaz, and Yadollah Yaghoobzadeh. 2022. Metaphors in pre-trained language models: Probing and generalization across datasets and languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2037–2050, Dublin, Ireland. Association for Computational Linguistics.

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

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

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Julia Birke and Anoop Sarkar. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 329–336, Trento, Italy. Association for Computational Linguistics.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.

Marianna M. Bolognesi, Claudia Roberta Combei, Marta La Pietra, and Francesca Masini. 2024. What makes an awfully good oxymoron? *Language and Cognition*, 16(1):242–262.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Christian Burgers, Britta C Brugman, Kiki Y Renardel de Lavalette, and Gerard J Steen. 2016. Hip: A method for linguistic hyperbole identification in discourse. *Metaphor and Symbol*, 31(3):163–178.

Tommaso Caselli, Irene Dini, and Felice Dell'Orletta. 2022. How about time? probing a multilingual language model for temporal relations. In *Proceedings*

of the 29th International Conference on Computational Linguistics, pages 3197–3209, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Cheng-Han Chiang, Sung-Feng Huang, and Hung-yi Lee. 2020. Pretrained language model embryology: The birth of ALBERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6813–6828, Online. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

Verna Dankers, Christopher Lucas, and Ivan Titov. 2022. Can transformer be too compositional? analysing idiom processing in neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3608–3626, Dublin, Ireland. Association for Computational Linguistics.

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

Mohsen Fayyaz, Ehsan Aghazadeh, Ali Modarressi, Hosein Mohebbi, and Mohammad Taher Pilehvar. 2021. Not all models localize linguistic knowledge in the same place: A layer-wise probing on BERToids' representations. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 375–388, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Marcos Garcia, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2021. Probing for idiomaticity in vector space models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3551–3564, Online. Association for Computational Linguistics.

Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. 2015. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21, Lisbon, Portugal. Association for Computational Linguistics.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *Trans. Mach. Learn. Res.*, 2023.

Adi Haviv, Ido Cohen, Jacob Gidron, Roei Schuster, Yoav Goldberg, and Mor Geva. 2023. Understanding transformer memorization recall through idioms. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 248–264, Dubrovnik, Croatia. Association for Computational Linguistics.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online. Association for Computational Linguistics.

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

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

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Omid Kashefi, Andrew T. Lucas, and Rebecca Hwa. 2018. Semantic pleonasm detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 225–230, New Orleans, Louisiana. Association for Computational Linguistics.

Arne Köhn. 2015. What's in an embedding? analyzing word embeddings through multilingual evaluation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073, Lisbon, Portugal. Association for Computational Linguistics.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.

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

Christian Lehmann. 2005. *Pleonasm and hypercharacterisation*, pages 119–154. Springer Netherlands, Dordrecht.

Hector J. Levesque. 2011. The winograd schema challenge. In *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*. AAAI.

Emmy Liu and Graham Neubig. 2022. Are representations built from the ground up? an empirical examination of local composition in language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9053–9073, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3622–3628. ijcai.org.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Zeyu Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith. 2021. Probing across time: What does RoBERTa know and when? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 820–842, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2024. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3245–3276, Mexico City, Mexico. Association for Computational Linguistics.

Alessio Miaschi, Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. 2020. Linguistic profiling of a neural language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 745–756, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Michael Mohler, Mary Brunson, Bryan Rink, and Marc Tomlinson. 2016. Introducing the LCC metaphor datasets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4221–4227, Portorož, Slovenia. European Language Resources Association (ELRA).

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 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.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about

how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. 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 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

Nina Schneidermann, Daniel Hershcovich, and Bolette Pedersen. 2023. Probing for hyperbole in pre-trained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 200–211, Toronto, Canada. Association for Computational Linguistics.

Ekaterina V Shutova. 2011. Computational approaches to figurative language. Technical report, University of Cambridge, Computer Laboratory.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, and 17 others. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Carlo Strapparava. 2018. Metaphor: A computational perspective by tony veale, ekaterina Shutova and beata beigman klebanov. *Computational Linguistics*, 44(1):191–192.

Minghuan Tan and Jing Jiang. 2021. Does BERT understand idioms? a probing-based empirical study of BERT encodings of idioms. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1397–1407, Held Online. INCOMA Ltd.

Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. 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 2062–2068, Minneapolis, Minnesota. Association for Computational Linguistics.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 38274–38290. Curran Associates, Inc.

Enrica Troiano, Carlo Strapparava, Gözde Özbal, and Serra Sinem Tekiroğlu. 2018. A computational exploration of exaggeration. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3296–3304, Brussels, Belgium. Association for Computational Linguistics.

Bram van Dijk, Tom Kouwenhoven, Marco Spruit, and Max Johannes van Duijn. 2023. Large language models: The need for nuance in current debates and a pragmatic perspective on understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12641–12654, Singapore. Association for Computational Linguistics.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.

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

Jonas Wallat, Jaspreet Singh, and Avishek Anand. 2020. BERTnesia: Investigating the capture and forgetting of knowledge in BERT. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 174–183, Online. Association for Computational Linguistics.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Fan Xu, Ziyun Zhu, and Xiaojun Wan. 2023. Creative destruction: Can language models interpret oxymorons? In *Natural Language Processing and Chinese Computing - 12th National CCF Conference, NLPCC 2023, Foshan, China, October 12-15, 2023, Proceedings, Part I*, volume 14302 of *Lecture Notes in Computer Science*, pages 645–656. Springer.

Kelly Zhang and Samuel Bowman. 2018. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361, Brussels, Belgium. Association for Computational Linguistics.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024a. Explainability for large language models: A survey. *ACM Trans. Intell. Syst. Technol.*, 15(2).

Yang Zhao, Li Du, Xiao Ding, Kai Xiong, Zhouhao Sun, Shi Jun, Ting Liu, and Bing Qin. 2024b. Deciphering the impact of pretraining data on large language models through machine unlearning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9386–9406, Bangkok, Thailand. Association for Computational Linguistics.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024c. How do large language models handle multilingualism? In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

A Additional Dataset Details

A.1 Generation of the Negative Instances for the Oxymoron Dataset

As we explained in Section 3, to generate the negative examples for the oxymorons dataset we leverage the official OpenAI API, in particular, the Chat Completion API with the gpt-4-turbo-2024-04-09 model snapshot. First, we split each oxymoron into single words and then we ask GPT-4 to produce a single sentence with the literal meaning of each word. For every oxymoron, we pass to GPT-4 two different prompts: the *system prompt*, containing the general instructions and the behaviour it has to follow, and the *user prompt*, which contains the list of words that must appear in the generated sentence. In the following, we report the system prompt:

> Given a list of words you need to create a single sentence that includes all the listed words, using each of them in their most direct and literal sense. The sentence should be coherent and contextually sensible, showcasing the literal meaning of each word without relying on figurative language or idiomatic expressions.
> Example:
> Words: book, light, plant
> Literal Sentence:
> She read her book under natural light next to the window where her favourite plant was placed.

