How Does BERT Learn?

Diego Zertuche and **Benjamin Liu**
Harvard University
{diego_zertucheserna, tliu}@g.harvard.edu

Abstract

Large language models that trained on large corpora have been shown to be very useful for natural language processing tasks. Given the size of these models, understanding where and when the information is obtained has become a subject of study. Recent work applied several probes to intermediate training stages to observe the developmental process of a large-scale model (Chiang et al., 2020). Building upon this, we probe the intermediate training stages adding a layer by layer analysis to the probes, to better understand where in the model architecture different information is contained. We also build upon (Gururangan et al., 2020) work, by performing task adaptation pretraining to the intermediate training checkpoint models to understand how different levels of pretraining affect the effectiveness of task adaptation procedures. We find that as pretraining keeps going in the BERT model, BERT learns to separate syntactic and semantic knowledge, and pushes syntactic information to the lower layers. Task adaptation procedures are effective from early stages of the pretraining of BERT, achieving similar performances in downstream tasks as the fully trained model. Semantic and commonsense knowledge gets swapped out for the new information contained in the task adaptation corpora when performing task adaptation, while the syntactic information is retained by the BERT model.

1 Introduction

Large language models (LLMs) have been an exciting area of research in the past few years, especially because of the state of the art performance they are achieving in traditionally very difficult tasks. Understanding how these models learn and forget during pre-training and how the learning mechanisms work layer by layer is an important research question that is still not answered. Having more insights in these areas can help improve and build upon these models in a principled way. Our goal is to gain insights into the learning dynamics and forgetting patterns of LLMs, and shed light in the learning mechanisms that take place layer by layer in these models. In specific, we study the linguistic and commonsense knowledge that LLMs contain at different points of the pre-training phase, in a localized manner. We take the BERT models introduced by (Sellam et al., 2021) to perform this analysis and to be robust against parameter initialization. We utilize the edge probing technique introduced in (Tenney et al., 2019b) with the modifications outlined in (Tenney et al., 2019a) to perform the linguistic layer-by-layer analysis. We also probe the models utilizing LAMA tasks introduced by (Petroni et al., 2019a) to understand the quantity and quality of commonsense knowledge embedded in the models at different points of the pre-training procedure.

After having the performance logged of our baseline probing tasks, we perform task adaptation to the models in certain domains (chemistry) to measure the effectiveness of task adaptation at different stages of pre-training and also to quantify how much information is forgotten after adaptation with respect of our baseline probing tasks.

2 Related Works

2.1 Large Language Models

The BERT model (Devlin et al., 2019) has shown state of the art performances in many traditionally difficult tasks, and the context embeddings created with this model are often used in downstream tasks that perform really well. In this work, we focus on the BERT model checkpoints provided by (Sel-

lam et al., 2021), which consists in a series of 25 fully pre-trained BERT models that where initialized with different random seeds, and that were pre-trained in the same way and with the same data as in (Devlin et al., 2019). For 5 of the 25 models, 28 intermediate checkpoints of the pre-training phase are also provided, which can be used to look into the learning dynamics of the BERT models. Having multiple BERT models can allow for statistical hypothesis testing and enables researchers to draw robust and statistically justified conclusions about pre-training procedures.

2.2 Edge Probing

There are several techniques to probe models for their linguistic knowledge. One approach is shown in (Tenney et al., 2019b) work, where they create an 'edge probing' approach that bases itself on creating sub-sentence tasks derived from the traditional structured NLP pipeline. These tasks are structured as labeled graphs anchored to spans in the sentences. The spans act as labeled edges of the graph and the span embeddings are fed into a classifier model that predicts the label of the set of spans. (Tenney et al., 2019a) work builds on this, adding scalar mixing weights to the outputs of all layers, which can provide a way of knowing which layers, in combination, are important when the probing classifier has access to the whole BERT model. This technique was initially introduce in the ELMo model. It introduces scalar parameters γ_τ and $a_\tau^{(0)}, a_\tau^{(1)}, ..., a_\tau^{(L)}$, and we let:

$$\mathbf{h}_{i,\tau} = \gamma_\tau \sum_{\ell=0}^{L} s_\tau^{(\ell)} \mathbf{h}_i^{(\ell)} \tag{1}$$

where $\mathbf{s}_\tau = \text{softmax}(\mathbf{a}_\tau)$. A summary statistic of the layer weights is also introduced, called Center of Gravity, which is define as:

$$\bar{E}_s[\ell] = \sum_{\ell=0}^{L} \ell \cdot s_\tau^{(\ell)} \tag{2}$$

which reflects the average layer attended to for each task. A higher center of gravity (CoG) value means that the information needed for that task is captured by higher layers.

2.3 Commonsense Knowledge

Language models like BERT has been widely used for transfer learning and fine-tuned to complete cross-domain tasks. To understand what knowledge is stored in language models, various knowledge probing techniques have been researched on the language models. In particular, the Language Model Analysis (LAMA) approach (Petroni et al., 2019b) has shown great ability to analyze factual and commonsense knowledge in off-the-shelf BERT model. LAMA probe utilized language data containing factual knowledge that are formatted in cloze-style with single token answers about commonsense relations, such as asking for the capital of a country or the birthplace of a well-known person. Previous works have shown that BERT, without fine-tuning, can retain a certain level of commonsense factual knowledge and some types of knowledge is learned better than others. LAMA probe is considered a benchmark to test the amount of knowledge stored in language models like BERT.

2.4 Task Adaptation

The work from (Gururangan et al., 2020) shows that a second phase of pretraining in-domain (domain-adaptive pretraining) on a different corpora leads to performance gains in downstream tasks that are related to the new corpora, under both high-resource and low-resource settings. Moreover, adapting to the task's unlabeled data (task-adaptive pretraining) improves performance even after domain-adaptive pretraining and by itself. Yet, it has not been established if this task-adaptive pretraining causes information in the original model to be forgotten, or if different levels of pretraining in the original dataset affects performance when performing task-adaptive pretraining.

3 Data and Preprocessing

The data used for the linguistic edge probing are the English Web Treebank (Silveira et al., 2014) for the dependencies task, SemEval 2010 Task 8 dataset (Hendrickx et al., 2009) for relation classification tasks, and the OntoNotes 5.0 dataset (Weischedel et al., 2013) for all other tasks. The data was split into train, validation and test splits and was converted into the edge probing format using modified scripts from the Jiant package (Wang et al., 2019).

