BERTnesia: Investigating the capture and forgetting of knowledge in BERT

Jonas Wallat
L3S Research Center
Hannover, Germany
wallat@l3s.de

Jaspreet Singh
L3S Research Center
Hannover, Germany
singh@l3s.de

Avishek Anand
L3S Research Center
Hannover, Germany
anand@l3s.de

Abstract

Probing complex language models has recently revealed several insights into linguistic and semantic patterns found in the learned representations. In this paper, we probe BERT specifically to understand and measure the relational knowledge it captures. We utilize knowledge base completion tasks to probe every layer of pre-trained as well as fine-tuned BERT (ranking, question answering, NER). Our findings show that knowledge is not just contained in BERT's final layers. Intermediate layers contribute a significant amount (17-60%) to the total knowledge found. Probing intermediate layers also reveals how different types of knowledge emerge at varying rates. When BERT is fine-tuned, relational knowledge is forgotten but the extent of forgetting is impacted by the fine-tuning objective but not the size of the dataset. We found that ranking models forget the least and retain more knowledge in their final layer. We release our code on github[1] to repeat the experiments.

1 Introduction

Large pre-trained language models like BERT (Devlin et al., 2019) have heralded an *Imagenet* moment for NLP[2] with not only significant improvements made to traditional tasks such as question answering and machine translation but also in the new areas such as knowledge base completion. BERT has over 100 million parameters and essentially trades off transparency and interpretability for performance. Loosely speaking, probing is a commonly used technique to better understand the inner workings of BERT and other complex language models (Dasgupta et al., 2018; Ettinger et al., 2018). Probing, in general, is a procedure by which one tests for a specific pattern – like local syntax, long-range semantics or even compositional reasoning – by constructing inputs whose expected output would not be possible to predict without the ability to detect that pattern. While a large body of work exists on probing BERT for linguistic patterns and semantics, there is limited work on probing these models for the factual and relational knowledge they store.

Recently, Petroni et al. (2019) probed BERT and other language models for relational knowledge (e.g., *Trump* is the president of the *USA*) in order to determine the potential of using language models as automatic knowledge bases. Their approach converted queries in the knowledge base (KB) completion task of predicting arguments or relations from a KB triple into a natural language cloze task, e.g., [MASK] is the president of the *USA*. This is done to make the query compatible with the pre-training masked language modeling (MLM) objective. They consequently showed that a reasonable amount of knowledge is captured in BERT by considering multiple relation probes. However, there are some natural questions that arise from these promising investigations: *Is there more knowledge in* BERT *than what is reported? What happens to relational knowledge when* BERT *is fine-tuned for other tasks? Is knowledge gained and lost through the layers?*

Our Contribution. In this paper, we study the emergence of knowledge through the layers in BERT by devising a procedure to estimate knowledge contained in every layer and not just the last (as done by Petroni et al. (2019)). While this type of layer-by-layer probing has been conducted for syntactic, grammatical, and semantic patterns; knowledge probing has only been conducted on final layer representations. Observing only the final layer (as we will show in our experiments)

---

[1] https://github.com/jwallat/knowledge-probing
[2] https://thegradient.pub/nlp-imagenet/

(i) underestimates the amount of knowledge and (ii) does not reveal how knowledge emerges. Furthermore, we explore how knowledge is impacted when fine-tuning on knowledge-intensive tasks such as question answering and ranking. We list the key research questions we investigated and key findings corresponding to them:

RQ I: Do intermediary layers capture knowledge not present in the last layer? (Section 4.1)

We find that a substantial amount of knowledge is stored in the intermediate layers ($\approx 24\%$ on average)

RQ II: Does all knowledge emerge at the same rate? Do certain types of relational knowledge emerge more rapidly? (Section 4.2)

We find that not all relational knowledge is captured gradually through the layers with 15% of relationship types essentially doubling in the last layer and 7% of relationship types being maximally captured in an intermediate layer.

RQ III: What is the impact of fine-tuning data on knowledge capture? (Section 4.3)

We find that the dataset size does not play a major role when the training objective is fixed as MLM. Fine-tuning on a larger dataset does not lead to less forgetting.

RQ IV: What is the impact of the fine tuning objective on knowledge capture? (Section 4.4)

Fine tuning always causes forgetting. When the size of the dataset is fixed and training objective varies, the ranking model (RANK-MSMARCO in our experiments) forgets less than the QA model.

2 Related Work

In this section, we survey previous work on probing language models (LMs) with a particular focus on contextual embeddings learned by BERT. Probes have been designed for both static and contextualized word representations. Static embeddings refer to non-contextual embeddings such as GloVe (Pennington et al., 2014). For the static case, the reader can refer to this survey by Belinkov and Glass (2019). Now we detail probing tasks for contextualized embeddings from language models.

2.1 Probing for syntax, semantics, and grammar

Initial work on probing dealt with linguistic pattern detection. Peters et al. (2018) investigated the ability of various neural network architectures that learn contextualized word representations to capture local syntax and long-range semantics like coreference resolution while Dasgupta et al. (2018); Ettinger et al. (2018) probed language models for compositional reasoning.

McCoy et al. (2019); Goldberg (2019) found that BERT is able to effectively learn syntactic heuristics with natural language inference specific probes. Tenney et al. (2019); Liu et al. (2019); Jawahar et al. (2019) investigated BERT layer-by-layer for various syntactic and semantic patterns like part-of-speech, named entity recognition, coreference resolution, entity type prediction, semantic role labeling, etc. They all found that basic linguistic patterns like part of speech emerge at the lower layers. However, there is no consensus with regards to semantics with somewhat conflicting findings (equally spread vs final layer (Jawahar et al., 2019)). Kovaleva et al. (2019) found that the last layers of fine-tuned BERT contain the most amount of task-specific knowledge. van Aken et al. (2019) showed the same result for fined tuned QA BERT with specially designed probes. They found that the lower and intermediary layers were better suited to linguistic subtasks associated with QA. For a comprehensive survey we point the reader to (Rogers et al., 2020)

Our work is similar to these studies in terms of setup. In particular, our probes function on the sentence level and are applied to each layer of a pre-trained BERT model as well as BERT fine-tuned on several tasks. However, we do not focus on detecting linguistic patterns and focus on relational and factual knowledge.