In the first part of the prompt, we explain to the model which task it has to solve, which input it has to expect (a list of words) and how it has to respond (with a single sentence). In the second part, to reinforce the instructions we gave, we provide a simple example (one-shot) of some words and how they are used literally in a sentence.

The results were validated by two of the authors (a PhD student and a post-doc researcher both male under 40 from Europe) who simply checked if the sentences generated contained an oxymoron (in that case, such a sentence would be excluded) or not.

A.2 Licenses and Terms of Use

In the following, we discuss the licenses and terms of use for the datasets and the models we used.

Hyperbole The dataset is freely distributed under the Creative Commons License (Troiano et al., 2018).

Metaphor The authors of LCC (Mohler et al., 2016) did not set a specific license, but they state in the paper that their dataset can be used at no cost for research purposes.

Oxymoron As for Metaphors, the authors did not specify a license or terms of use but they state in Xu et al. (2023) that the oxymorons they collected are publicly available. Our dataset, containing also the negative instances we generated, will be released (upon acceptance) under Creative Commons License.

Pleonasm The dataset is released under the terms of GNU General Public License without any warranty by Kashefi et al. (2018).

GPT-NeoX This architecture is released under the Apache License 2.0 by Black et al. (2022)[4].

Pythia These benchmark models are released under the Apache 2.0 license by Biderman et al. (2023)[5].

Wikipedia and Project Gutenberg These datasets are taken from Dolma (Soldaini et al., 2024) under the ODC-BY licence[6].

B Model Hyperparameters and Implementation Details

In Table 2 we report the dimensions and the characteristics of the models considered in terms of number of layers, number of heads per layer and embedding dimension. The implementation was made in Python 3.10.13, in particular exploiting the following packages: numpy 1.26.3, scikit-learn 1.0.2, pandas 2.1.4, torch 2.1.2 and transformers 4.37.1. For the MDL method, our code is based on the one released by Voita and Titov (2020), available at https://github.com/lena-voita/description-length-probing.

C Additional Results

In Figure 4, we show the learning trajectories, in terms of compression, for our four figures of speech, considering the best performing layer of the GPT-NeoX models. Results in terms of accuracy are available in Figure 5 (considering the last layer) and Figure 6 (considering the best layer).

In Figure 7, we show the learning trajectories for the last layer of the $410M$ model considering two different inputs for the probe: (i) the full sentence and (ii) only the portion of tokens (span) containing the figure of speech.

More detailed results for the analysis of the impact of the training data are in Figures 8, 9, 10, 11. More specifically, in Figure 8, we show the learning trajectories for the best layer, in terms of compression, for our four figures of speech, for the $70M$ and $160M$ trained on the original dataset (Pythia), the Project Gutenberg data (PG) and the Wikipedia dump (Wiki). Similarly, in Figure 9 we show the same results but considering the last layer. Figures 10 and 11 show the learning trajectories in terms of accuracy respectively considering the best and the last layer.

[4] https://github.com/EleutherAI/gpt-neox
[5] https://github.com/EleutherAI/pythia
[6] https://huggingface.co/datasets/allenai/dolma

Model	Num. Layers	Num. Head	Embed. Dim.
14M	6	4	128
31M	6	8	256
70M	6	8	512
160M	12	12	768
410M	24	16	1024
1.4B	24	16	2048
6.9B	32	32	4096

Table 2: Dimensions of the models considered in terms of number of layers, number of heads per layer and embedding dimension.

In Table 3, we show the comparison of the learning trajectories of the GPT-NeoX models, BERT and RoBERTa, for all four figures of speech. Each column indicates a different training percentage.

In Figure 12 we show the compression (on the y-axis of each plot) of all model layers (on the x-axis), considering their final checkpoint. Each column presents the results for a different figure of speech.

D Models Benchmark Scores

Following Biderman et al. (2023), we evaluate the models presented in Section 5.3 on the same linguistic benchmark: (i) the LAMBADA dataset (Paperno et al., 2016), designed to predict the endings of text passages and testing language prediction skills; (ii) the Physical Interaction Question Answering tasks (PIQA) (Bisk et al., 2020) to test physical commonsense reasoning; (iii) the WinoGrande task (Sakaguchi et al., 2020), a large-scale dataset for coreference resolution; (iv) the Winograd Schema Challenge (WSC) (Levesque, 2011), a test of commonsense reasoning and coreference resolution; (v-vi) the AI2 Reasoning Challenge (ARC) (Clark et al., 2018), with tasks involving complex reasoning over a diverse set of questions sorted in easy and challenging questions; (vii) the Science Question Answering tasks (SciQ) (Welbl et al., 2017) to assess understanding of scientific concepts in multiple-choice format with 4 answer options each; and (viii) the LogiQa (Liu et al., 2020) a set of logical reasoning tasks requiring advanced inference and deduction.

As we can see in Figure 13, our models follow the behaviour of the original model. For WinoGrande, WSC, ARC-Challenge and LogiQA, there are no significant differences in performance, whereas for LAMBADA, PiQA, ARC-Easy, and SciQ the original models perform slightly better than our trained models.

