# K-ADAPTER: Infusing Knowledge into Pre-Trained Models with Adapters

Ruize Wang [* 1]  Duyu Tang [2]  Nan Duan [2]  Zhongyu Wei [3]  Xuanjing Huang [4]  Jianshu ji [5]  Guihong Cao [5]
Daxin Jiang [6]  Ming Zhou [2]

## Abstract

We study the problem of injecting knowledge into large pre-trained models like BERT and RoBERTa. Existing methods typically update the original parameters of pre-trained models when injecting knowledge. However, when multiple kinds of knowledge are injected, they may suffer from catastrophic forgetting. To address this, we propose K-ADAPTER, which remains the original parameters of the pre-trained model fixed and supports continual knowledge infusion. Taking RoBERTa as the pre-trained model, K-ADAPTER has a neural adapter for each kind of infused knowledge, like a plug-in connected to RoBERTa. There is no information flow between different adapters, thus different adapters are efficiently trained in a distributed way. We inject two kinds of knowledge, including factual knowledge obtained from automatically aligned text-triplets on Wikipedia and Wikidata, and linguistic knowledge obtained from dependency parsing. Results on three knowledge-driven tasks (total six datasets) including relation classification, entity typing and question answering demonstrate that each adapter improves the performance, and the combination of both adapters brings further improvements. Probing experiments further show that K-ADAPTER captures richer factual and commonsense knowledge than RoBERTa.

## 1. Introduction

Language representation models, which are pre-trained on large-scale text corpus through unsupervised objectives like (masked) language modeling, such as BERT (Devlin et al., 2019), GPT (Radford et al., 2018; 2019), XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019) and T5 (Raffel et al., 2019), have established state-of-the-art performances on various NLP downstream tasks.

Despite the huge success of these large pre-trained models in emperical studies, recent studies suggest that models learned in such an unsupervised manner struggle to capture rich knowledge. For example, Poerner et al. (2019) suggest that although language models do well in reasoning about the surface form of entity names, they fail in capturing rich factual knowledge. Kassner & Schütze (2019) observe that BERT mostly did not learn the meaning of negation (e.g. "not"). Talmor et al. (2019) find that language models fail completely on half of eight reasoning tasks that require symbolic operations such as comparison, conjunction, and composition. These observations motivate us to study the injection of knowledge into pre-trained models like BERT and RoBERTa.

Recently, some efforts have been made to exploit injecting knowledge into pre-trained language models (Zhang et al., 2019; Lauscher et al., 2019; Levine et al., 2019; Peters et al., 2019; He et al., 2019; Xiong et al., 2020). Most previous works (as shown in Table 1) augment the standard language modeling objective with knowledge-driven objectives and update model parameters in a multi-task learning manner. Although these methods, with updated pre-trained models, obtain better performance on downstream tasks, they fail to continual learning (Kirkpatrick et al., 2017). Model parameters need to be retrained when we want to inject many new kinds of knowledge, which may result in the catastrophic forgetting of previously injected knowledge. Meanwhile, the resulting pre-trained models produce entangled representations, which makes it hard to investigate the effect of each knowledge when multiple kinds of knowledge are injected.

In this paper, we propose K-ADAPTER, a flexible and simple approach that infuses knowledge into large pre-trained models. K-ADAPTER has attractive properties including supporting continual knowledge infusion and producing disentangled representations. It remains the original representation of a pre-trained model unchanged and exports different representations for different types of infused knowledge. This is achieved by the integration of compact neural models, dubbed adapters here. Adapters are knowledge-specific models plugged outside of a pre-trained model, whose inputs are the output hidden-states of intermediate layers of

*Work is done during internship at Microsoft Research Asia
[1] Academy for Engineering and Technology, Fudan University, Shanghai, China [2] Microsoft Research Asia, Beijing, China [3] School of Data Science, Fudan University, Shanghai, China [4] School of Computer Science, Fudan University, Shanghai, China [5] Microsoft AI and Research, Redmond WA, USA [6] Microsoft Search Technology Center Asia, Beijing, China.