2.2 Probing for knowledge

In parallel, there have been investigations into probing for factual and world knowledge. Most recently, Petroni et al. (2019) found that LMs like BERT can be directly used for the task of knowledge base completion since they are able to memorize more facts than some automatic knowledge bases. They created cloze statement tasks for factual and commonsense knowledge and measured cloze-task performance as a proxy for the knowledge contained. However, using the same

probing framework, Kassner and Schütze (2020) showed that this factoid knowledge is influenced by surface-level stereotypes of words. For example, BERT often predicts a typically German name as a German citizen. Tangentially, Forbes et al. (2019) investigated BERT's awareness of the world. They devised object property and action probes to estimate BERT's ability to reason about the physical world. They found that BERT is relatively incapable of such reasoning but is able to memorize some properties of real-world objects. This investigation tested common sense spatial reasoning rather than pure factoid knowledge.

Rather than focusing on newer knowledge types, we focus on the true coverage of already known relations and facts in BERT. In terms of experiments, we do not focus on knowledge containment in different language models, rather focus on investigating how knowledge emerges specifically in BERT. Here, we are more interested in relative differences. To this end, we devise a procedure to adapt the layerwise probing methodology often employed for linguistic pattern detection by van Aken et al. (2019); Tenney et al. (2019); Liu et al. (2019) for the probe tasks suggested in Petroni et al. (2019).

3 Experimental Setup

3.1 Models

BERT is a bidirectional text encoder built by stacking several transformer layers. BERT is often pre-trained with two tasks: next sentence classification and masked language modeling (MLM). MLM is cast as a classification task over all tokens in the vocabulary. It is realized by training a decoder that takes as input the mask token embedding and outputs a probability distribution over vocabulary tokens. In our experiments we used BERT base (12 layers) pretrained on the BooksCorpus (Zhu et al., 2015) and English Wikipedia. We use this model for fine-tuning to keep comparisons consistent. Henceforth, we refer to pre-trained BERT as just BERT. The following is a list of all fine-tuned models used in our experiments:

1. NER-CONLL: (cased) named entity recognition model tuned on Conll-2003 (Sang and Meulder, 2003).

2. QA-SQUAD-1: A question answering model (span prediction) trained on SQuAD

1 (Rajpurkar et al., 2016). The trained model achieved an F1 score of 88.5 on the test set.

3. QA-SQUAD-2: QA span prediction trained Squad 2 (Rajpurkar et al., 2018). The F1 score was 67 (note: SQUAD 2 is a more challenging version of SQUAD 1).

4. RANK-MSMARCO: Ranking model trained on the MSMarco passage reranking task (Nguyen et al., 2016). We used the fine-tuning procedure described in (Nogueira and Cho, 2019) to obtain a regression model that predicts a relevance score given query and passage.

5. MLM-MSMARCO: BERT fine-tuned on the passages from the MSMarco dataset using the masked language modeling objective as per (Devlin et al., 2019). 15% of the tokens masked at random.

6. MLM-SQUAD: BERT fine-tuned on text from SQUAD using the masked language modeling objective as per Devlin et al. (2019). 15% of the tokens masked at random.

When fine-tuning, our goal was to not only achieve good performance but also to minimize the number of extra parameters added. More parameters outside BERT may increase the chance of knowledge being stored elsewhere leading to unreliable measurement. We used the Huggingface transformers library (Wolf et al., 2019) for implementing all models in our experiments. More details on hyperparameters and training can be found in the Appendix.

3.2 Knowledge probes

We utilized the existing suite of LAMA knowledge probes suggested in (Petroni et al., 2019)[3] for our experiments. Table 1 briefly summarizes the key details. The probes are designed as cloze statements and limited to single token factual knowledge, i.e., multi-word entities and relations are not included.

Each probe in LAMA is constructed to test a specific relation or type of relational knowledge. ConceptNet is designed to test for general conceptual knowledge since it masks single token objects from randomly sampled sentences whereas T-REx consists of hundreds of sentences for 41

---

[3]https://github.com/facebookresearch/LAMA

Name	#Rels	#Instances	Example	Answer
ConceptNet	-	12514	Rocks are [MASK].	solid
T-REx	41	34017	The capital of Germany is [MASK].	Berlin
Google-RE	3	5528	Eyolf Kleven was born in [MASK].	Copenhagen
Squad	-	305	Nathan Alterman was a [MASK].	Poet

Table 1: Knowledge probes used in the experiments. Petroni et al. (2019) subsampled ConceptNet (Speer and Havasi, 2012), T-REx (ElSahar et al., 2018), Google-RE (Orr, 2013) and Squad (Rajpurkar et al., 2016).

specific relationship types like *member of* and *language spoken*. Google-RE tests for 3 specific types of factual knowledge related to people: place-of-birth (2937), date-of-birth (1825), and place-of-death (766 instances). The date-of-birth is a strict numeric prediction that is not covered by T-REx. Finally, Squad uses context insensitive questions from SQuAD that has been manually rewritten to cloze-style statements. Note that this is the same dataset used to train QA-SQUAD-1 and QA-SQUAD-2.

3.3 Probing Procedure

Our goal is to measure the knowledge stored in BERT via knowledge probes. LAMA probes rely on the MLM decoding head to complete cloze statement tasks. Note that this decoder is only trained for the mask token embedding of the final layer and is unsuitable if we want to probe all layers of BERT. To overcome this we train a new decoding head for each layer of a BERT model under investigation.