LAMA probe uses datasets from 4 knowledge sources: Google-RE[1], T-REx (Elsahar et al., 2018), ConceptNet (Speer and Havasi, 2012), and SQuAD (Rajpurkar et al., 2016). The facts in Google-RE

[1]https://code.google.com/archive/p/relation-extraction-corpus/

come from 3 relations: "place of birth", "date of birth" and "place of death". The T-REx dataset contains facts from 41 Wikipedia relations. The facts from ConceptNet cover 16 relations from Open Mind Common Sense (OMCS), but the relations are not specified in LAMA probe. In the SQuAD dataset, a Wikipedia based question answering dataset, 305 questions were extracted and rephrased into cloze-style questions with single-token answer. Combined, LAMA uses 51,329 commonsense facts in cloze style from these 4 data sources. Detailed statistics are summarized in Table 1.

Name	# Relations	# Facts
Google-RE	3	5527
T-REx	41	34039
ConceptNet	\	11458
SQuAD	\	305

Table 1: Number of relations and facts in each of the data sources in LAMA probe

For the task adaptation dataset, we utilize the corpus from the ChemProt dataset (Krallinger et al., 2017). This dataset consists of text exhaustively annotated by hand with mentions of chemical compounds/drugs and genes/proteins, as well as 22 different types of compound-protein relations. We use the unlabeled text for the task adaptation procedure, and use the annotations provided in the dataset for the ChemProt classification task.

4 Methods

4.1 Linguistic Probing

There are several common tasks that are used to evaluate the linguistic knowledge of NLP models. We base our tasks from (Tenney et al., 2019a) approach, where 6 tasks are used: part-of-speech tagging (POS), dependencies (Dep.), named entity recognition (NER), semantic role labeling (SRL), coreference resolution (Coref.), and relation classification (Rel.). Some tasks are more syntactic in nature, for example POS and NER tasks, while other tasks evaluate semantic understanding. Figure 1 provides a description of the tasks, as well as the level of syntactic-semantic information the particular task probes.

We use the edge probing techniques with scalar mixing at all intermediate checkpoints provided in (Sellam et al., 2021) to derive insights of the linguistic learning dynamics of BERT at a layer by layer basis. To make sure that we are probing for information that the model contains at training time, we freeze the encoder (BERT) weights and only train the edge classifier head. The metric used to evaluate performance is the F-1 micro metric, to have an uniform metric that we will use across our probing suite. The code used to train the edge probing classifier head was adapted from (Wang et al., 2019).

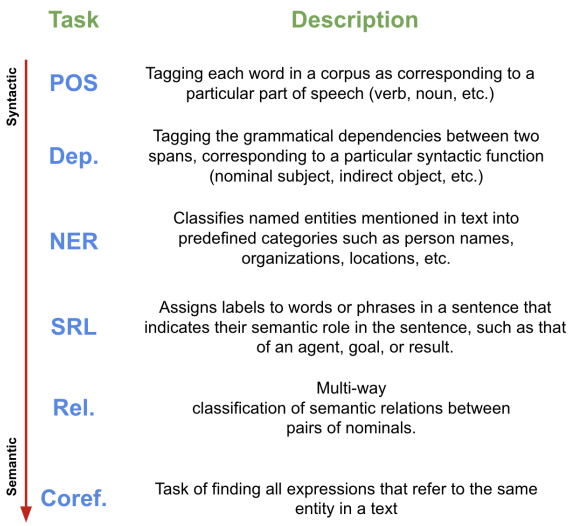

Figure 1: Linguistic probing task descriptions.

4.2 Knowledge Probe

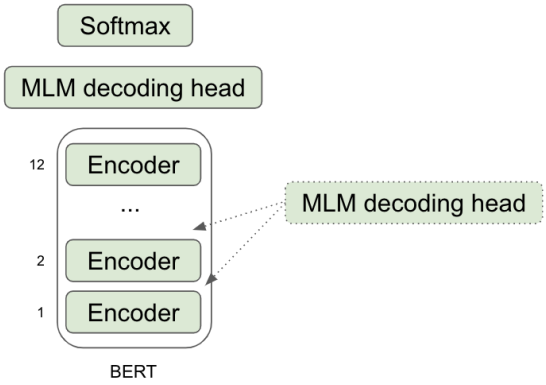

Figure 2: Architecture of knowledge probe

Our knowledge probe method and data is based on (Petroni et al., 2019b). LAMA probe is done by querying the models with a masked natural language prompts, which are the facts in the LAMA dataset, such as "Dante was born in [MASK] in the year 1264". More examples of the facts from the 4 data sources can be found in Table 17. Given that BERT's training approaches include masked-language modeling (MLM), its MLM decoding

head can be used to predict a masked word in a sequence. The architecture is as illustrated in Figure 2. For each fact, we use it as input and let BERT's MLM decoding head generate the logits for the masked index. The logits will have the dimension of the length of vocabulary. The logits are then passed through log softmax to generate log probabilities of all words in the vocabulary. The language models are used off-the-shelf, meaning no further training is done for them. The log probability ranking of the words will determine what answer BERT predicts the masked token to be. MLM decoding heads will be added at the end of each of the 12 BERT encoder layers to conduct layer-by-layer analysis. To evaluate how much knowledge is stored in BERT, we look at the top k ranking of the log probabilities to understand where the correct answer lies in the model's output.

The evaluation of the probing results for each fact is done using the metric precision at k (P@k), which has a value of 1 if the correct answer lies in top k, 0 otherwise. In our experiment, we mostly use P@10 as the metric. Collectively, for each corpus (Google-RE, T-REx, ConceptNet, or SQuAD), we evaluate the mean p@10 for all facts. A P@10 of 50% means that the correct answer is ranked among the top 10 in 50% of the facts in the corpus. This process will be replicated for all BERT models with various checkpoints and initialization seeds as described in Section 4.4 and all BERT models adapted with specific tasks as described in Section 4.3.

4.3 Task Adaptation

We performed the task-adaptation pretraining procedure described in (Gururangan et al., 2020) to all of our model checkpoints, utilizing the ChemProt dataset described in Section 3. We pretrained the models with the unlabeled ChemProt dataset performing random masked language modeling. We randomly chose 15% of words in the corpus to mask, and trained the models to predict this masked words. All models were trained for 5k steps.

After doing the taks adaptation pretraining, we attached a simple linear classifier head to the adapted model and trained the classifier head to perform a multi-class classification. This task relies in trying to classify the chemical interaction that occurs between two given compounds in a given sequence.