*Table 1.* Comparison between our approach (K-ADAPTER) and previous works on injecting knowledge into BERT.

| Model | Knowledge Source | Objective | BERT fixed in training? | Continual knowledge infusion? |
|---|---|---|---|---|
| ERNIE (Zhang et al., 2019) | Wikipedia, WikiData | entity linking | N | N |
| LIBERT (Lauscher et al., 2019) | WordNet | synonym word prediction, hyponym-hypernym prediction | from scratch | N |
| SenseBERT (Levine et al., 2019) | WordNet | word-supersense prediction | from scratch | N |
| KnowBERT (Peters et al., 2019) | Wordnet, Wikipedia, CrossWikis | entity linking , hypernym linking | N | N |
| WKLM (Xiong et al., 2020) | WikiPedia, WikiData | replaced entity detection | N | N |
| BERT-MK (He et al., 2019) | Unified Medical Language System | discriminate between real and fake facts | N | N |
| K-Adapter (this work) | Wikipedia, Wikidata, dependency parser | predication prediction, dependency relation prediction | Y | Y |

the pre-trained model. In this work, we take RoBERTa (Liu et al., 2019) as the base pre-trained model and integrate two types of knowledge, including factual knowledge obtained by aligned Wikipedia text to Wikidata triplets, linguistic knowledge obtained by applying off-the-shell dependency parser to web texts. In the pre-training phase, we train two adapters independently on relation classification task and dependency relation prediction task, respectively, while remaining the original parameters of RoBERTa frozen. Since adapters have much less trainable parameters compared with RoBERTa, the training process is memory efficient.

We conduct extensive experiments on six benchmark datasets across three knowledge-driven tasks, i.e., relation classification, entity typing and question answering. Experiments show that K-ADAPTER consistently performs better than RoBERTa, and achieves state-of-the-art performance on five datasets, and comparable performance compared with CosmosQA SOTA. We further conduct probing experiments on LAMA (Poerner et al., 2019) and LAMA-UHN (Petroni et al., 2019), demonstrating that K-ADAPTER captures richer factual and commonsense knowledge than RoBERTa.

The contributions of this paper are summarized as follows:

- We present K-ADAPTER, a flexible approach that supports continual knowledge infusion into large pre-trained models (e.g. RoBERTa in this work).
- We infuse factual knowledge and linguistic knowledge, and show that adapters for both kinds of knowledge work well on downstream tasks.
- K-ADAPTER achieves state-of-the-art or comparable performance by fine-tuning parameters on three downstream tasks, and captures richer factual and commonsense knowledge than RoBERTa on probing experiments.

## 2. Related Work

Our work relates to the area of injecting knowledge into pre-trained models such as BERT. As stated in Table 1, previous works mainly differ from the knowledge sources and the objective used for training.