Training: We train a new decoding head for each layer the same way as standard pre-training using MLM. We also used Wikipedia (WikiText-2 data) – sampling passages at random and then randomly masking 15% of the tokens in each. Our decoding head uses the same architecture as proposed by Devlin et al. (2019) – a fully connected layer with GELU activation and layer norm (epsilon of 1e-12) resulting in a new 768 dimension embedding. This embedding is then fed to a linear layer with softmax activation to output a probability distribution over the 30K vocabulary terms. In total, the decoding head possesses ∼24M parameters. We froze BERT's parameters and trained the the decoding head only for every layer using the same training data. We initialized the new decoding heads with the parameters of the pre-trained decoding and then fine-tuned it. Our experiments with random initialization yielded no significant difference. We used a batch size of 8 and trained until validation loss was minimized using

the Adam optimizer (Kingma and Ba, 2015). With the new decoding heads, the LAMA probes can be applied to every layer.

Measuring Knowledge We convert the probability distribution output of the decoding head to a ranking with the most probable token at rank 1. The amount of knowledge stored at each layer is measured by precision at rank 1 (P@1 for short). We use P@1 as the main metric in all our experiments. Since rank depth of 1 is a strict metric, we also measured P@10 and P@100. We found the trends to be similar across varying rank depths. For completeness, results for P@10 and P@100 can be found in the appendix. Additionally, we measure the total amount of knowledge contained in BERT by

$$\mathcal{P}@1 = max(\{P^l@1|\ \forall l \in L\})$$

where L is the set of all layers and $P^l@1$ is the P@1 for a given layer l. In our experiments $|L| = 12$. This metric allows us to consider knowledge captured at all layers of BERT, not just a specific layer. If knowledge is always best captured at one specific layer l then $\mathcal{P}@1 = P^l@1$. If the last layer always contains the most information then total knowledge is equal to the knowledge stored in the last layer.

Caveats of probing with cloze statements: Note that BERT, MLM-MSMARCO, and MLM-SQUAD are trained for the task of masked word prediction which is exactly the same task as our probes. The last layers of BERT have shown to contain mostly task-specific knowledge – how to predict the masked word in this case (Kovaleva et al., 2019). Hence, good performance in our probes at the last layers for MLM models can be partially attributed to task-based knowledge.

4 Results

In contrast to existing work, we want to analyze relation knowledge across layers to measure the total knowledge contained in BERT and observe

the evolution of relational knowledge through the layers.

4.1 Intermediate Layers Matter

The first question we tackle is – Does knowledge reside strictly in the last layer of BERT?

Figure 1 compares the fraction of correct predictions in the last layer as against all the correct predictions computed at any intermediate layer in terms of $\mathcal{P}@1$. It is immediately evident that a significant amount of knowledge is stored in the intermediate layers. While the last layer does contain a reasonable amount of knowledge, a considerable proportion of relations seem to be forgotten and the **intermediate layers contain relational knowledge that is absent in the final layer.** Specifically, 18% for `T-REx` and 33% approximately for the others are forgotten by BERTs last layer. For instance, the answer to `Rocky Balboa was born in [MASK]` is correctly predicted as `Philadelphia` by Layer 10 whereas the rank of `Philadelphia` in the last layer drops to 26 for BERT.

The intermediary layers also matter for fine-tuned models. Models with high $\mathcal{P}@1$ tend to have a smaller fraction of knowledge of stored in the intermediate layers – 20% for RANK-MSMARCO on `T-REx`. In other cases, the amount of knowledge lost in the final layer is more drastic – 3× for QA-SQUAD-2 on `Google-RE`.

We also measured the fraction of relationship types in `T-REx` that are better captured in the intermediary layers (Table 2). On average, 7% of all relation types in `T-REx` are forgotten in the last layer for BERT. RANK-MSMARCO forgets the least amount of relation types (2%) whereas QA-SQUAD-1 forgets the most (43%) in `T-REx`, while also being the least knowledgeable (lowest or second-lowest $\mathcal{P}@1$ in all probes). This is further proof of our claim that BERT's overall capacity can be better estimated by probing all layers. Surprisingly, RANK-MSMARCO is able to consistently store nearly all of its knowledge in the last layer. We postulate that for ranking in particular, relational knowledge is a key aspect of the task specific knowledge commonly found in the last layers.

4.2 Relational Knowledge Evolution

Next, we study the evolution of relational knowledge through the BERT layers presented in Figure 2 that reports P@1 at different layers.

We observe that the amount of **relational knowledge captured increases steadily with each additional layer**. While some relations are easier to capture early on, we see an almost-exponential growth of relational knowledge after Layer 8. This indicates that relational knowledge is predominantly stored in the last few layers as against low-level linguistic patterns are learned at the lower layers (similar to van Aken et al. (2019)). In Figure 3 we inspect relationship types that show uncharacteristic growth or loss in `T-REx`.

While `member of` is forgotten in the last layers, the relation `diplomatic relation` is never learned at all, and `official language of` is only identifiable in the last two layers. Note that the majority of relations follow the nearly exponential growth curve of the mean performance in Figure 2 (see line `T-REx`). From our calculations, nearly 15% of relationship types double in mean P@1 at the last layer.

We now analyze evolution in fine-tuned models to understand the impact of fine-tuning on the knowledge contained through the layers. There are two effects at play once BERT is fine-tuned. First, during fine-tuning BERT observes additional task-specific data and hence has either opportunity to monotonically increase its relational knowledge or replace relational knowledge with more task-specific information. Second, the task-specific loss function might be misaligned with the MLM probing task. This means that fine-tuning might result in difficulties in retrieving the actual knowledge using the MLM head. In the following, we first look at the overall results and then focus on specific effects thereafter.

Figure 4 shows the evolution of knowledge in 3 different models when compared to BERT.

All models possess nearly the same amount of knowledge until layer 6 but then start to grow at different rates. Most surprisingly, RANK-MSMARCO's evolution is closest to BERT whereas the other models forget information rapidly. With previous studies indicating that the last layers make way for task-specific knowledge (Kovaleva et al., 2019), the ranking model can retain a larger amount of knowledge when compared to other fine-tuning tasks in our experiments.