After training the classifier head on the adapted models, we also train a classifier head for the same task on the original, unadapted, models to be able to compare the performances between the adapted and unadapted models. We also run the probes described in Sections 4.1 and 4.2 on the adapted models to understand how this task-adaptation pretraining procedure caused forgetting in linguistic and commonsense knowledge information.

4.4 Models

We apply our techniques described previously on the BERT models provided by (Sellam et al., 2021) hosted on Hugging Face. The repository has saved BERT models with 5 initialization seeds and each with 28 intermediate checkpoints.

5 Results

Our results from the linguistic probe, knowledge probe and task adaptation experiment attempt to answer the following questions: how and when is information learned in the pre-training process; how does initialization seeds influence the amount of information learned; in which layers is this information stored?; how does pretraining cause forgetting? We will discuss each of these questions in the subsequent sections.

5.1 Linguistic Information

How does initialization seeds influence the amount of linguistic information learned?

Our first findings are that the BERT model is mostly robust against parameter initialization, as the performances of the probes do not variate excessively between random seeds. We could note though, that the initialization seed is more critical in semantic-heavy tasks, as the performances on these tasks results variate more (around 1% in F1-Score) than their syntactic counterparts, which performances stay almost virtually the same, as shown in Figure 3. This hints that random seed initialization is more critical when pretraining a model if the downstream tasks intended for that model is more semantic in nature.

How and when is information learned in the pretraining process?

Comparing the F1 scores achieved at different intermediate checkpoints, we can see that the BERT model learns linguistic information pretty early in the pretraining phase. Figure 4 shows the relative increase of a particular checkpoint against the previous checkpoint, and we can note that this in-

Check.	Rel.	Dep.	Coref.	NER	POS	SRL
0k	0.339	0.731	0.885	0.809	0.870	0.735
20k	0.645	0.910	0.929	0.934	0.962	0.875
60k	0.694	0.922	0.935	0.941	0.967	0.893
100k	0.697	0.924	0.938	0.943	0.969	0.897
200k	0.711	0.926	0.941	0.945	0.970	0.900
400k	0.727	0.929	0.945	0.949	0.971	0.905
700k	0.728	0.932	0.948	0.951	0.972	0.908
1000k	0.729	0.934	0.949	0.953	0.973	0.910
1500k	0.742	0.935	0.948	0.955	0.973	0.913
1800k	0.742	0.935	0.950	0.955	0.973	0.913
2000k	0.739	0.935	0.951	0.955	0.973	0.913

Table 2: Average performance of all linguistic tasks for each BERT checkpoint.

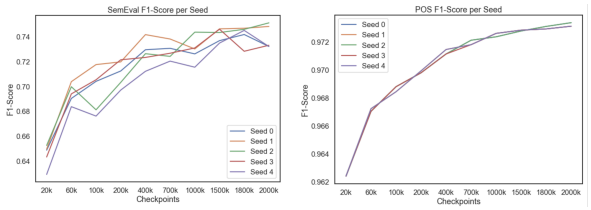

Figure 3: Performance of each model initialized with a different random in the Relations and POS tasks. The variability in performance between seeds in the Relations task and in semantic tasks in general are higher than those for the syntactic tasks.

crease tails off pretty early on the training in virtually all tasks. This hints that the amount of linguistic information in the model stays mostly the same pretty early on the the pretraining procedure and you might not benefit in pretraining the BERT model for longer.

Check.	Rel.	Dep.	Coref.	NER	POS	SRL
0k	5.973	5.383	5.336	5.269	3.727	4.831
20k	6.405	6.902	6.770	6.533	7.132	7.464
60k	6.431	6.868	7.051	6.472	6.872	7.552
100k	6.425	6.872	7.069	6.390	6.719	7.530
200k	6.421	6.865	7.071	6.304	6.529	7.149
400k	6.405	6.831	7.047	6.269	6.348	7.013
700k	6.377	6.720	6.959	6.203	6.220	6.848
1000k	6.357	6.644	6.890	6.156	6.150	6.778
1500k	6.348	6.578	6.878	6.116	6.061	6.681
1800k	6.338	6.570	6.884	6.089	6.034	6.660
2000k	6.343	6.564	6.892	6.109	6.027	6.649

Table 3: Layer mixing weights Center of Gravity of all linguistic tasks per BERT checkpoint.

Doing the layer by layer analysis utilizing the scalar mixing weights, we can see that in general, the syntactic tasks show more importance to the lower layers, while the semantic tasks give more importance to the layers in the higher-end, which is demonstrated in the Center of Gravity values in

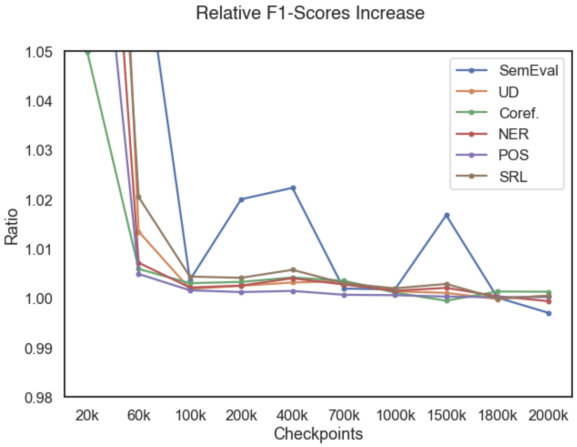

Figure 4: Relative performance of the different BERT checkpoints for all linguistic tasks. The relative performance is calculated by taking the ratio of the performance of the current checkpoint against the performance of the previous checkpoint.

Table 3. POS and NER have consistently the lower COG values, and are the most syntactic tasks, while the Coref. and SRL tasks have the highest COG values, and these are tasks that are more semantic in nature. This shows that syntactic information is stored in the lower layers of the BERT model, while the semantic information is stored in the higher layers. Intuitively this makes sense, as the BERT model is learning to first parse and figure out the structure of the given sequence in its lower layers, and as information is passed to the higher layers, BERT is trying to understand the meaning behind this already parsed and structured sequence.

Another phenomena that was uncovered is the shifting of the layer importances as pretraining goes on. As shown in Figure 5, at the beginning of pretraining (20k checkpoint), the scalar weights are generally higher in the higher layers, and as pretraining keeps going, we can see a shift in the mixing weights, where the lower layers start getting higher mixing weights. This shifting of weights was observed in all tasks. Interestingly, a more dramatic shift is found in the syntactic-heavy tasks. Figure 6 shows the COG value of each task throughout the pretraining procedure. This is consistent with our finding above, and hints that at the beggining of the pretraining, the BERT model has all information stored in the higher layers, and as pretraining continues, BERT learns to separate semantic and syntactic information, and pushed this syntactic information to the lower layers of the model.