**ERNIE** (Zhang et al., 2019) injects a knowledge graph into BERT. They align entities from Wikipedia sentences to fact triples in WikiData, and discard sentences with less than three entities. In the training process, the input includes sentences and linked facts, and the knowledge-aware learning objective is to predict the correct token-entity alignment. Entity embeddings are trained on fact triples from WikiData via TransE (Bordes et al., 2013). **LIBERT** (Lauscher et al., 2019) injects pairs of words with synonym and hyponym-hypernym relations in WordNet. The model takes a pair of words separated by a special token as the input, and is optimized by a binary classification problem, which predicts whether the input holds a particular relation or not. **Sense-BERT** (Levine et al., 2019) considers word-supersense knowledge. It inject knowledge by predicting the supersense of the masked word in the input, where the candidates are nouns and verbs and the ground truth comes from WordNet. **KnowBERT** (Peters et al., 2019) incorporates knowledge bases into BERT using Knowledge attention and recontextualization, where the knowledge comes from synset-synset and lemma-lemma relationships in WordNet, and entity linking information in Wikipedia. If entity linking supervision is available, the model is learned with an additional knowledge-aware log-likelihood or max-margin objective. **WKLM** (Xiong et al., 2020) also use documents from Wikipedia aligned with fact triples from WikiData. It replaces entity mentions in the original document with names of other entities of the same type. The model is trained to distinguish the correct entity mention from randomly chosen ones. **BERT-MK** (He et al., 2019) integrates
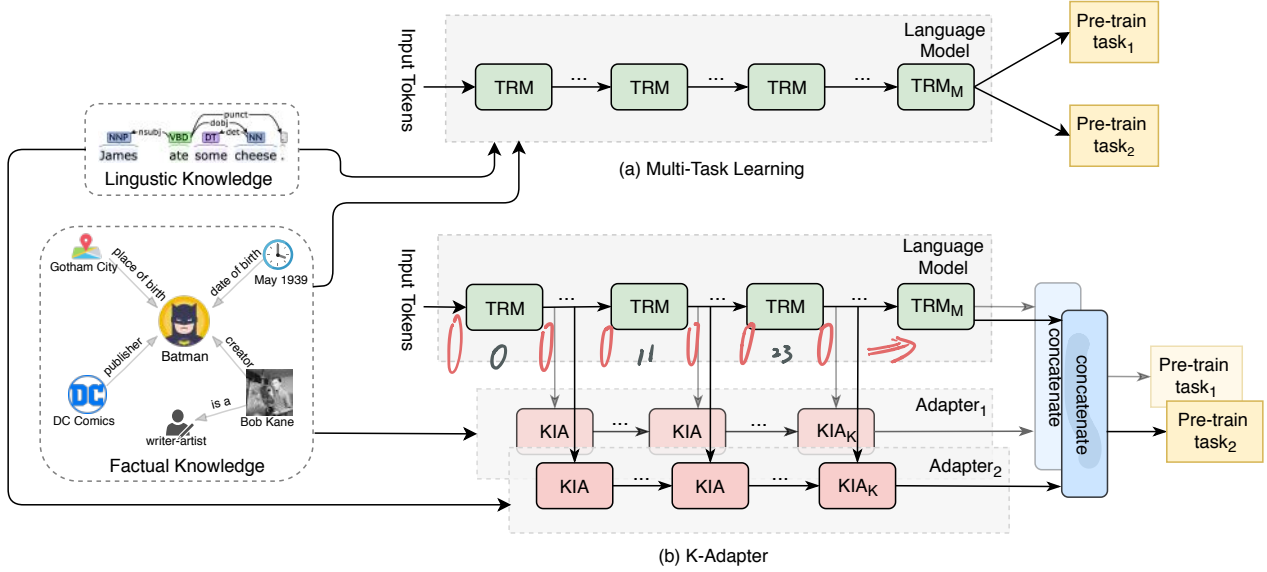
*Figure 1.* (a) Pre-trained language models inject multiple kinds of knowledge with multi-task learning. Model parameters need to be retrained when injecting new kinds of knowledge, which may result in the catastrophic forgetting (b) Our K-ADAPTER injects multiple kinds of knowledge by training adapters independently on different pre-train tasks, which supports continual knowledge infusion. When we inject new kinds of knowledge, the existing knowledge-specific adapters will not be affected. KIA represents the adapter layer and TRM represents the transformer layer, both of which are shown in Figure 2.
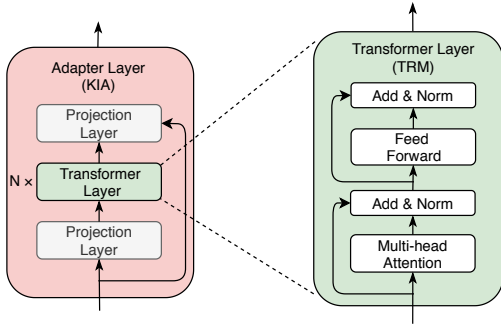


*Figure 2.* Structure of the adapter layer (left). The adapter layer consists of two projection layers and $N=2$ transformer layers, and a skip-connection between two projection layers.

fact triples from knowledge graph. For each entity, it sample incoming and outcoming instances from the neighbors on the knowledge graph, and replaces head or tail entity to create negative instances. The model is learned to discriminate between real and fake facts.

As shown in Table 1, our model (K-ADAPTER) differs from previous studies in three aspects. First, we consider both fact-related objective (i.e. predicate/relation prediction) and linguistic-related objective (i.e. dependency relation prediction). Second, the original parameter of BERT is clamped in the knowledge infusion process. Third, our approach supports continual learning, which means that the learning

of different adapters are not entangled. This flexibility enables us to efficiently inject different types of knowledge independently, and inject more types of knowledge without any loss on the previously injected knowledge.

## 3. K-ADAPTER

As illustrated in Figure 1 (a), most of the previous works enhance pre-trained language models by injecting knowledge and update model parameters through multi-task learning. Regardless of these different versions of knowledge-injected methods with multi-task learning, common issues not fully studied are catastrophic forgetting of previous knowledge. To stress this, we present K-ADAPTER as shown in Figure 1(b), where multiple kinds of knowledge are injected into different compact neural models (i.e., adapters in this paper) individually instead of directly injecting knowledge into pre-trained models. It remains the original representation of a pre-trained model fixed and supports continual knowledge infusion, i.e., injecting each kind of knowledge into the corresponding knowledge-specific adapter and producing disentangled representation. Specifically, adapters are knowledge-specific models (with few parameters) plugged outside of a pre-trained model, which inputs are the output hidden-states of intermediate layers of the pre-trained model. We pre-train each adapter on different pre-train tasks independently for injecting different knowledge while the original parameters of the pre-trained model are frozen. In this paper, we exploit RoBERTa (Liu et al., 2019) as the

pre-trained model, and mainly infuse factual knowledge and linguistic knowledge with two kinds of adapters, i.e., factual adapter and linguistic adapter which are pre-trained on the relation classification task and dependency relation prediction task, respectively. We will first describe the structure of our adapter, and then present the process of pre-training knowledge-specific adapters in the following sections.