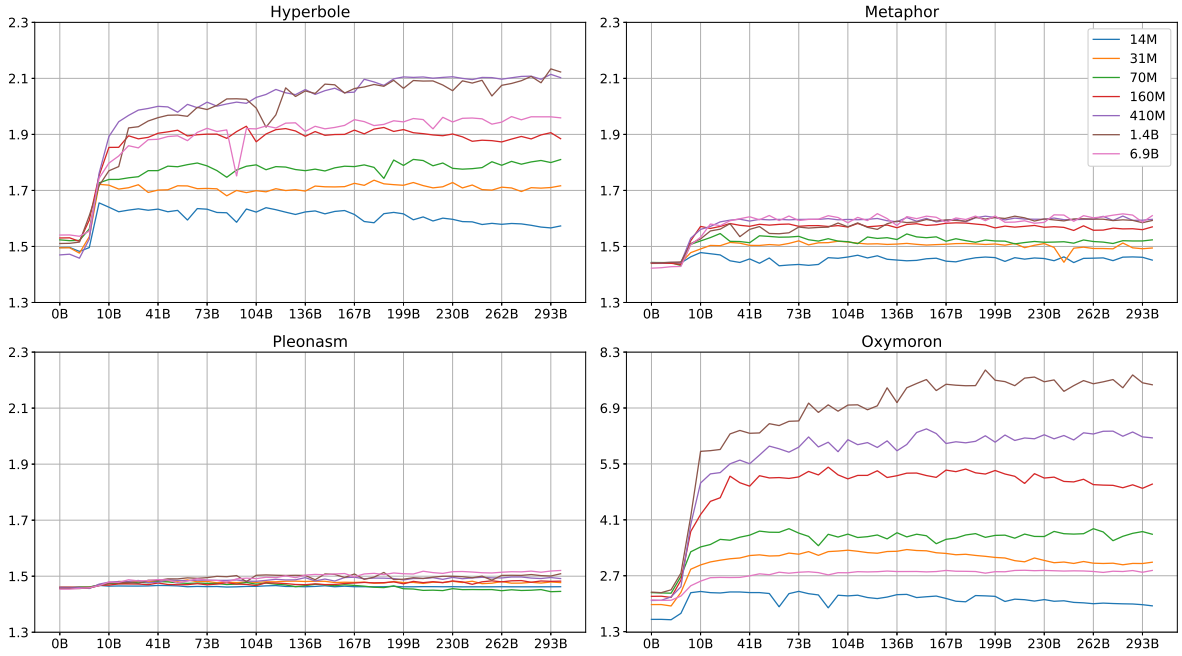

Figure 4: Learning trajectories of the GPT-NeoX models considering the best performing layer. Each plot represents a different figure of speech. On the x-axis, we indicate the number of tokens used for training the model; on the y-axis we indicate the compression calculated using MDL. Each line represents a different model: in blue, the one with $14M$ parameters, in orange with $31M$, in green with $70M$, in red with $160M$, in purple with $410M$, in brown with $1.4B$ and in pink with $6.9B$.

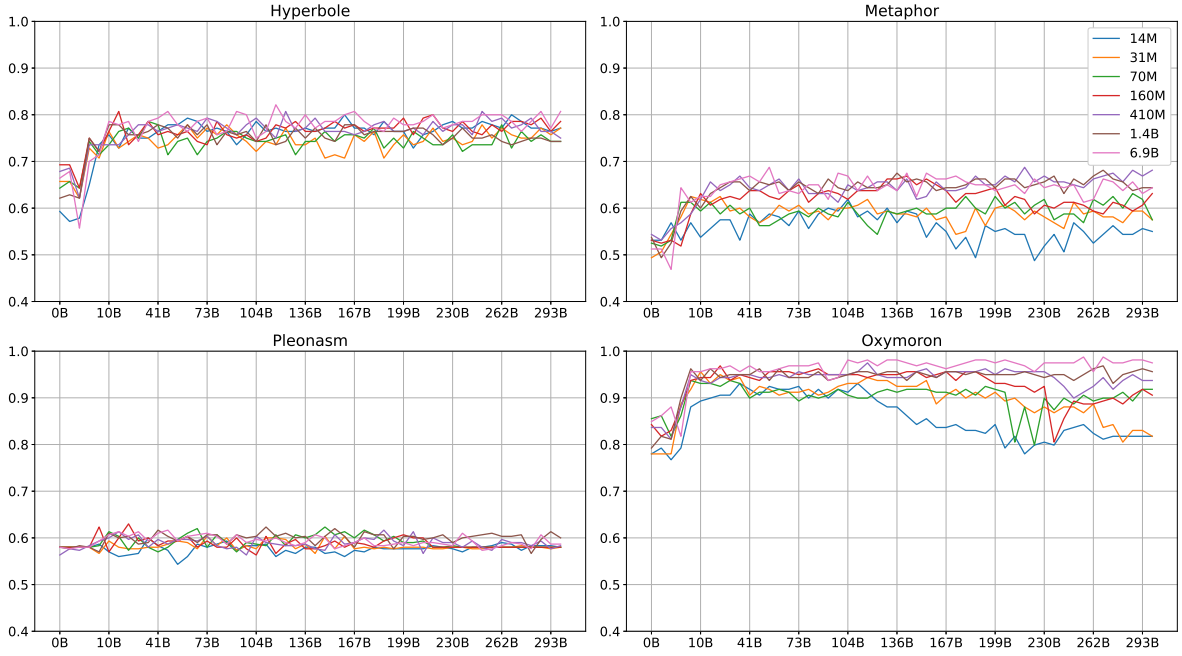

Figure 5: Learning trajectories of the GPT-NeoX models considering the last layer in terms of accuracy. Each plot represents a different figure of speech. On the x-axis, we indicate the number of tokens used for training the model; on the y-axis we indicate the accuracy. Each line represents a different model: in blue, the one with $14M$ parameters, in orange with $31M$, in green with $70M$, in red with $160M$, in purple with $410M$, in brown with $1.4B$ and in pink with $6.9B$.