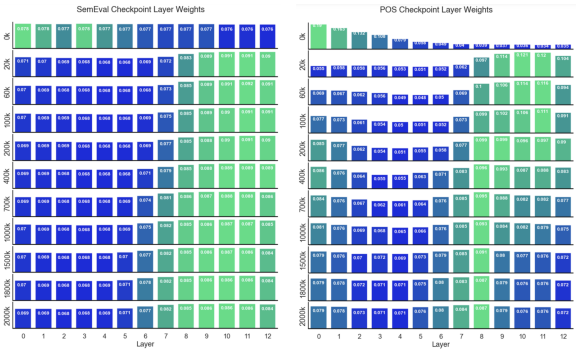

Figure 5: Layer mixing weights after the softmax function. A shift from higher mixing weights in the higher layers to the lower layers happens as the training steps keep increasing. This phenomena occurs for all tasks.

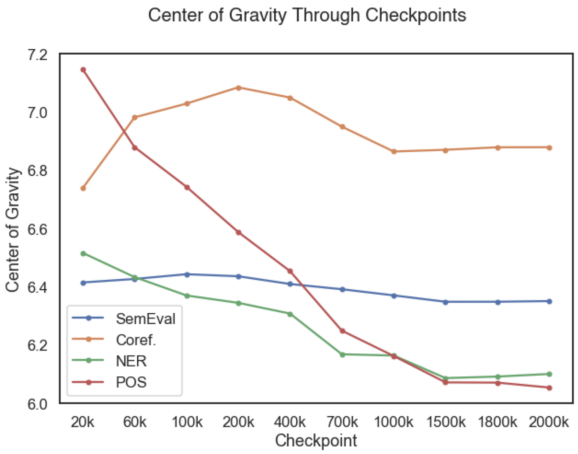

Figure 6: Layer Center of Gravity shift throughout the model checkpoints. A shift from higher COGs to lower COGs happens as the training steps keep increasing. This effect is stronger in syntactic tasks (POS, NER) than in the semantic tasks (SemEval (Relations), Coref.).

5.2 Commonsense Knowledge

How is commonsense knowledge learned in the pre-training process? Figure 7 shows the P@10 results averaged across all relations within each of Google-RE, SQuAD, and T-REx corpora. We observe that all BERT models with 5 different initialization seeds can achieve a decent P@10 at around checkpoint 60k. Further training does not significantly increase the models' performance in terms of commonsense knowledge understanding.

How does initialization seeds influence the amount of knowledge learned? We observe that different initialization seeds can have great or little influence on the performance of BERT models understanding commonsense knowledge, even when the models are trained for as long as 2000k steps. In Figure 8, we show 4 example relations in T-REx

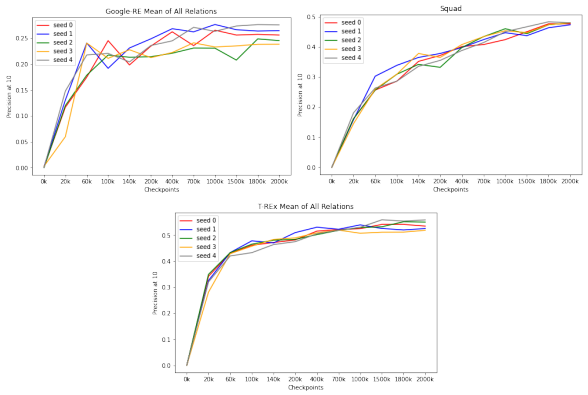

Figure 7: The progression of P@10 performance as training goes on when probing with Google-RE, SQuAD, and T-REx. P@10 of Google-RE and T-REx is an averaged P@10 of all relations in the corresponding corpus.

corpus and their corresponding P@10 performance for models at different checkpoints. In certain relations such as P36 and P1001, different seeds don't influence the performance very much, while in relations like P140 and P413, the difference in p@10 can range from nearly 0% to 50%. This implies that, for certain relations, BERT doesn't learn very much knowledge with one seed but learns a fairly great amount with another. The difficulties of the facts in these relations can play a role in the discrepancy of performance. In Table 4 we show example facts from the 4 relations compared in Figure 8. Place and country names are likely easier to learn compared to religion and sports.

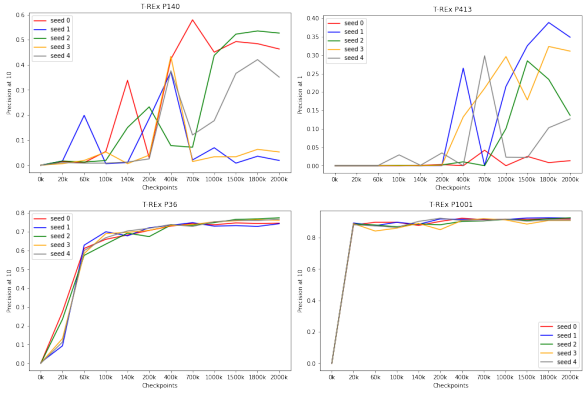

Figure 8: Difference in P@10 performance on 4 different T-REx relations when applying 5 different seeds. BERT learns P36 and P1001 more robustly against seeds than it learns P140 and P413.

In which layers is commonsense knowledge learned? Figure 9 shows the learning progression of commonsense knowledge in Google-RE

Relation Code	Relation Type	Example
P140	Religion	Following the victory of [MASK] emperor Constantine in the Civil Wars of the Tetrarchy...
P413	Sports	Thunder [MASK] Jonathan Quinn earned MVP honors with his performance of 25 completions...
P36	Place	The capital of Azerbaijan is the ancient city of [MASK]...
P1001	Country	On 10 May 1981 François Mitterrand was elected President of [MASK].

Table 4: T-REx example relation types and facts

and SQuAD corpora at layer level. We observe that most of the commonsense knowledge is learned in higher layers. There exist differences of learning pattern based on tasks as well. The P@10 performance starts going up after around layer 8 in SQuAD but around layer 6 in Google-RE. It is also shown that as training proceeds, with more training steps, knowledge can be learned at earlier layers. This implies that the commonsense knowledge learning ability shifts to lower layers as we pre-train. This finding aligns with the center of gravity shift towards lower layers.

