The Effectiveness of Masked Language Modeling and Adapters for Factual Knowledge Injection

Sondre Wold
University of Oslo

Abstract

This paper studies the problem of injecting factual knowledge into large pre-trained language models. We train adapter modules on parts of the ConceptNet knowledge graph using the masked language modeling objective and evaluate the success of the method by a series of probing experiments on the LAMA probe. Mean P@K curves for different configurations indicate that the technique is effective, increasing the performance on subsets of the LAMA probe for large values of *k* by adding as little as 2.1% additional parameters to the original models.

1 Introduction

Large pre-trained language models (PLMs) are difficult to interpret due to their complexity and large parameter size. This can partly be explained by the nature of popular training regimens, such as the masked language modelling objective, which encodes distributional knowledge. Such regimens have proven effective for a range of downstream NLP tasks, but they also make it difficult to determine and validate the origin of whatever knowledge the models end up with.

Consequently, there have been multiple efforts to integrate structured information into PLMs (Peters et al., 2019; Yasunaga et al., 2021; Kaur et al., 2022). This has not only been motivated by the promise of better interpretability, but also the observation that there exist scenarios where we would want to stress information that might not be so easily encoded by modelling long range dependencies between fragments of text. This includes knowledge intensive tasks where employing the correct factual knowledge is crucial, for example within the medical domain (Zhang et al., 2021) and question answering (Zhang et al., 2022). At the same time, there exist multiple structured sources that attempt to capture factual knowledge. These sources range from domain specific knowledge graphs for medical information (Shi et al., 2017), commonsense graphs like Yago or ConceptNet (Suchanek et al., 2007; Speer et al., 2017), to lexico-semantic networks like WordNet (Miller, 1995).

In this paper, we attempt to inject the structured information found in the ConceptNet knowledge graph (Speer et al., 2017) into pre-trained language models. The injection is done by training relatively small neural networks, known as adapter modules (Houlsby et al., 2019; Pfeiffer et al., 2020), on subject—predicate—object triples. As in Lauscher et al. (2020), we extract the triples using a random walk procedure and then translate them into natural language so that we can use masked language modeling as the training objective. The resulting adapters are injected into all layers of two popular pre-trained language models: BERT base (Devlin et al., 2019) and RoBERTa base (Liu et al., 2019). Our code and data is made publicly available[1].

For the injection to be deemed effective, we argue that the adapter-injected models must be able to use the knowledge gained from the adapter training together with what the models learned during their initial pre-training. In order to quantitatively assess this, we evaluate our models in a zero-shot setting on the ConceptNet subset of the LAMA probe (Petroni et al., 2019). As ConceptNet is the source for both our training corpus and the LAMA probe, we can better measure how much of the factual knowledge seen during adapter training the models can be expected to recall.

2 Related work

Combining structured information with language models is a standing problem in NLP. One approach to overcome this has been to combine knowledge graphs with PLMs, augmenting the distributional knowledge encoded in the models with the structured information found in the graphs (Sun et al.,

[1] https://github.com/SondreWold/adapters-mlm-injection

Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing, pages 54–59
October 16, 2022.

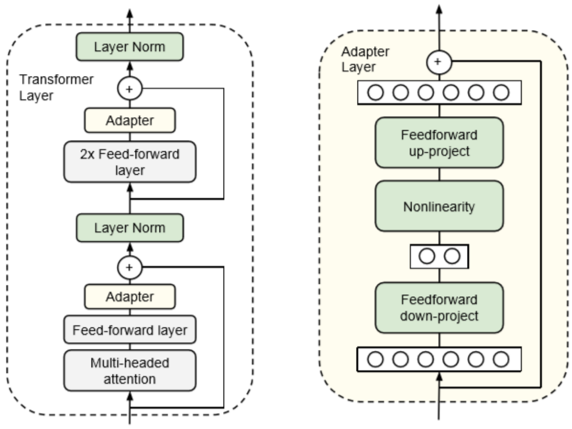


Figure 1: Left: how adapters are injected into each transformer layer. Right: the components of each adapter module. Figure from Houlsby et al. (2019).

2021; Liu et al., 2020; Wang et al., 2021). Within this approach, we find several uses of adapters. First introduced for NLP by Houlsby et al. (2019), and popularized by the AdapterHub framework Pfeiffer et al. (2020), adapters are small neural networks injected into larger, often pre-trained models. During training the original model weights are kept static, and only the set of newly introduced weights from the adapter are adjusted. Figure 1 illustrates the architecture proposed by Houlsby et al. (2019) and how it is injected into a transformer layer.