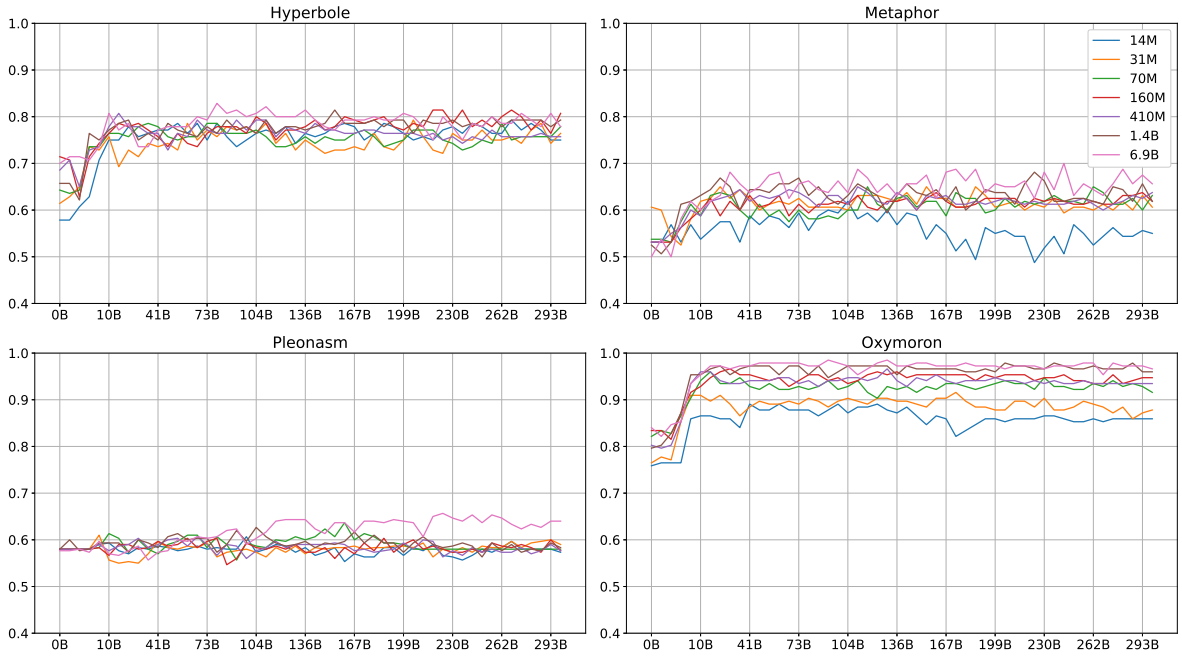

Figure 6: Learning trajectories of the GPT-NeoX models considering the best layer in terms of accuracy. Each plot represents a different figure of speech. On the x-axis, we indicate the number of tokens used for training the model; on the y-axis we indicate the accuracy. Each line represents a different model: in blue, the one with $14M$ parameters, in orange with $31M$, in green with $70M$, in red with $160M$, in purple with $410M$, in brown with $1.4B$ and in pink with $6.9B$.

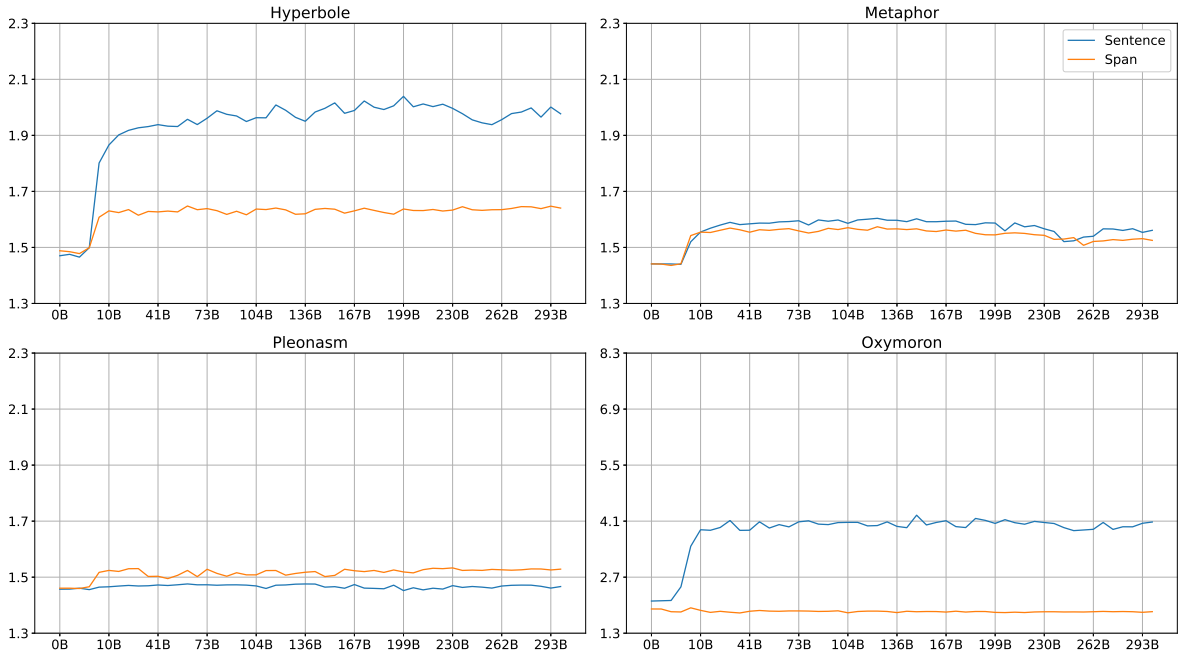

Figure 7: Learning trajectories comparison between a $410M$ GPT-NeoX model with different probing strategies: (i) Sentence, where the probe is fed with the sentence representation, and (ii) Span, where we give to the probe only the portion of the sentence containing the figure of speech. Each plot represents a different figure of speech. On the x-axis, we indicate the number of tokens used for training the model; on the y-axis we indicate the compression.