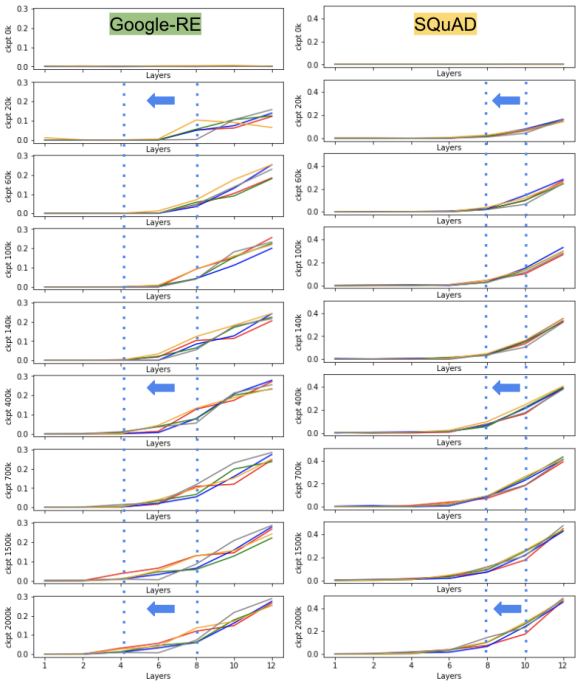

Figure 9: Progression of P@10 performance from layer 1 to 12 of BERT's architecture, probed with Google-RE and SQuAD. Both exhibit shift of layer where learning starts.

5.3 Task Adaptation

After performing the task adaptation pretraining in the BERT checkpoints, there were several interesting results. First, we could see that there was some variation in the performances between the seeds, which is akin to the variation seen in the commonsense knowledge and semantic task probes. This again hints that seed initialization might be important when a BERT model is going to be used for task or domain adaptation. The performances of the adapted models can be seen in Figure 10. When we compare the performance of the adapted models versus the original, non-adapted models (Figure 11), we can see that the task adaptation in fact provides a boost in performance in downstream tasks that are related to the corpora that was used for the task adaptation. Interestingly, the performance that a not-fully pretrained model that was adapted is comparable to the performance of the fully pretrained model that was adapted. We can see in Figure 11 that the performance achieved by Checkpoint 1000k is the same as the performance of the fully trained, 2000k checkpoint. This shows that if you want to do task adaptation to the BERT model, you do not need to use the fully trained BERT model to have the best results possible.

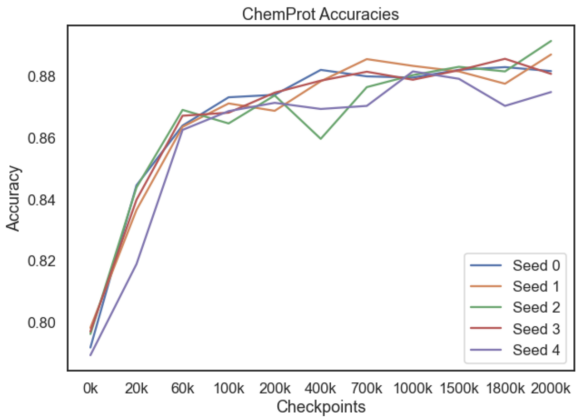

Figure 10: Accuracies of all adapted models in the ChemProt classification task, for all seeds and checkpoints. There is some variability in the performances between seeds akin to that of the semantic tasks in the probing tasks.

When running the probes in the adapted models, we could see that the performances for all tasks went down, although more for some tasks than others. The tasks that took the biggest hit in performance were the semantic tasks, as the performances went down considerably for all checkpoints. Figure 12 shows the comparison of the per-

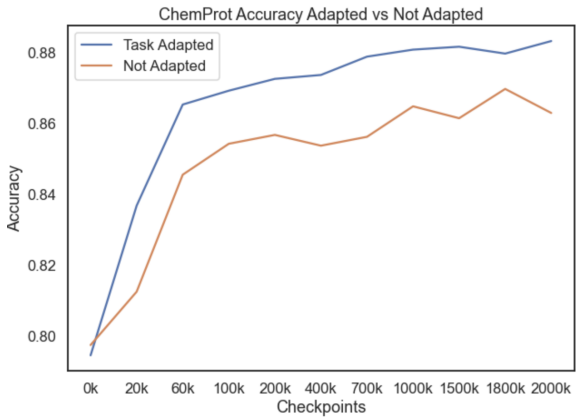

Figure 11: Accuracies of all adapted models in the ChemProt classification task, for all seeds and checkpoints. There is some variability in the performances between seeds akin to that of the semantic tasks in the probing tasks.

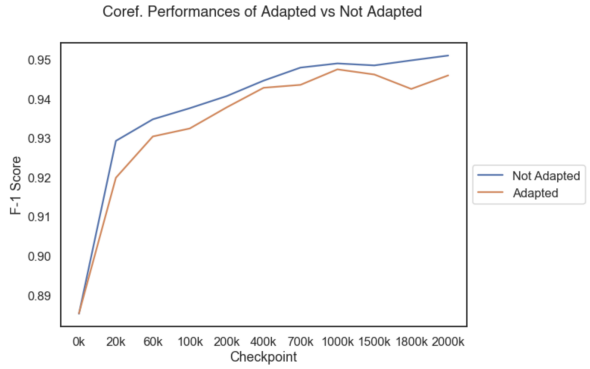

Figure 12: Performance of adapted model versus performance of non-adapted model for the coreference resolution task. The adapted model presents a lower performance for all checkpoints in comparison to the original, non-adapted model.

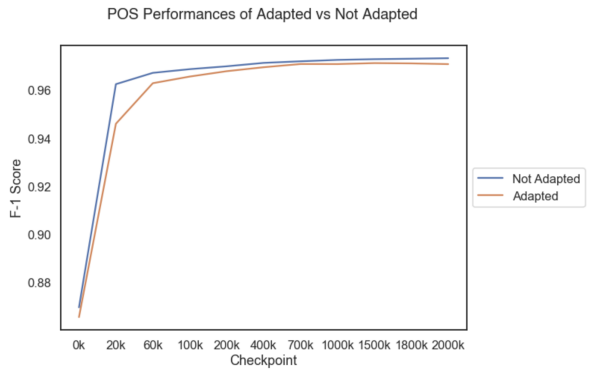

Figure 13: Performance of adapted model versus performance of non-adapted model for the POS task. The adapted model presents a slightly lower performance for some checkpoints in comparison to the original, non-adapted model, although most checkpoint performances are the same.

formance in the Coreference resolution probe between the adapted and non adapted models, while Figure 13 shows the comparison of the adapted and non adapted models for the POS probe. It is clear that the performance in the syntactic task stays mostly the same, while the semantic evaluation tasks take a hit in performance. These finding hints at the forgetting pattern that is present in the BERT model. When adapting the BERT model to a different corpora, the semantic information that is contained in the higher layers of the models gets 'swapped out' for the new information that is in the new corpora. The syntactic information that is contained in the lower layers is mostly retained. Looking at the mixing weights of the layers in the adapted models in Figures 15 and 14 also confirms this, as in all probing tasks the layer importances are lower than that of the original models that were probed in the previous sections. Figure **??** also confirms that the performance probing commonsense knowledge, behaving just like semantic tasks, drastically drops to near 0 after task adaptation. Specific values are shown in Figure **??**. This shows that the information that gets wiped out by the task adaptation procedure is mostly contained in the higher layers, and thus the probes give more importance to the lower layers, as these higher layers now have information that is not relevant for the probes.