These results raise the question: Is RANK-MSMARCO able to retain more knowledge because MSMarco is a bigger dataset or is it because
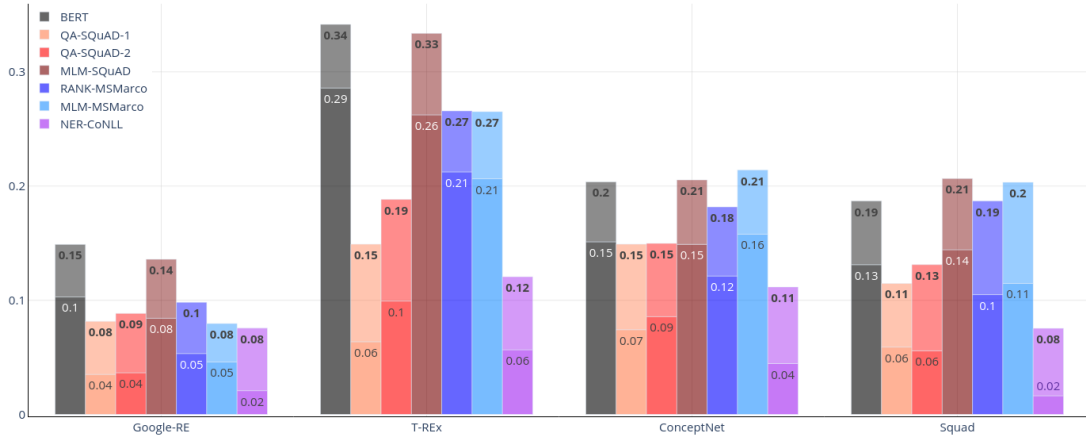
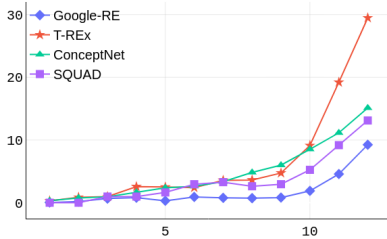Figure 1: $\mathcal{P}@k$ (upper value) vs last layer P@k (lower value) for all models for each LAMA probe.

Figure 2: Mean P@1 of BERT across all layers.
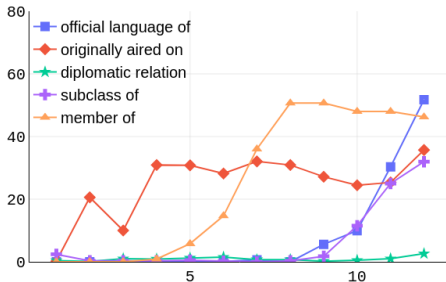

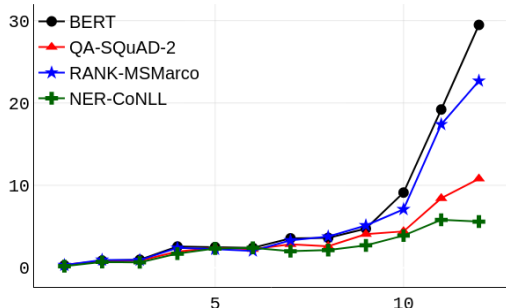Figure 3: P@1 across all layers for BERT for select relationship types from `T-REx`.

Figure 4: Knowledge contained per layer measured in terms of P@1 on `T-REx`.

the ranking objective is better suited to knowledge retention as compared to QA, MLM or NER?

4.3 Effect of fine-tuning data

To isolate the effect of the fine-tuning dataset, we first fix the fine-tuning objective. We experimented with an MLM and a QA span prediction objective. For MLM, we used models trained on fine-tuning task data of varying size – BERT, MLM-MSMARCO ($\sim$ 8.8 million unique passages) and MLM-SQUAD ($\sim$ 500+ unique articles). For the QA objective, we experimented with QA-SQUAD-1 and QA-SQUAD-2 which utilize the same dataset of passages but QA-SQUAD-2 is trained on 50K extra unanswerable questions.

Figure 1 shows the total knowledge and Figure 5 shows the evolution of knowledge for both MLM models as compared to BERT. When fine-tuning, BERT seemingly tends to forget some relational knowledge to accommodate for more domain-specific knowledge. We suspect it forgets certain relations (found in the probe) to make way for other knowledge not detectable by our probes. In the case where the probe is aligned with the fine tuning data (`Squad`), MLM-SQUAD learns more about its domain and outperforms BERT but only by a small margin ($< 5\%$). Even though MLM-MSMARCO uses a different dataset it is able to retain a similar level of knowledge in `Squad`. The evolution trends in Figure 5 further confirm that fine tuning leads to forgetting mostly in the last layers. Since the fine tuning objective and probing tasks are aligned, it is more evident in these experiments that relational knowledge is being forgotten or replaced.

When observing $\mathcal{P}@1$ and $P@1$, according to `T-REx` and `Google-RE` in particular, MLM-MSMARCO forgets a large amount of knowledge but retains common sense knowledge
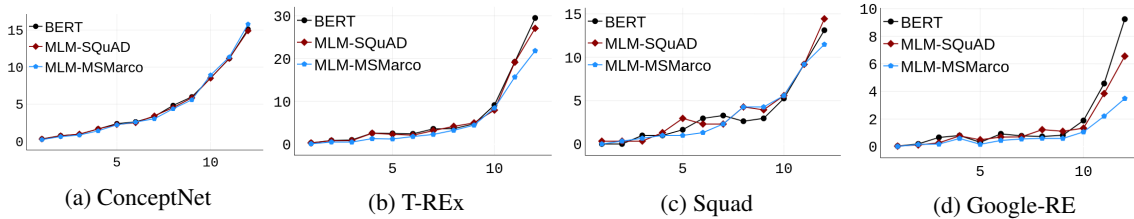
179

Figure 5: Effect of dataset size. Mean P@1 across layers for BERT, MLM-MSMARCO and MLM-SQUAD.
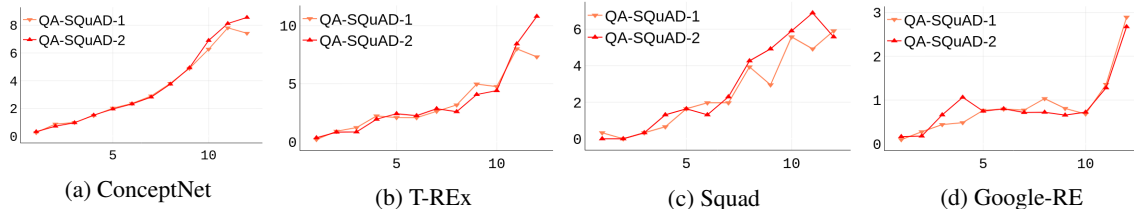

Figure 6: Effect of dataset size. Mean P@1 across layers for QA-SQUAD-1 and QA-SQUAD-2.