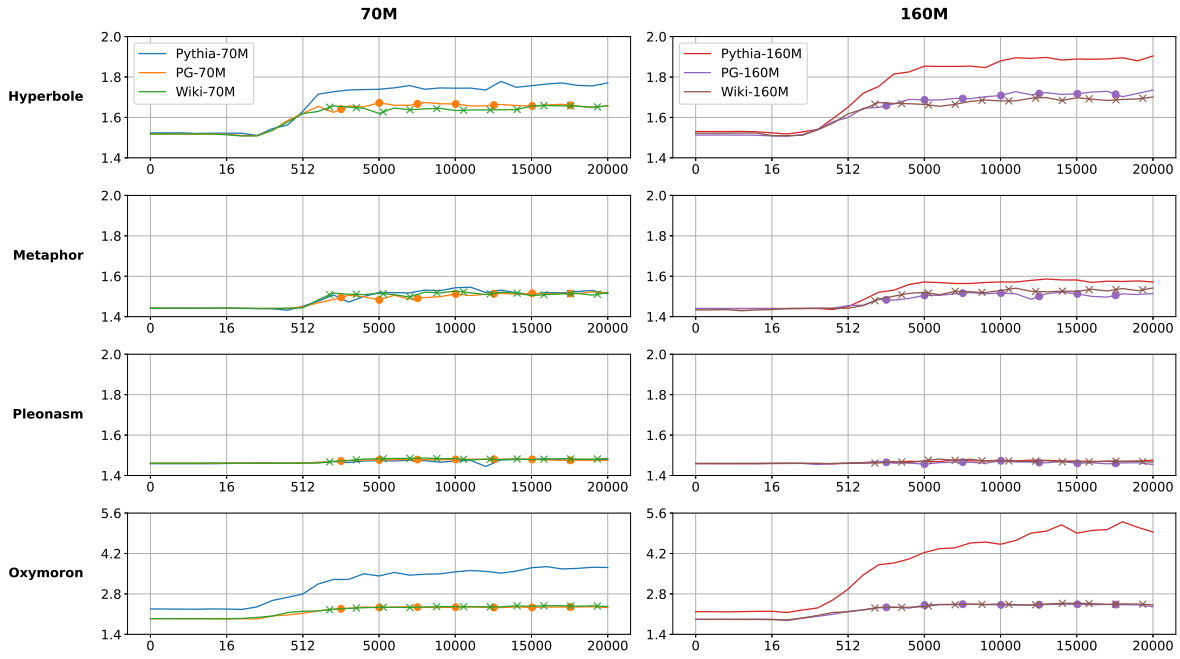

Figure 8: Learning trajectories of the best layer for the GPT-NeoX models trained from scratch. On the left column we show the $70M$ models, whereas on the right column the $160M$ models. For each task we report the compression value for the first $20K$ iterations. Each line represents a different model: in blue the original $70M$ model; in orange the $70M$ model trained on the Project Gutenberg dataset; in green the $70M$ model trained on Wikipedia; in red the original $160M$ model; in purple the $160M$ model trained on Project Gutenberg; in brown the $160M$ model trained on Wikipedia. The dot in the PG model lines and the cross in the Wiki model lines represent the completion of an epoch.

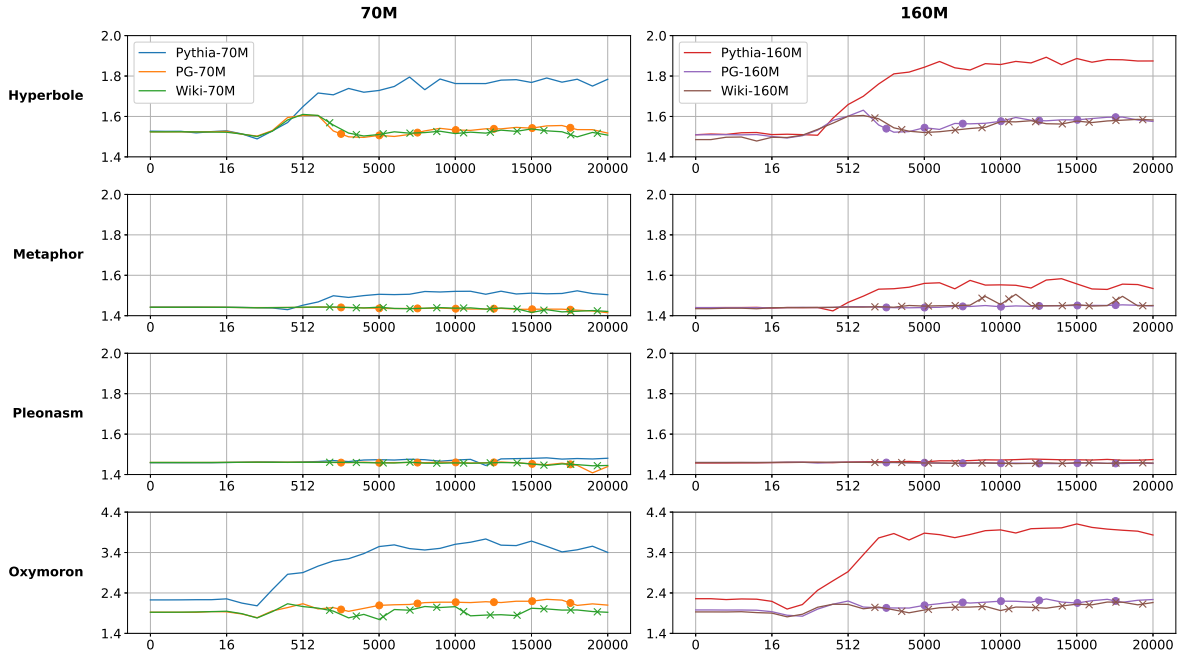

Figure 9: Learning trajectories of the last layer for the GPT-NeoX models trained from scratch. On the left column we show the $70M$ models, whereas on the right column the $160M$ models. For each task we report the compression value for the first $20K$ iterations. Each line represents a different model: in blue the original $70M$ model; in orange the $70M$ model trained on the Project Gutenberg dataset; in green the $70M$ model trained on Wikipedia; in red the original $160M$ model; in purple the $160M$ model trained on Project Gutenberg; in brown the $160M$ model trained on Wikipedia. The dot in the PG model lines and the cross in the Wiki model lines represent the completion of an epoch.