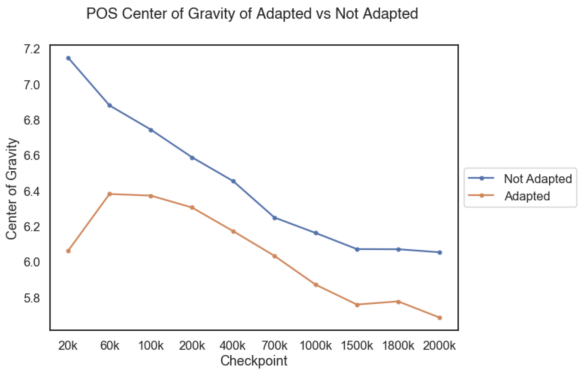

Figure 14: Layer center of gravity of adapted model versus center of gravity of non-adapted model for the POS task. The adapted model presents a lower CoG for all checkpoints in comparison to the original, non-adapted model.

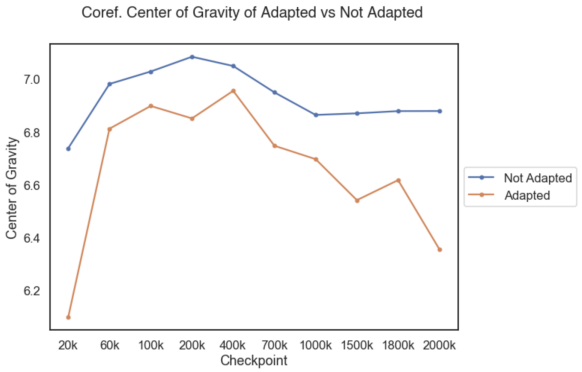

Figure 15: Layer center of gravity of adapted model versus center of gravity of non-adapted model for the coreference resolution task. The adapted model presents a lower CoG for all checkpoints in comparison to the original, non-adapted model.

6 Conclusion

In this paper, we utilize multiple BERT models with 5 initialization seeds and various checkpoints from untrained (0k) to fully trained (2000k). We conduct experiment with them on linguistics- and knowledge-based tasks. Variation of performance using different seeds is less obvious in learning of syntactic tasks, but can vary more in learning certain types of knowledge, depending on the difficulty of the knowledge and its abundance in training data. The number of training steps is not a deterministic factor of BERT's performance on these tasks. This results implies that computing power spent on long training steps can be saved while developing a large language model if necessary to achieve similar results. It could worth more to use models applied with multiple seeds to reduce the influence of random seeds in downstream task performance.

Through layer by layer probing approaches, we have identified that lower layers learn more syntactic tasks while higher layers learn more semantic and knowledge-based tasks. BERT learns the structure of a language before learning the meanings of its components.

References

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.

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.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O Seaghdha, Sebastian Pado, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions.*

Martin Krallinger, Obdulia Rabal, Saber Ahmad Akhondi, Martín Pérez Pérez, Jesus Santamaría, Gael Pérez Rodríguez, Georgios Tsatsaronis, Ander Intxaurrondo, José Antonio Baso López, Umesh K. Nandal, Erin M. van Buel, Ambika Chandrasekhar, Marleen Rodenburg, Astrid Lægreid, Marius A. Doornenbal, Julen Oyarzábal, Anália Lourenço, and Alfonso Valencia. 2017. Overview of the biocreative vi chemical-protein interaction track.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019a. Language Models as Knowledge Bases? *Computer Science*, arXiv:1909.01066.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019b. Language models as knowledge bases?

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

Thibault Sellam, Steve Yadlowsky, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn

Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, Ian Tenney, and Ellie Pavlick. 2021. The MultiBERTs: BERT Reproductions for Robustness Analysis. *Computer Science*, arXiv:2106.16163. Version 4.

Natalia Silveira, Timothy Dozata, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A Gold Standard Dependency Corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation.*

Robyn Speer and Catherine Havasi. 2012. Representing general relational knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679–3686, Istanbul, Turkey. European Language Resources Association (ELRA).

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT Rediscovers the Classical NLP Pipeline. *Computer Science*, arXiv:1905.05950.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? Probing for sentence structure in contextualized word representations. *Computer Science*, arXiv:1905.06316.

Alex Wang, Ian F. Tenney, Yada Pruksachatkun, Phil Yeres, Jason Phang, Haokun Liu, Phu Mon Htut, , Katherin Yu, Jan Hula, Patrick Xia, Raghu Pappagari, Shuning Jin, R. Thomas McCoy, Roma Patel, Yinghui Huang, Edouard Grave, Najoung Kim, Thibault Févry, Berlin Chen, Nikita Nangia, Anhad Mohananey, Katharina Kann, Shikha Bordia, Nicolas Patry, David Benton, Ellie Pavlick, and Samuel R. Bowman. 2019. `jiant` 1.3: A software toolkit for research on general-purpose text understanding models. `http://jiant.info/`.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xuea, Ann Taylor, Jeff Kaufman, and Michelle Franchini. 2013. OntoNotes release 5.0 LDC2013T19. Philadelphia, PA. Linguistic Data Consortium.

A Appendix

A.1 Full Linguistic Probe Results

Check.	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4	Average
0k	0.886	0.884	0.884	0.886	0.886	0.885
20k	0.929	0.929	0.929	0.929	0.929	0.929
60k	0.934	0.935	0.935	0.934	0.934	0.935
100k	0.938	0.936	0.936	0.938	0.938	0.938
200k	0.942	0.938	0.938	0.942	0.942	0.941
400k	0.946	0.942	0.942	0.946	0.946	0.945
700k	0.949	0.947	0.947	0.947	0.949	0.948
1000k	0.948	0.950	0.950	0.950	0.948	0.949
1500k	0.948	0.949	0.949	0.948	0.948	0.948
1800k	0.949	0.951	0.951	0.949	0.949	0.950
2000k	0.950	0.952	0.952	0.950	0.950	0.951

Table 5: F1-Scores for all seeds for the Coref. task.