## 3.1. Adapter Structure

There are many ways to implement adapters (Rebuffi et al., 2017; Houlsby et al., 2019). In this paper, we present a different adapter structure as shown in Figure 2 as the knowledge-specific adapter. Each adapter model consists of $K$ adapter layers that contain $N$ transformer (Vaswani et al., 2017) layers and two projection layers. A skip-connection is applied across two projection layers. Specifically, for each adapter model, we plug adapter layers among different transformer layers of the pre-trained model. We concatenate the output hidden feature of the transformer layer of the pre-trained model and the output feature of the former adapter layer as the input feature of the current adapter layer. For each knowledge-specific adapter, we use the concatenation of the last hidden feature of the pre-trained model and the last hidden feature of the adapter as the final output feature of this adapter model.

In the pre-training procedure, we train each knowledge-specific adapter on different pre-training tasks individually. For various downstream tasks, K-ADAPTER can adopt the fine-tuning procedure similar to RoBERTa and BERT. When only one knowledge-specific is adopted, we can take the final output feature of this adapter model as the input for task-specific layers of the downstream task. When multiple knowledge-specific adapters are adopted, we concatenate the output features of different adapter models as the input for task-specific layers of the downstream task.

## 3.2. Pre-training settings

To be more specific, we use the RoBERTa$_{LARGE}$ (L=24, H=1024, A=16, 355M params) implementation by Hugging-face[1] as the pre-trained model in all our experiments. As for each adapter layer, we denote the number of transformer layer as $N$, the hidden dimension of transformer layer as $H_A$, the number of self-attention heads of transformer layer as $A_A$, the hidden dimension of down-projection layer as $H_d$ and the hidden dimension of up-projection layer as $H_u$. In detail, we have the following adapter size: $N = 2$, $H_A = 768$, $A_A = 12$, $H_u = 1024$ and $H_d = 768$. The RoBERTa layers where adapter layers plug in are {0,11,23}, and different adapter layers do not share parameters. Thus the total parameters for each adapter model are about 42M,

which are much smaller than RoBERTa$_{LARGE}$ and make the training process memory efficient. Then we will describe how to inject different knowledge into knowledge-specific adapters below.

## 3.3. Factual Adapter

Factual knowledge can be described as the basic information that concerned with facts or contains facts. In this work, we acquire factual knowledge from the relationships among entities in natural language language. We extract a sub-dataset T-REx-rc from T-REx (ElSahar et al., 2018) which is a large scale alignment dataset between Wikipedia abstracts and Wikidata triples, having 685 unique relations. To be specific, T-REx-rc only contains the sentences where a surface form of the relation appears, and then we discard all relations having lees than 50 entity pairs, collecting 430 relations and 5.5M sentences. In order to inject factual knowledge, we propose pre-training a knowledge-specific adapter called facAdapter on the T-REx-rc dataset using relation classification task. This task requires a model to classify relation labels of given entity pairs based on context. Specifically, we use the concatenation of the last hidden feature of RoBERTa and the last hidden feature of facAdapter as the input representation, and then apply the pooling layer to input representations of the given entities, and then concatenate two entity representations to perform relation classification. RoBERTa is fixed during training and the parameters of the facAdapter are trainable and initialized randomly. More training details of factual adapter can be found in the supplementary material.

## 3.4. Linguistic Adapter

Linguistic knowledge is implicitly contained in natural language texts, e.g., syntax and semantics information. In this work, we acquire linguistic knowledge from dependency relationships among words in natural language text. We build a dataset for training the linguistic adapter. We run the off-the-shell dependency parser from Standford Parser[2] (Chen & Manning, 2014) on a part of Book Corpus (Zhu et al., 2015) consisting of 1M examples. To inject linguistic knowledge, we pre-train another knowledge-specific adapter called linAdapter on dependency relation prediction. This task aims to predict the father index of each token in the given sentence. Similar to training the facAdapter, we use the concatenation of the last hidden feature of RoBERTa and the last hidden feature of linAdapter as the input representation, and then apply a linear layer to input representations of each tokens to perform classification. RoBERTa is fixed during training and the parameters of the linAdapter are trainable and initialized randomly. We describe the training details of linguistic adapter in the supplementary material.