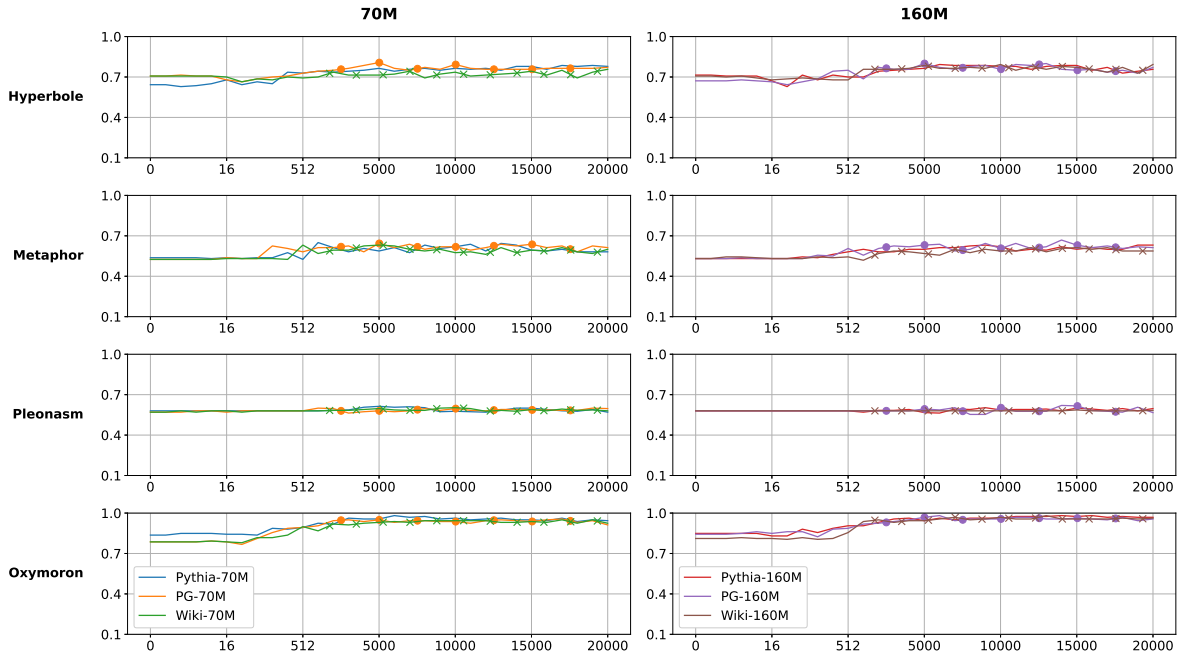

Figure 10: Learning trajectories of the best layer for the GPT-NeoX models trained from scratch. On the left column we show the $70M$ models, whereas on the right column the $160M$ models. For each task we report the accuracy for the first $20K$ iterations. Each line represents a different model: in blue the original $70M$ model; in orange the $70M$ model trained on the Project Gutenberg dataset; in green the $70M$ model trained on Wikipedia; in red the original $160M$ model; in purple the $160M$ model trained on Project Gutenberg; in brown the $160M$ model trained on Wikipedia. The dot in the PG model lines and the cross in the Wiki model lines represent the completion of an epoch.

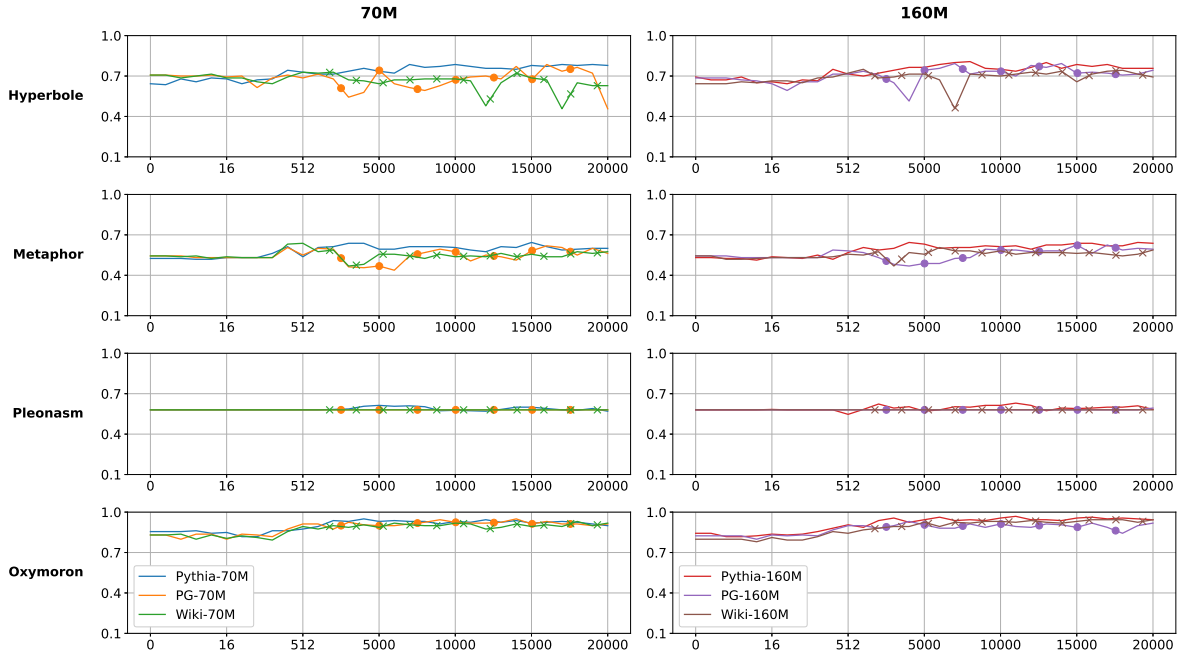

Figure 11: Learning trajectories of the last layer for the GPT-NeoX models trained from scratch. On the left column we show the $70M$ models, whereas on the right column the $160M$ models. For each task we report the accuracy for the first $20K$ iterations. Each line represents a different model: in blue the original $70M$ model; in orange the $70M$ model trained on the Project Gutenberg dataset; in green the $70M$ model trained on Wikipedia; in red the original $160M$ model; in purple the $160M$ model trained on Project Gutenberg; in brown the $160M$ model trained on Wikipedia. The dot in the PG model lines and the cross in the Wiki model lines represent the completion of an epoch.