(`ConceptNet`). MLM-SQUAD contains substantially more knowledge overall according to 2/4 probes and nearly the same in the others as compared to MLM-MSMARCO. Seemingly, the amount of knowledge contained in fine tuned models is not directly correlated with the size of the dataset. There can be several contributing factors to this phenomenon potentially related to the data distribution and alignment of the probes with the fine tuning data. We leave these avenues open to future work.

Considering the QA span prediction objective, we first see that the total amount of knowledge stored ($\mathcal{P}@1$) in QA-SQUAD-2 is higher for 3/4 knowledge probes (from Figure 1). Figure 6 shows the evolution of knowledge captured for QA-SQUAD-1 vs QA-SQUAD-2. QA-SQUAD-2 captures more knowledge at the last layer in 3/4 probes with both models showing similar knowledge emergence trends. This result hints to the fact that a more difficult task (SQUAD2) on the same dataset forces BERT to remember more relational knowledge in its final layers as compared to the relatively simpler SQUAD1. This point is further emphasized in Table 2. Only 17% of relation types are better captured in the intermediary layers of QA-SQUAD-2 as compared to 43% for QA-SQUAD-1.

4.4 Effect of fine tuning objective

The second effect that we previously discussed is the effect of the task objective function that might be misaligned with the probing procedure. To study this effect, we conducted 2 experiments where we fixed the dataset and compared the MLM objective (MLM-MSMARCO) vs the ranking objective RANK-MSMARCO and MLM-SQUAD vs the span prediction objective (QA-SQUAD-2). Figure 8 shows the evolution of knowledge captured for MLM-MSMARCO vs RANK-MSMARCO.

We observe that RANK-MSMARCO performs quite similar to MLM-MSMARCO across all probes and layers. Although MLM-MSMARCO has the same training objective as the probe, the ranking model can retain nearly the same amount of knowledge. We hypothesize that this is because the downstream fine-tuning task is sensitive to relational information. Specifically, ranking passages for open-domain QA is a task that relies heavily on identifying pieces of knowledge that are strongly related – For example, given the query: *How do you mow the lawn?*, RANK-MSMARCO must effectively identify concepts and relations in candidate passages that are related

Models	P@1	P@10	P@100
BERT	0.07	0.02	0.07
QA-SQUAD-1	0.43	0.38	0.38
QA-SQUAD-2	0.17	0.19	0.17
MLM-SQUAD	0.12	0.07	0.07
RANK-MSMARCO	0.02	0.05	0.05
MLM-MSMARCO	0.10	0.10	0.14
NER-CoNLL	0.26	0.33	0.43

Table 2: Fraction of relationship types (of the 41 T-REx) that are forgotten in the last layer. If mean $P^{12}@1 <$ mean $P^l@1$ for a particular relation type then that relation is considered to be forgotten at the last layer.
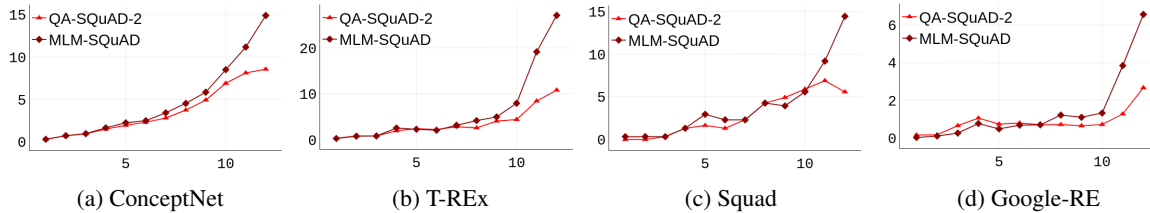
Figure 7: Effect of Fine-Tuning Objective on fixed size data: SQUAD.
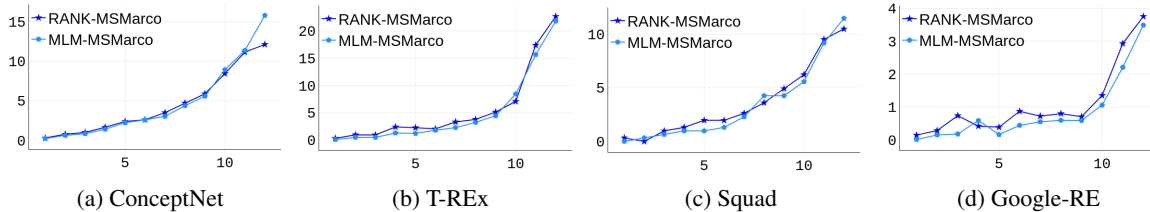

Figure 8: Effect of Fine-Tuning Objective on fixed size data: MSMarco.

to lawn mowing (like types of grass and lawnmowers) to estimate relevance.

Reading comprehension /span prediction (QA) however seems to be a less knowledge-intensive task both in terms of total knowledge and at the last layer (Figure 1). In Figure 7 we see that the final layers are the most impacted here as well. From Table 2 we observe that MLM-SQUAD forgets less in its final layer (12% vs 17%), with QA-SQUAD-2 seemingly forgoing relational knowledge for span prediction task knowledge.

5 Discussion and Conclusion

In this paper, we introduce a framework to probe all layers of BERT for knowledge. We experimented on a variety of probes and fine-tuning tasks and found that BERT contains more knowledge than was reported earlier. Our experiments shed light on the hidden knowledge stored in BERT and also some important implications to model building. Since intermediate layers contain knowledge that is forgotten by the final layers to make way for task-specific knowledge, our probing procedure can more accurately characterize the knowledge stored.