The methodology in this paper is inspired by Lauscher et al. (2020), who inject commonsense information and world knowledge into BERT by using such adapter modules. As in our work, the adapters train with the masked language modeling objective over subject—predicate—object triples from the ConceptNet graph, but they are evaluated on the GLUE benchmark (Wang et al., 2018). Although the result are inconclusive for most of the tasks in GLUE, the injected models perform better than their base model counterparts on the world knowledge and commonsense categories of the diagnostic set.

A similar approach is taken by Wang et al. (2021). Their K-DAPTER model has one adapter for factual knowledge, trained on aligned text triplets from Wikipedia and Wikidata, and one for linguistic knowledge, obtained via dependency parsing. Results on knowledge-driven tasks, including relation classification, entity typing, and question answering, show that this setup improves performance, and furthermore, that K-ADAPTER captures more versatile knowledge than ROBERTA.

In a more domain specific context, Meng et al. (2021) use adapter modules to infuse a large biomedical knowledge graph into an underlying BERT model. By partitioning the large graph into smaller sub-graphs, which are then fed into distinct adapter modules and fused using a mixture layer that combine the knowledge from all the adapters using an attention layer, they achieve a new state-of-the-art performance on five domain specific datasets.

3 Experiments

Following Lauscher et al. (2020), we use the same configuration for our adapter modules as in Houlsby et al. (2019). We set the size of the adapter modules to 64, which implies a reduction factor of 12 from the original transformer layer size of 768 in BERT$_{\text{BASE}}$. This increases the total amount of parameters by 2.1%. We use GELU (Hendrycks and Gimpel, 2020) as the activation function inside the adapters, and the Adam optimizer from (Kingma and Ba, 2017). We set the learning rate to *1e-4* with 10.000 warm-up steps and weight decay factor of 0.01. We allow the adapter to train for 100.000 optimization steps while freezing all the original transformer weights. The adapters are implemented using the `adapter-transformers` library (Pfeiffer et al., 2020). Throughout the remainder of this paper, the resulting configuration is referred to as CN$_{\text{Houlsby 100k}}$ in figures and as the Houlsby configuration in text.

The adapters train on the same subset of ConceptNet as in Lauscher et al. (2020). As this study was named Retrograph, we refer to this particular set of predicate types as the Retrograph predicate set. The predicates in this set are: ANTONYMOF, SYNONYMOF, ISA and MANNEROF. Subject—predicate—object triplets with one of these predicates in their middle position are extracted through a random traversal procedure[2] and then subsequently chained so that we get blocks of text in natural language on the following format:

possible is a synonym of possibility.
possibility is a concept.
concept is a synonym of conception.
conception is a synonym of fertilization.
fertilization is a enrichment.
enrichment is a gift.

[2] Details on this traversal procedure can be found in Lauscher et al. (2020) or in appendix A.

The corpus is processed using masked language modeling (MLM), parsed line for line with a MLM probability of 0.15, as in the original BERT paper (Devlin et al., 2019). We also experiment by training on the corpus by a maximum sequence length instead of line by line training. However, this did not affect the performance of the models in any significant way.

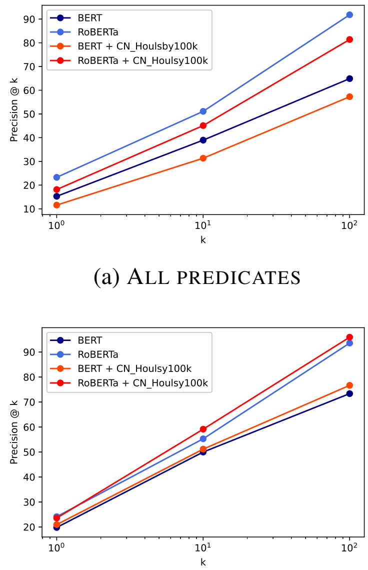
3.1 Evaluation

We evaluate our injected models on the ConceptNet split of the LAMA (LAnguage Model Analysis) probe (Petroni et al., 2019), which allows for testing of the factual and commonsense knowledge of language models. Facts are presented as fill-in-the-blank cloze statements, e.g: "Ibsen was born in [MASK] in the year 1828", and models are ranked based on how highly it ranks the ground truth token. All models are evaluated in a zero-shot setting, using the same prediction head as in their pre-training.