Check.	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4	Average
0k	5.275	5.582	5.275	5.275	5.275	5.336
20k	6.738	6.817	6.738	6.817	6.738	6.770
60k	6.981	7.157	6.981	7.157	6.981	7.051
100k	7.028	7.131	7.028	7.131	7.028	7.069
200k	7.084	7.019	7.084	7.084	7.084	7.071
400k	7.049	7.040	7.049	7.049	7.049	7.047
700k	6.949	6.973	6.949	6.973	6.949	6.959
1000k	6.864	6.997	6.864	6.864	6.864	6.890
1500k	6.870	6.913	6.870	6.870	6.870	6.878
1800k	6.878	6.907	6.878	6.878	6.878	6.884
2000k	6.878	6.944	6.878	6.878	6.878	6.892

Table 6: Layer Center of Gravity for all seeds for the Coref. task.

Check.	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4	Average
0k	0.872	0.866	0.872	0.872	0.866	0.870
20k	0.962	0.962	0.962	0.962	0.962	0.962
60k	0.967	0.967	0.967	0.967	0.967	0.967
100k	0.969	0.968	0.969	0.969	0.968	0.969
200k	0.970	0.970	0.970	0.970	0.970	0.970
400k	0.971	0.971	0.971	0.971	0.971	0.971
700k	0.972	0.972	0.972	0.972	0.972	0.972
1000k	0.972	0.973	0.972	0.973	0.973	0.973
1500k	0.973	0.973	0.973	0.973	0.973	0.973
1800k	0.973	0.973	0.973	0.973	0.973	0.973
2000k	0.973	0.973	0.973	0.973	0.973	0.973

Table 7: F1-Scores for all seeds for the POS task.

Check.	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4	Average
0k	3.757	3.610	3.757	3.757	3.757	3.727
20k	7.146	7.073	7.146	7.146	7.146	7.132
60k	6.879	6.847	6.879	6.879	6.879	6.872
100k	6.743	6.703	6.743	6.703	6.703	6.719
200k	6.587	6.491	6.587	6.491	6.491	6.529
400k	6.453	6.278	6.453	6.278	6.278	6.348
700k	6.248	6.178	6.248	6.178	6.248	6.220
1000k	6.161	6.105	6.161	6.161	6.161	6.150
1500k	6.071	6.024	6.071	6.071	6.071	6.061
1800k	6.070	5.980	6.070	6.070	5.980	6.034
2000k	6.053	5.988	6.053	6.053	5.988	6.027

Table 8: Layer Center of Gravity for all seeds for the POS task.

Check.	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4	Average
0k	0.733	0.735	0.736	0.735	0.736	0.735
20k	0.875	0.876	0.874	0.876	0.874	0.875
60k	0.891	0.892	0.894	0.893	0.894	0.893
100k	0.896	0.896	0.897	0.897	0.897	0.897
200k	0.902	0.900	0.900	0.900	0.900	0.900
400k	0.906	0.905	0.905	0.905	0.905	0.905
700k	0.908	0.908	0.908	0.908	0.908	0.908
1000k	0.911	0.910	0.910	0.911	0.910	0.910
1500k	0.913	0.912	0.913	0.912	0.913	0.913
1800k	0.913	0.912	0.913	0.912	0.913	0.913
2000k	0.913	0.913	0.913	0.913	0.913	0.913

Table 9: F1-Score for all seeds for the SRL task.

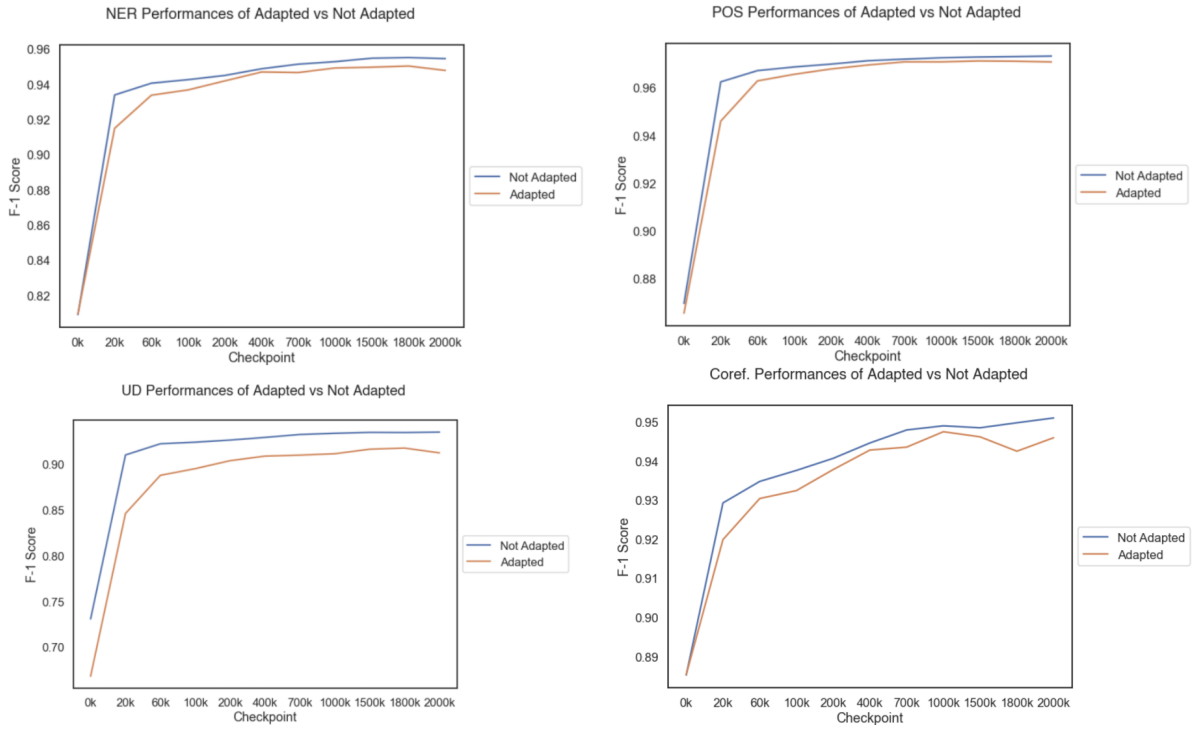

Figure 16: All probes performances of adapted models versus non-adapted models.