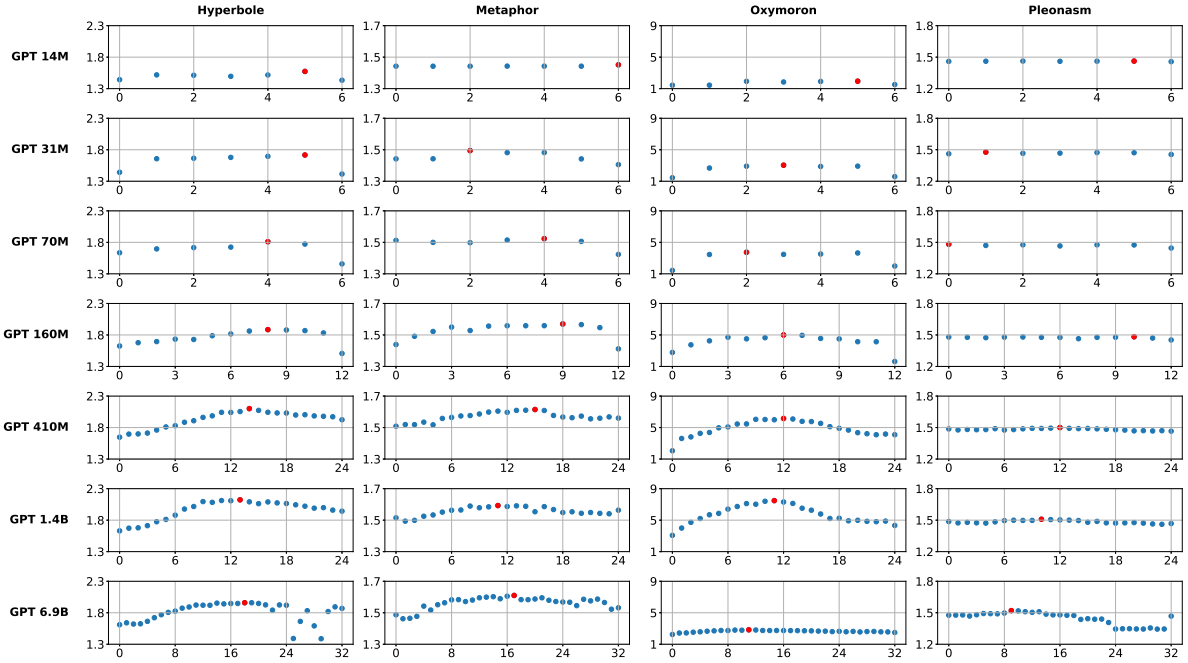

Figure 12: Compression on the final checkpoint calculated for each layer (on the x-axis of each plot) of all the considered models. For each model (grouped in rows) the figure shows four different plots, one per figure of speech analysed. The best layer is highlighted in red.

	Model	Last Layer								Best Layer							
		2%	4%	6%	8%	10%	20%	50%	100%	2%	4%	6%	8%	10%	20%	50%	100%
Hyperbole	BERT	**1.81**	1.79	1.82	1.83	1.81	1.84	1.86	1.93	1.81	1.79	1.83	1.83	1.86	1.91	1.94	2.03
	RoBERTa	1.73	1.78	1.83	1.82	1.84	1.86	1.87	1.87	1.72	1.78	1.83	1.82	1.83	1.92	1.96	2.04
	GPT 14M	1.66	1.64	1.63	1.64	1.63	1.6	1.62	1.57	1.65	1.64	1.63	1.63	1.63	1.59	1.63	1.57
	GPT 31M	1.69	1.72	1.68	1.69	1.70	1.70	1.71	1.71	1.71	1.72	1.69	1.71	1.72	1.72	1.71	1.72
	GPT 70M	1.74	1.75	1.79	1.76	1.78	1.79	1.77	1.78	1.74	1.75	1.75	1.75	1.75	1.79	1.78	1.81
	GPT 160M	**1.81**	**1.87**	1.86	1.87	1.86	1.86	1.89	1.84	1.81	1.85	1.85	1.90	1.88	1.89	1.91	1.88
	GPT 410M	**1.81**	**1.87**	**1.92**	**1.92**	**1.93**	**1.96**	**1.99**	**1.98**	**1.82**	**1.89**	**1.94**	**1.97**	**1.99**	**2.01**	**2.06**	2.10
	GPT 1.4B	1.75	1.79	1.81	1.90	1.91	1.93	1.90	1.94	1.75	1.79	1.86	1.92	1.93	1.96	1.98	**2.12**
	GPT 6.9B	1.73	1.78	1.79	1.80	1.84	1.80	1.88	1.87	1.76	1.82	1.83	1.86	1.85	1.88	1.91	1.96
Metaphor	BERT	**1.55**	1.56	1.54	1.56	1.56	1.55	1.55	1.55	1.54	1.53	1.55	1.54	1.55	1.53	1.56	1.57
	RoBERTa	1.53	1.55	**1.57**	1.57	1.58	1.57	1.55	**1.56**	1.53	1.55	**1.58**	1.57	1.56	1.59	1.58	1.60
	GPT 14M	1.47	1.48	1.47	1.47	1.45	1.43	1.45	1.45	1.47	1.48	1.47	1.47	1.45	1.43	1.45	1.45
	GPT 31M	1.47	1.49	1.51	1.49	1.50	1.49	1.49	1.41	1.49	1.49	1.50	1.50	1.51	1.50	1.51	1.49
	GPT 70M	1.49	1.50	1.52	1.52	1.51	1.51	1.51	1.42	1.47	1.52	1.53	1.55	1.52	1.53	1.53	1.52
	GPT 160M	1.53	1.56	1.55	1.55	1.58	1.54	**1.58**	1.41	1.53	**1.57**	1.57	1.57	1.58	1.58	1.58	1.57
	GPT 410M	1.54	**1.57**	1.55	**1.58**	**1.59**	**1.59**	**1.58**	**1.56**	**1.55**	**1.57**	1.57	**1.59**	**1.59**	**1.60**	1.59	1.60
	GPT 1.4B	1.50	1.55	1.56	1.56	1.56	1.58	1.57	**1.56**	1.52	1.56	1.56	1.56	1.58	1.55	1.59	1.59
	GPT 6.9B	1.51	1.55	**1.57**	1.57	1.57	1.57	**1.58**	1.53	1.50	**1.57**	1.57	1.57	**1.59**	1.59	**1.61**	**1.61**
Oxymoron	BERT	3.55	3.80	3.91	3.88	4.01	4.29	4.06	4.23	3.81	3.93	4.30	4.49	4.13	4.89	5.02	5.68
	RoBERTa	3.52	**4.25**	4.14	4.17	4.37	**4.39**	4.25	3.81	3.91	4.83	4.69	4.86	4.95	5.25	5.80	6.05
	GPT 14M	2.56	2.58	2.67	2.74	2.80	2.61	1.80	1.52	2.27	2.27	2.24	2.27	2.29	1.92	2.25	1.95
	GPT 31M	3.09	3.10	3.05	3.16	3.09	3.22	2.94	1.59	2.87	3.00	3.03	3.09	3.13	3.20	3.31	3.04
	GPT 70M	3.25	3.59	3.50	3.65	3.57	3.55	3.38	1.99	3.30	3.54	3.50	3.61	3.59	3.79	3.61	3.74
	GPT 160M	**3.87**	3.84	3.94	3.88	4.01	3.92	4.22	1.63	3.87	4.37	4.60	4.65	5.19	5.16	5.35	4.99
	GPT 410M	3.78	3.75	3.91	3.96	4.13	4.03	4.13	4.09	4.33	4.94	5.14	5.28	5.50	5.88	6.04	6.15
	GPT 1.4B	3.81	4.21	**4.27**	**4.61**	**4.45**	4.36	**4.63**	**4.33**	**4.99**	**5.46**	**5.48**	**5.86**	**6.25**	**6.46**	**7.33**	**7.48**
	GPT 6.9B	2.37	2.47	2.47	2.46	2.45	2.57	2.57	2.49	2.48	2.62	2.64	2.66	2.66	2.79	2.81	2.83
Pleonasm	BERT	**1.48**	**1.48**	**1.48**	1.47	1.47	**1.48**	1.48	1.47	**1.48**	**1.48**	**1.48**	**1.49**	**1.49**	**1.49**	**1.49**	1.49
	RoBERTa	**1.48**	1.47	**1.48**	1.48	1.48	1.47	1.48	**1.48**	1.47	**1.48**	**1.48**	1.48	1.48	1.48	**1.49**	1.49
	GPT 14M	1.47	1.46	1.46	1.46	1.45	1.46	1.46	1.46	1.47	1.46	1.46	1.46	1.46	1.46	1.46	1.46
	GPT 31M	1.47	1.47	1.46	1.46	1.46	1.46	1.45	1.45	1.47	**1.48**	**1.48**	1.48	1.48	**1.49**	1.48	1.48
	GPT 70M	1.46	1.47	1.47	1.48	**1.48**	1.47	1.47	1.45	1.46	1.47	1.47	1.48	1.48	1.47	1.47	1.45
	GPT 160M	1.46	1.47	1.47	1.47	1.47	1.47	1.47	1.45	1.47	1.47	1.47	1.47	1.47	1.47	1.46	1.48
	GPT 410M	1.47	1.47	1.47	1.47	1.47	**1.48**	1.46	1.47	1.48	**1.48**	**1.48**	1.48	1.48	1.48	**1.49**	1.49
	GPT 1.4B	1.46	1.47	1.47	1.47	1.47	**1.48**	**1.48**	1.47	1.45	**1.48**	1.47	1.48	1.47	**1.49**	**1.49**	**1.51**
	GPT 6.9B	1.47	**1.48**	**1.48**	**1.49**	**1.48**	**1.48**	**1.51**	**1.52**	1.47	**1.48**	**1.48**	**1.49**	1.48	1.48	1.51	1.52