We show that factual knowledge, like syntactic and semantic patterns, is also forgotten at the last layers due to fine-tuning. However, the last layer can also make way for more domain specific knowledge when the fine tuning objective is the same as the pretraining objective (MLM) as observed in Squad. Interestingly, forgetting is not mitigated by larger datasets which potentially con-

tain more factual knowledge (MLM-MSMARCO < MLM-SQUAD as measured by $\mathcal{P}@1$). Instead, we find that knowledge-intensive tasks like ranking do mitigate forgetting compared to span prediction. Although the fine-tuned models always contain less knowledge, with significant (and expected) forgetting in the last layers, RANK-MSMARCO remembers relatively more relationship types than BERT (2% vs 7% forgotten) in its last layer (Table 2). This result can partially explain findings in Chang et al. (2019) where they found that pretraining BERT with inverse cloze tasks aids it's transferability to a retrieval and ranking setting. Essentially, ranking tasks encourage the retention of factual knowledge (as measured by cloze tasks) since they are seemingly required for reasoning between the relative relevance of documents to a query.

Our results have direct implications on the use of BERT as a knowledge base. By effectively choosing layers to query and adopting early exiting strategies knowldge base completion can be improved. The performance of RANK-MSMARCO also warrants further investigation into ranking models with different training objectives – pointwise (regression) vs pairwise vs listwise. More knowledge-intensive QA models like answer generation models may also show a similar trend as ranking tasks but require investigation. We also believe that our framework is well suited to studying variants of BERT architecture and pretraining methods.

References

Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. 2019. How does BERT answer questions?: A layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 1823–1832. ACM.

Yonatan Belinkov and James R. Glass. 2019. Analysis methods in neural language processing: A survey. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3348–3354. Association for Computational Linguistics.

Wei-Cheng Chang, X Yu Felix, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2019. Pre-training tasks for embedding-based large-scale retrieval. In *International Conference on Learning Representations*.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel Gershman, and Noah D. Goodman. 2018. Evaluating compositionality in sentence embeddings. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society, CogSci 2018, Madison, WI, USA, July 25-28, 2018*. cognitivesciencesociety.org.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Hady ElSahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique 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, May 7-12, 2018*. European Language Resources Association (ELRA).

Allyson Ettinger, Ahmed Elgohary, Colin Phillips, and Philip Resnik. 2018. Assessing composition in sentence vector representations. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1790–1801, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. Do neural language representations learn physical commonsense? In *Proceedings of the 41th Annual Meeting of the Cognitive Science Society, CogSci 2019: Creativity + Cognition + Computation, Montreal, Canada, July 24-27, 2019*, pages 1753–1759. cognitivesciencesociety.org.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287.

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.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7811–7818. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4364–4373. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1073–1094. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3428–3448. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings*

of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *CoRR*, abs/1901.04085.

Dave Orr. 2013. 50,000 lessons on how to read: a relation extraction corpus. *Online: Google Research Blog*, 11.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1499–1509. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *CoRR*, abs/1806.03822.

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, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how BERT works. *CoRR*, abs/2002.12327.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147. ACL.

R. 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 2012, Istanbul, Turkey, May 23-25, 2012*, pages 3679–3686. European Language Resources Association (ELRA).

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4593–4601. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society.

6 Appendix

6.1 Models

- BERT: Off the shelf "bert-base-uncased" from the huggingface transformers library (Wolf et al., 2019)

- QA-SQUAD-1: Both SQuAD QA models are trained with the huggingface question answering training script [4]. This adds a span prediction head to the default BERT, I.e. a linear layer that computes logits for the span start and span end. So for a given question and a context, it classifies the indices in in which the answer starts and ends. As a loss function it uses crossentropy. The model was trained on a single GPU. We used the huggingface default training script and standard parameters: 2 epochs, learning rate 3e-5, batch size 12.

- QA-SQUAD-2: Single GPU, also using huggingface training script with standard parameters. Learning rate was 3e-5, batch size 12, best model after 2 epochs.

- MLM-SQUAD: Fine tuned on text from SQUAD using the masked language modeling objective as per (Devlin et al., 2019). 15% of the tokens masked at random. Trained for 4 epochs with LR 5e-5. Single GPU.

- RANK-MSMARCO: Trained as described in (Nogueira and Cho, 2019). MSMARCO, 100k iterations with batch size 128 (on a TPUv3-8).

- MLM-MSMARCO: 15% of the tokens masked at random. 3 epochs, batch size 8, LR 5e-5. Single gpu.

6.2 Experimental results:

- Computing infrastructure used: Everything can be run in Colab notebook with 12gb of RAM and the standard GPU. The experiments, however, have been run on a computing cluster with 6 nodes. Every node had 4 gtx 1080ti and 128gb RAM. Thus being able to parallize the probing of different layers.

---

[4]https://github.com/huggingface/transformers

- Average runtime: Circa 3 hours per layer (that is training the MLM head and probing the LAMA probes) on a single GPU.

- Number of parameters: Since we use standard BERT, the base model + MLM head combined have 110,104,890 parameters. The MLM head itself has 24,459,834 parameters.

- Validation performance for test results: Since we probed the data, we could not do validation on it.

- Explanation of evaluation metrics used with links to code: It is done in knowledge_probing/probing/metrics.py. But the one that we use are Precision @ k where we just check if the model predicts the correct token at index $<=$ k (P@k)

6.3 Hyperparameter seach:

Not applicable.

6.4 Datasets:

- Wikitext-2: Used for fine-tuning the MLM head. Subset of the Englisch Wikipedia for long term dependency language modeling. 2,088,628 tokens for training, 217.646 for validation, 245.569 for testing. Vocab size: 33,278 out of vocab: 2.6% of tokens. It can be downloaded from here: https://www.salesforce.com/products/einstein/ai-research/the-wikitext-dependency-language-modeling-dataset/

- LAMA probe data: Can be downloaded from their github: https://github.com/facebookresearch/LAMA . Only used for testing. Consists of: Google-RE: 5528 instances over 3 relations. T-REx: 34017 instances over 41 relations. ConceptNet: 12514 instances. This is not grouped into relations. Squad: 305 instances. Context in-sensitive questions rewritten to cloze-statements. No specific relation either.