Check.	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4	Average
0k	4.405	4.487	4.640	4.524	6.099	4.831
20k	7.114	7.273	7.200	7.273	8.460	7.464
60k	7.247	7.327	7.341	7.295	8.551	7.552
100k	7.258	7.272	7.299	7.277	8.541	7.530
200k	7.168	7.145	7.145	7.145	7.145	7.149
400k	7.059	6.982	7.059	6.982	6.982	7.013
700k	6.895	6.817	6.895	6.817	6.817	6.848
1000k	6.789	6.733	6.789	6.789	6.789	6.778
1500k	6.685	6.664	6.685	6.685	6.685	6.681
1800k	6.669	6.624	6.669	6.669	6.669	6.660
2000k	6.654	6.628	6.654	6.654	6.654	6.649

Table 10: Layer Center of Gravity for all seeds for the SRL task.

Check.	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4	Average
0k	0.810	0.804	0.826	0.806	0.799	0.809
20k	0.934	0.935	0.933	0.935	0.933	0.934
60k	0.941	0.942	0.940	0.941	0.938	0.941
100k	0.943	0.943	0.942	0.942	0.942	0.943
200k	0.946	0.947	0.944	0.943	0.944	0.945
400k	0.950	0.949	0.949	0.950	0.946	0.949
700k	0.952	0.951	0.951	0.952	0.951	0.951
1000k	0.954	0.952	0.952	0.955	0.952	0.953
1500k	0.954	0.955	0.954	0.956	0.955	0.955
1800k	0.955	0.956	0.954	0.956	0.954	0.955
2000k	0.954	0.955	0.954	0.956	0.954	0.955

Table 11: F1-Scores for all seeds for the NER task.

Check.	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4	Average
0k	5.153	5.314	5.314	5.305	5.257	5.269
20k	6.516	6.496	6.496	6.476	6.680	6.533
60k	6.432	6.469	6.469	6.393	6.596	6.472
100k	6.369	6.356	6.356	6.365	6.503	6.390
200k	6.344	6.253	6.253	6.272	6.399	6.304
400k	6.307	6.216	6.216	6.244	6.363	6.269
700k	6.167	6.183	6.183	6.171	6.309	6.203
1000k	6.163	6.127	6.127	6.116	6.248	6.156
1500k	6.085	6.088	6.088	6.123	6.196	6.116
1800k	6.091	6.066	6.066	6.132	6.091	6.089
2000k	6.100	6.086	6.086	6.136	6.136	6.109

Table 12: Layer Center of Gravity for all seeds for the NER task.

Check.	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4	Average
0k	0.810	0.804	0.826	0.806	0.799	0.809
20k	0.934	0.935	0.933	0.935	0.933	0.934
60k	0.941	0.942	0.940	0.941	0.938	0.941
100k	0.943	0.943	0.942	0.942	0.942	0.943
200k	0.946	0.947	0.944	0.943	0.944	0.945
400k	0.950	0.949	0.949	0.950	0.946	0.949
700k	0.952	0.951	0.951	0.952	0.951	0.951
1000k	0.954	0.952	0.952	0.955	0.952	0.953
1500k	0.954	0.955	0.954	0.956	0.955	0.955
1800k	0.955	0.956	0.954	0.956	0.954	0.955
2000k	0.954	0.955	0.954	0.956	0.954	0.955

Table 13: F1-Scores for all seeds for the Relations task.

Check.	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4	Average
0k	5.974	5.951	5.991	5.972	5.978	5.973
20k	6.414	6.413	6.384	6.401	6.414	6.405
60k	6.426	6.432	6.439	6.410	6.449	6.431
100k	6.442	6.420	6.416	6.417	6.430	6.425
200k	6.435	6.429	6.400	6.413	6.429	6.421
400k	6.409	6.417	6.398	6.381	6.421	6.405
700k	6.391	6.381	6.359	6.380	6.373	6.377
1000k	6.370	6.362	6.354	6.329	6.367	6.357
1500k	6.348	6.359	6.360	6.319	6.357	6.348
1800k	6.348	6.351	6.321	6.325	6.345	6.338
2000k	6.350	6.347	6.344	6.324	6.348	6.343

Table 14: Layer Center of Gravity for all seeds for the Relations task.

Check.	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4	Average
0k	0.732	0.732	0.733	0.728	0.729	0.731
20k	0.911	0.911	0.910	0.911	0.907	0.910
60k	0.923	0.923	0.923	0.923	0.918	0.922
100k	0.925	0.925	0.925	0.923	0.921	0.924
200k	0.927	0.927	0.927	0.926	0.924	0.926
400k	0.931	0.931	0.929	0.929	0.927	0.929
700k	0.933	0.933	0.932	0.933	0.932	0.932
1000k	0.934	0.934	0.933	0.934	0.933	0.934
1500k	0.935	0.935	0.934	0.934	0.935	0.935
1800k	0.935	0.935	0.933	0.935	0.934	0.935
2000k	0.935	0.935	0.936	0.935	0.934	0.935

Table 15: F1-Scores for all seeds for the Dependencies task.

Check.	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4	Average
0k	5.416	5.362	5.337	5.403	5.398	5.383
20k	6.887	6.903	6.881	6.903	6.935	6.902
60k	6.868	6.867	6.851	6.857	6.894	6.868
100k	6.855	6.876	6.834	6.864	6.931	6.872
200k	6.855	6.851	6.851	6.857	6.914	6.865
400k	6.828	6.810	6.824	6.798	6.894	6.831
700k	6.725	6.702	6.679	6.724	6.768	6.720
1000k	6.618	6.616	6.654	6.642	6.690	6.644
1500k	6.577	6.516	6.613	6.611	6.575	6.578
1800k	6.574	6.506	6.617	6.594	6.557	6.570
2000k	6.541	6.501	6.624	6.595	6.560	6.564

Table 16: Layer Center of Gravity for all seeds for the Dependencies task.

Name	Relation	Example
Google-RE	date of birth	Thomas Henning (born [MASK]) is a German astrophysicist.
	place of birth	Graham Salisbury (born April 11, 1944, [MASK], Pennsylvania) is an American author.
	place of death	Captain Bonser died just over a year later at Providence St Vincent Medical Center in [MASK].
SQuAD	\	Newton played as [MASK] during Super Bowl 50.

Table 17: Example data in LAMA