---

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

[2] http://nlp.stanford.edu/software/lex-parser.html

*Table 2.* Results on two entity typing datasets OpenEntity and FIGER.

| Model | OpenEntity | | | FIGER | | |
|---|---|---|---|---|---|---|
| | **P** | **R** | **F1** | **Acc** | **Ma-F1** | **Mi-F1** |
| NFGEC (Shimaoka et al., 2016) | 68.80 | 53.30 | 60.10 | 55.60 | 75.15 | 71.73 |
| BERT-base (Zhang et al., 2019) | 76.37 | 70.96 | 73.56 | 52.04 | 75.16 | 71.63 |
| ERNIE (Zhang et al., 2019) | 78.42 | 72.90 | 75.56 | 57.19 | 75.61 | 73.39 |
| KnowBERT (Peters et al., 2019) | 78.60 | 73.70 | 76.10 | - | - | - |
| KEPLER (Wang et al., 2019) | 77.20 | 74.20 | 75.70 | - | - | - |
| WKLM (Xiong et al., 2020) | - | - | - | 60.21 | 81.99 | 77.00 |
| RoBERTa | 77.55 | 74.95 | 76.23 | 56.31 | 82.43 | 77.83 |
| RoBERTa + multitask | 77.96 | **76.00** | 76.97 | 59.86 | 84.45 | 78.84 |
| K-ADAPTER (F+L) | 79.25 | 75.00 | 77.06 | **61.81** | **84.87** | **80.54** |
| K-ADAPTER (F) | 79.30 | 75.84 | **77.53** | 59.50 | 84.52 | 80.42 |
| K-ADAPTER (L) | **80.01** | 74.00 | 76.89 | 61.10 | 83.61 | 79.18 |

# 4. Experiments

We evaluate our K-ADAPTER on three downstream tasks, i.e., entity typing, question answering and relation classification. Furthermore, we conduct probing experiments to examine the ability of models for learning factual knowledge. The notations of K-ADAPTER (F+L), K-ADAPTER (F), and K-ADAPTER (L) denote our model which consists of both factual adapter and linguistic adapter, only factual adapter and only linguistic adapter, respectively. The implementation details are in the supplementary material.

## 4.1. Entity Typing

We conduct experiments on fine-grained entity typing which aims to predict the types of a given entity and its context. For this task, we evaluate our models on Open Entity (Choi et al., 2018) and FIGER (Ling et al., 2015) following the same split setting as Zhang et al. (2019). The statistics of datasets are shown in the supplementary material. To fine-tune our models for entity typing, we modify the input token sequence by adding the special token "@" before and after a certain entity, then the first "@" special token representation is adopted to perform classification. To compare performance on the Open Entity dataset with previous works (Shimaoka et al., 2016; Zhang et al., 2019; Peters et al., 2019; Wang et al., 2019), we evaluate the models using loose micro precision, recall and F1, and adopt micro F1 score as the final metric to represent the model performance. As for FIGER dataset, we adopt strict accuracy, loose macro, loose micro F1 scores for evaluation following the same evaluation criteria used in the previous works.

**Baselines** NFGEC (**Shimaoka et al., 2016**) employs recursive neural networks to compose context representations and adapts an attention mechanism to focus on relevant expressions. **KEPLER (Wang et al., 2019)** integrates factual knowledge into pre-trained models with the supervision

of the knowledge embedding objective. They propose to encode textual descriptions of entities as their entity embeddings, and then jointly learn the knowledge embeddings and language representations. **RoBERTa+multitask** is our RoBERTa model pre-trained with multi-task learning (as shown in Figure 1(a)) for injecting multiple kinds of knowledge on two pre-training tasks, i.e., relation classification and dependency relation prediction. Other baseline models, such as BERT-base (Zhang et al., 2019), ERNIE (Zhang et al., 2019), KnowBERT (Peters et al., 2019) and WKLM (Xiong et al., 2020) are described in Section 2.