As we train our adapter modules on ConceptNet and also evaluate on the ConceptNet split from LAMA, it is important to note that what we test here is not the model's ability to generalize on unseen data in the traditional sense, but whether or not they are able to reproduce the factual information extracted from the knowledge graph during adapter training. The phrasing of the cloze statements in LAMA is not the same as in the training corpus for the adapters, although fairly similar. For example, one sentence in LAMA derived from the source triple `communicating hasSubevent knowledge` is presented in the probe as *Communicating is for gaining [MASK]*, while the same triple would be phrased as *communication has subevent knowledge* in the training corpus for the adapters. This makes it possible to control the degree of overlap between instances of factual knowledge in the training corpus and the concepts at the object position in the statements from LAMA. The degree of overlap is numerically specified in the discussion of each result.

3.1.1 Evaluation metric

Following Petroni et al. (2019), we use mean precision at different values of k as the evaluation metric over the LAMA resource. Normally, as in information retrieval, we calculate the precision of a retrieved collection as the number of relevant documents proportionate to the total number of retrieved documents. Here, however, we only have one true positive for collections of all sizes. Thus, the mean precision at various values for k is equal to the whether or not the correct word is a member of the set of predictions of size k. If $k = 100$, we return a precision of 1 if the correct word is one of the top 100 predictions.



(a) ALL PREDICATES

(b) THE ISA PREDICATE

Figure 2: Mean P@k curve for base models and the Houlsby adapter configuration. Base 10 log scale for the X axis. **a)** shows the result for all the predicates in the ConceptNet split of LAMA while **b)** shows results for the "ISA" predicate only

4 Results

Figure 2 shows the mean P@K curves for two language models, with and without an adapter. Part *(a)* of the figure shows the result over all the predicate types present in the ConceptNet split of LAMA ($N = 29774$). The injection of the adapter module decreases the performance of both BERT and ROBERTA for all values of k. However, the corpus with the Retrograph predicate set that adapters trained on only includes one of these types. Hence, there is little similarity between the two sets, and the reproduction of factual knowledge cannot be expected here. This also indicate that training on one set of predicate types does not improve the reproduction of facts on others.

Part *(b)* of figure 2, on the other hand, shows the same models and adapters, but with the probe restricted only to the ISA predicate type — which is then present both in the training corpus and in the

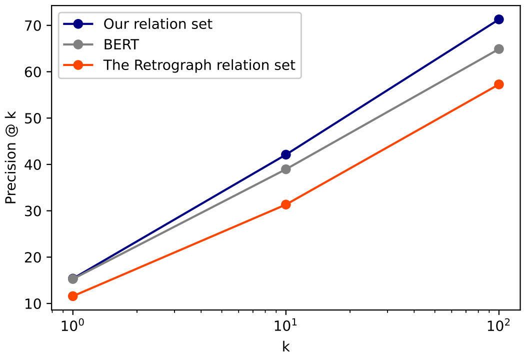


Figure 3: The result of different training configurations on the ConceptNet split of LAMA (Petroni et al., 2019). The two models, in dark blue and orange, use BERT$_{\text{BASE}}$ as the root model and the Houlsby configuration for their adapter, but are trained on different predicate sets of the ConceptNet graph. The gray line represents a BERT model without any adapter training.

probe. In the corpus from Lauscher et al. (2020), triples with this predicate type make up 23% of the total corpus ($N = 69843$).

Since both resources are extracted from ConceptNet, we check the overlap between the masked tokens in the object position in LAMA and the object position in the triplets in the training set for the adapters. The actual percentage will depend on the random walk procedure, but for the sets used in figure 2 there is a 5.7% overlap between concepts. That is, approximately five percent of the concepts from LAMA that the models are expected to predict are also in the training corpus in some form, either with the same predicate type as in the probe, IsA, or one of the others in the Retrograph set.

Despite this, the injected models perform consistently better. As this performance gain is achieved by adding only 2.1% additional parameters to the original model, and without adjusting the original weights at all, we interpret the results as an indication that this method of knowledge injection is effective.

5 Changing the predicate set

In order to further probe the effectiveness of the proposed method, we introduce a new corpus ($N = 99603$ triples) — distilled with the same random walk procedure, but over a new set of predicate types, namely the same set of predicate types found in the ConceptNet split of LAMA. By intuition, if the method is effective, the injected models should score higher on average over all these predicate types than their non-injected counterparts. A list of these predicate types can be found in appendix A.