- SQuAD 1.1: Can be downloaded from here: https://rajpurkar.github.io/SQuAD-explorer/ . 100,000+ question answer pairs based on wikipedia articles. Produced by crowdworkers.

- SQuAD 2: Can be downloaded from here: https://rajpurkar.github.io/SQuAD-explorer/

. Combines the 100,000+ question answer pairs with 50,000 unanswerable questions.

- MSMARCO: Can be downlaoded from here: https://microsoft.github.io/msmarco/ . For ranking: Dataset for passage reranking was used. Given 1,000 passages, re-rank by relevance. Dataset contains 8,8m passages. For MLM training: Dataset for QA was used. It consists of over 1m queries and the 8,8m passages. Each query has 10 candidate passages. For MLM, we appended the queries with all candidate passages before feeding into BERT.

6.5 Knowledge captured in BERT

6.5.1 Intermediate Layers Matter

Additional precisions for Figure 4 can be found in Figure 9.

6.5.2 Relational Knowledge Evolution

Additional precisions for Figure 2 can be found in Figure 10.

6.5.3 Effect of dataset size

Figure 11 and 12 show the P@10 and P@100 plots for Figure 5. Respectively, Figure 13 and 14 show the same for 6.

6.6 Effect of fine tuning objective

For comparing MLM and QA on SQuAD (7), Figure 15 and 16 show more precisions. Also, for comparing fine tune objectives on MSMARCO (Figure 8), Figure 17 and 18 show P@10 and P@100.
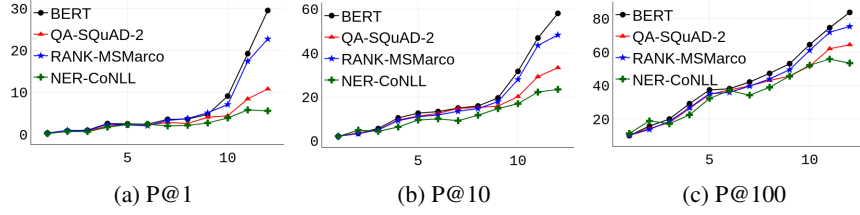
(a) P@1   (b) P@10   (c) P@100

Figure 9: Mean performance in different precisions on T-REx sets for BERT, QA-SQUAD-2, RANK-MSMARCO, NER-CONLL.

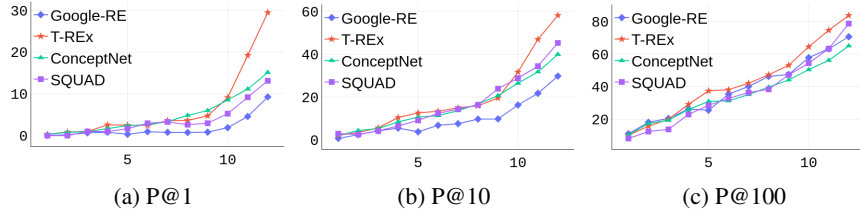(a) P@1   (b) P@10   (c) P@100

Figure 10: Mean performance of BERT across all layers and probe sets.

(a) ConceptNet   (b) T-REx   (c) Squad   (d) Google-RE
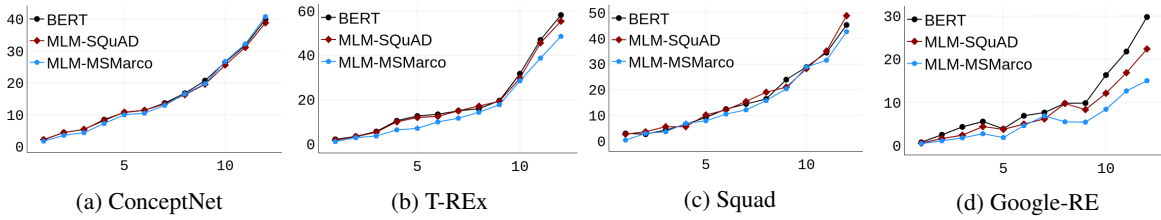
Figure 11: Effect of dataset size. Showing P@10

(a) ConceptNet   (b) T-REx   (c) Squad   (d) Google-RE
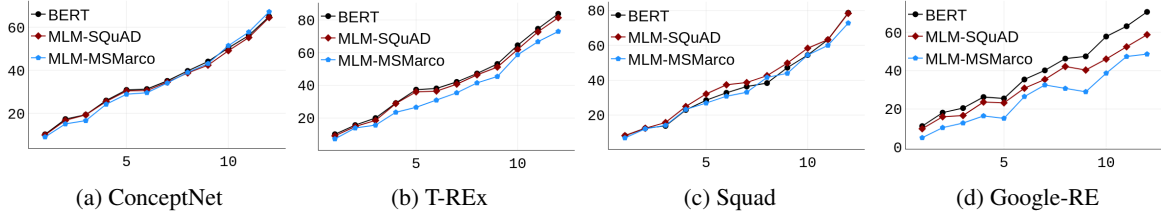
Figure 12: Effect of dataset size. Showing P@100

(a) ConceptNet   (b) T-REx   (c) Squad   (d) Google-RE
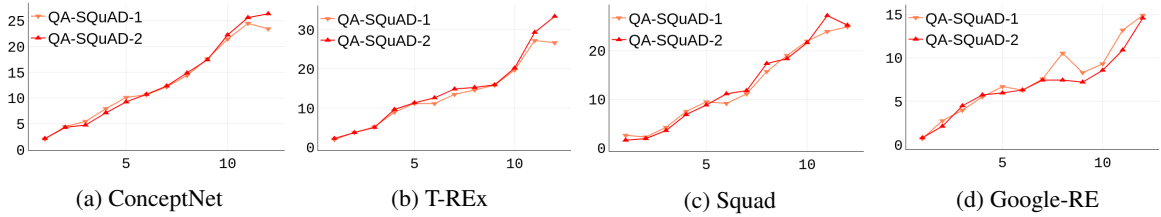
Figure 13: Effect of dataset size. Showing P@10 for the QA objective.