Table 3: Learning trajectory, in terms of compressions, of the GPT-NeoX, BERT and RoBERTa models for all 4 figures of speech considering the last layer (on the left) and the best performing layer (on the right). For each model, the table shows the compression across different training percentages (from 2% to 100%). The best results among the models are highlighted in bold.

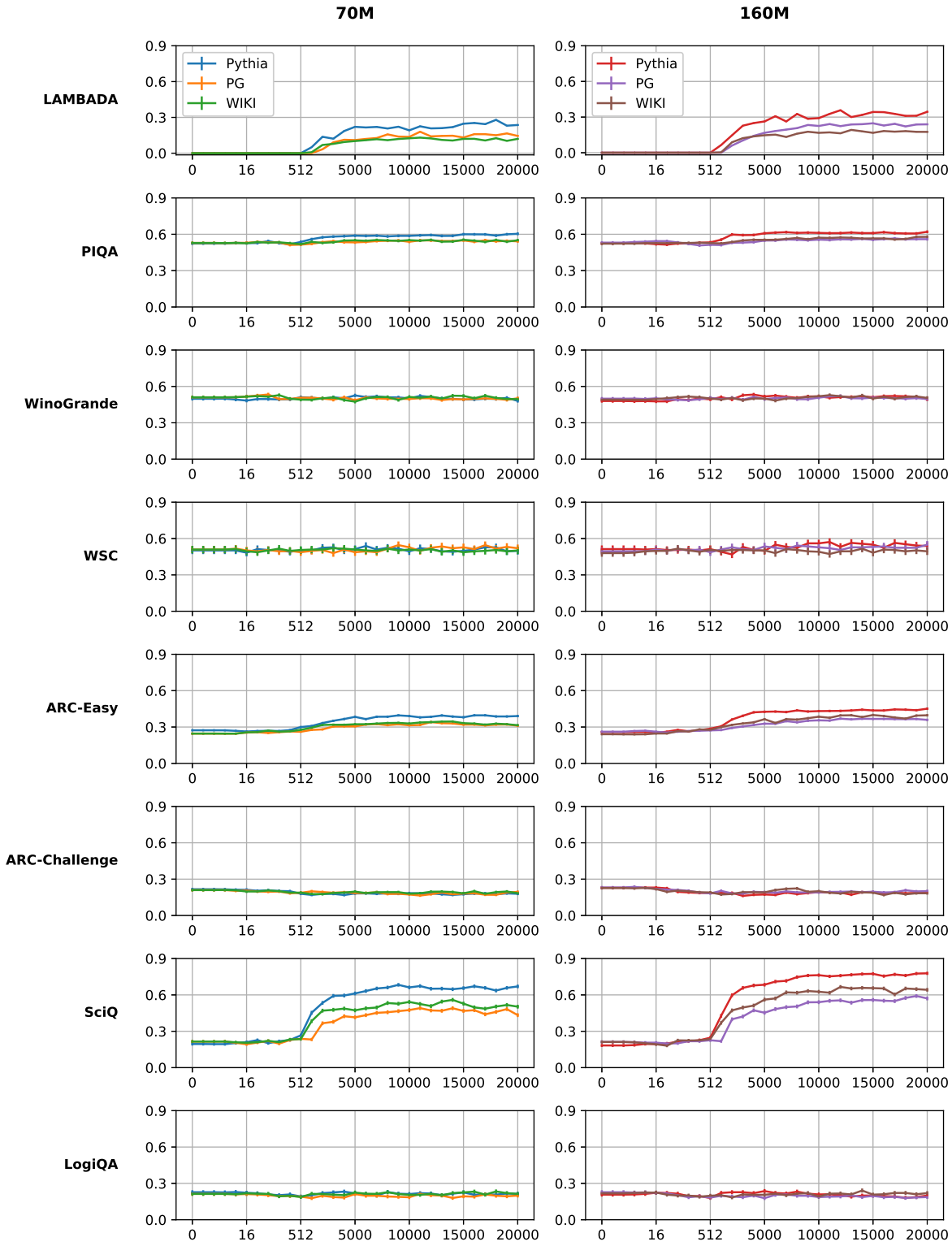

Figure 13: Linguistic benchmark evaluation for our model trained from scratch on specific datasets. On the left column we reported the $70M$ models, whereas on the right column the $160M$ models results; for each task we report the accuracy with the error bar. Each line represents a different model: in blue the original Pythia $70M$ model; in orange the $70M$ model trained on the Project Gutenberg dataset; in green the $70M$ model trained on Wikipedia; in red the original $160M$ model; in purple the $160M$ model trained on Project Gutenberg; in brown the $160M$ model trained on Wikipedia).