Figure 3 compares the result of the injected models trained over our predicate set with that of the Retrograph set and a plain BERT model for different values of k. As can be seen from the P@K curves, models trained over our predicate set improve the performance on the full ConceptNet split of the LAMA (N= 29774) probe by up to 6.39% for BERT at large values of k. For k=1, where the model must guess the correct masked object "at first try", we see little difference. Compared to the Retrograph set, which has fewer predicate types, the difference in performance indicate that predicate type specificity is important (e.g subgraph quality). For this comparison, the overlap between the training corpus for the adapters and the full ConceptNet split of LAMA is 36% on the object level, meaning that roughly one third of the concepts were seen during training in some form.

This provides some evidence for the success of the knowledge injection. Models are able to reproduce factual knowledge when queried over the LAMA probe, even though the phrasing of the questions in LAMA is different than the strict triplet-style of the training corpus.

6 Conclusion and Future Work

Combining structured information and large pre-trained language models is a standing problem in NLP research. In this work, we show that training adapter modules on triplets extracted from ConceptNet using masked language modeling can help language models reproduce factual knowledge. Experiments on the ConceptNet split of the LAMA probe show that our adapter-injected models perform better in a zero-shot setting than non-injected models, having seen only a third of the relevant factual knowledge during pre-training in some form, encoded into only 2.1% of the total parameters of the total model. Future work should investigate how this type of knowledge injection can augment language models on other types of tasks, such as language generation, multiple choice questions or natural language inference, which would require more fine-grained annotations of downstream tasks targeted at some form of knowledge.

References

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.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks.

Dan Hendrycks and Kevin Gimpel. 2020. Gaussian error linear units (gelus).

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Jivat Kaur, Sumit Bhatia, Milan Aggarwal, Rachit Bansal, and Balaji Krishnamurthy. 2022. LM-CORE: Language models with contextually relevant external knowledge. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 750–769, Seattle, United States. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Anne Lauscher, Olga Majewska, Leonardo F. R. Ribeiro, Iryna Gurevych, Nikolai Rozanov, and Goran Glavaš. 2020. Common sense or world knowledge? investigating adapter-based knowledge injection into pretrained transformers. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 43–49, Online. Association for Computational Linguistics.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2901–2908.

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

Zaiqiao Meng, Fangyu Liu, Thomas Clark, Ehsan Shareghi, and Nigel Collier. 2021. Mixture-of-partitions: Infusing large biomedical knowledge graphs into BERT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4672–4681, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

George A Miller. 1995. Wordnet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

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

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Longxiang Shi, Shijian Li, Xiaoran Yang, Jiaheng Qi, Gang Pan, and Binbin Zhou. 2017. Semantic health knowledge graph: semantic integration of heterogeneous medical knowledge and services. *BioMed research international*, 2017.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Yueqing Sun, Qi Shi, Le Qi, and Yu Zhang. 2021. Jointlk: Joint reasoning with language models and knowledge graphs for commonsense question answering.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Taolin Zhang, Zerui Cai, Chengyu Wang, Minghui Qiu, Bite Yang, and Xiaofeng He. 2021. SMedBERT: A knowledge-enhanced pre-trained language model with structured semantics for medical text mining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5882–5893, Online. Association for Computational Linguistics.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022. GreaseLM: Graph REASoning enhanced language models. In *International Conference on Learning Representations*.

A Appendix

The ConceptNet split of the LAMA probe includes the following predicate types:

atLocation, capableOf, causes, causesDesire, desires, hasA, hasPrerequisite, hasProperty, hasSubevent, isA, locatedNear, madeOf, motivatedByGoal, partOf, receivesAction, usedFor.

A.1 Random walk procedure

Retrograph uses the weighted random walk algorithm from NODE2VEC (Grover and Leskovec, 2016) in order to extract the `subject--predicate--object` triples from ConceptNet. The pseudocode from the original publication on this algorithm is presented below. The alias method refers to a way of sampling from a discrete probability distribution.[3]

Algorithm 1 The random walk procedure from Lauscher et al. (2020)

```
procedure NODE2VECWALK(Graph G' = (V, E, π), Start node u, Length l)
    $Inititalizewalkto[u]$
    for walk_iter = 1 to l do
        $curr = walk[-1]$
        $V_{curr} = GetNeighbors(curr, G')$
        $s = AliasSample(V_{curr}, \pi)$
        $Appendstowalk$
    return walk
```

[3] https://lips.cs.princeton.edu/the-alias-method-efficient-sampling-with-many-discret