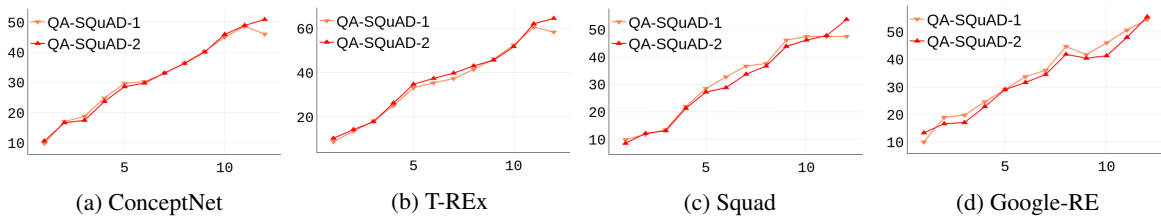(a) ConceptNet   (b) T-REx   (c) Squad   (d) Google-RE

Figure 14: Effect of dataset size. Showing P@100 for the QA objective.
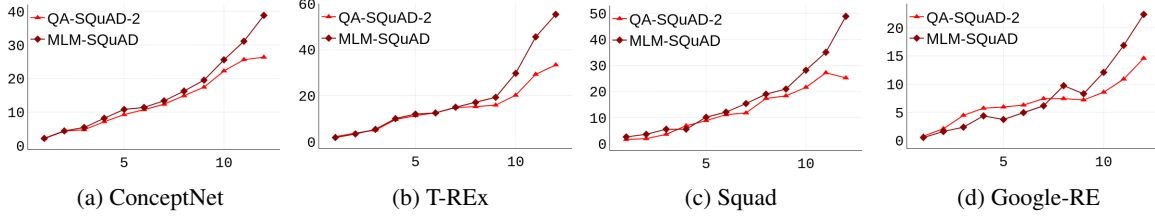
Figure 15: Effect of Fine-Tuning Objective on fixed size data: SQUAD. Showing P@10.
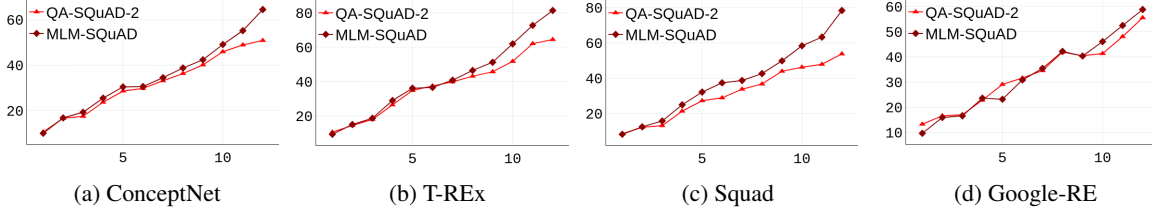

Figure 16: Effect of Fine-Tuning Objective on fixed size data: SQUAD. Showing P@100.
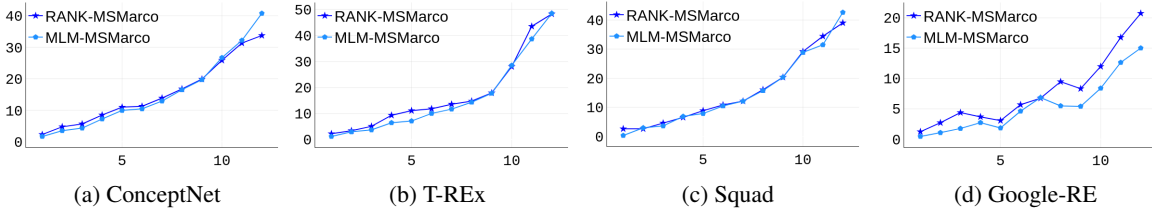

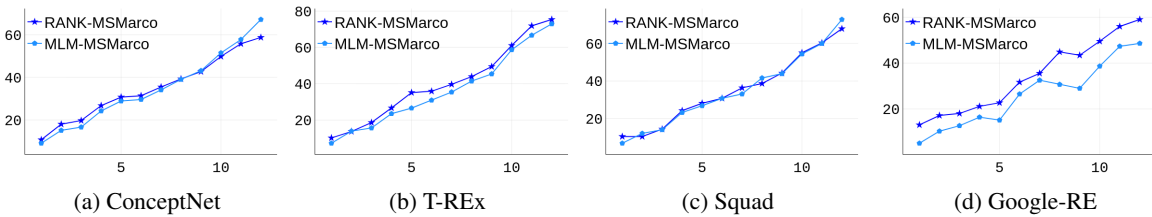Figure 17: Effect of Fine-Tuning Objective on fixed size data: MSMarco. Showing P@10.

Figure 18: Effect of Fine-Tuning Objective on fixed size data: MSMarco. Showing P@100.

Model	Google-RE		T-REx		ConceptNet		Squad	
	P@1	$\mathcal{P}$@1	P@1	$\mathcal{P}$@1	P@1	$\mathcal{P}$@1	P@1	$\mathcal{P}$@1
BERT	10	**15**	29	**34**	15	**21**	13	**20**
QA-SQuAD-1	3	**9**	6	**15**	7	**15**	5	**15**
QA-SQuAD-2	3	**9**	10	**19**	8	**16**	6	**13**
MLM-SQuAD	4	**10**	15	**23**	9	**16**	6	**16**
RANK-MSMarco	6	**11**	23	**29**	12	**19**	10	**20**
MLM-MSMarco	3	**7**	14	**21**	11	**17**	7	**12**

Table 3: Mean knowledge contained in the last layer (P@1) vs knowledge contained in all layers ($\mathcal{P}$@1) for each probe.