**Results and Discussion** The results on Open Entity and FIGER are shown in Table 2. Our K-ADAPTER (F+L) achieves consistent improvements across these two datasets. As for the Open Entity dataset, our RoBERTa has achieved better results than other baseline models. K-ADAPTER (F+L) achieves improvement of 0.83% F1 and 1.7% precision over RoBERTa, which means the factual knowledge and linguistic knowledge help to predict the types more accurately. As for the FIGER dataset, FIGER covers more entity types and thus more fine-grained than Open Entity. Compared with WKLM, our K-ADAPTER (F+L) improves the macro F1 by 2.88%, micro F1 by 2.54% and strict accuracy by 1.60%. This demonstrates that K-ADAPTER (F+L) benefits fine-grained entity typing.

## 4.2. Question Answering

We conduct experiments on two question answering tasks, i.e., commonsense question answering and open-domain question answering. Commonsense question answering aims to answer questions with commonsense. We adopt CosmosQA (Huang et al., 2019) to evaluate our models. CosmosQA requires commonsense-based reading comprehension, formulated as multiple-choice questions. To fine-tune our models for CosmosQA, for each answer, the input token sequence is modified as "<SEP>context

*Table 3.* Results on three question answering datasets including: CosmosQA, SearchQA and Quasar-T.

| Model | SearchQA | | Quasar-T | | CosmosQA |
|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | Accuracy |
| BiDAF (Seo et al., 2016) | 28.60 | 34.60 | 25.90 | 28.50 | - |
| AQA (Buck et al., 2018) | 40.50 | 47.40 | - | - | - |
| R^3 (Wang et al., 2017a) | 49.00 | 55.30 | 35.30 | 41.70 | - |
| DSQA (Lin et al., 2018) | 49.00 | 55.30 | 42.30 | 49.30 | - |
| Evidence Agg. (Wang et al., 2018) | 57.00 | 63.20 | 42.30 | 49.60 | - |
| BERT (Xiong et al., 2020) | 57.10 | 61.90 | 40.40 | 46.10 | - |
| WKLM (Xiong et al., 2020) | 58.70 | 63.30 | 43.70 | 49.90 | - |
| WKLM + Ranking (Xiong et al., 2020) | 61.70 | 66.70 | 45.80 | 52.20 | - |
| BERT-FT$_{RACE+SWAG}$ (Huang et al., 2019) | - | - | - | - | 68.70 |
| RoBERTa | 59.01 | 65.62 | 40.83 | 48.84 | 80.59 |
| RoBERTa + multitask | 59.92 | 66.67 | 44.62 | 51.17 | 81.19 |
| K-ADAPTER (F+L) | **61.96** | **67.31** | 45.69 | 52.48 | **81.83** |
| K-ADAPTER (F) | 61.85 | 67.17 | **46.75** | **53.27** | 80.93 |
| K-ADAPTER (L) | 61.15 | 66.82 | 45.66 | 52.39 | 80.76 |

</SEP>question</SEP>answer</SEP>", then the representation of the first token is adopted to perform classification, and will get a score for this answer. After getting four scores, the answer with the highest score will be selected. We report accuracy scores obtained from the leaderboard. Open-domain question answering aims to answer open-domain questions using external resources such as collections of documents and webpages. We evaluate our modes on two public open-domain QA datasets, i.e., Quasar-T (Dhingra et al., 2017) and SearchQA (Dunn et al., 2017). The statistics of these datasets are shown in the supplementary material. Specifically, we first retrieve paragraphs corresponding to the question from external resources using the information retrieval system and then extract the answer from these retrieved paragraphs through the reading comprehension technique. Following previous work(Lin et al., 2018), we use the retrieved paragraphs provided by Wang et al. (2017b) for these two datasets. To fine-tune our models for this task, the input token sequence is modified as "<SEP>question </SEP>paragraph</SEP>". We apply linear layers over the last hidden features of our model to predict the start and end position of the answer span. We adopt two metrics including ExactMatch (EM) and loose F1 scores to evaluate our models.

**Baselines** **BERT-FT**$_{RACE+SWAG}$ (Huang et al., 2019) is the BERT model sequentially fine-tuned on both RACE and SWAG datasets for knowledge transfer. **BiDAF (Seo et al., 2016)** is a reading comprehension model with a bi-directional attention network. **AQA (Buck et al., 2018)** is a reinforced system learning to re-write questions and aggregate the answers generated by the re-written questions. **R^3 (Wang et al., 2017a)** is a reinforced model making use of a ranker for selecting most confident paragraph to train the reading comprehension model. **Evidence Agg. (Wang**

et al., 2018) proposes making use of the aggregated evidence from across multiple paragraphs to better determine the answer with re-rankers. **BERT (Xiong et al., 2020)** is the BERT re-implementation by Xiong et al. (2020) for open-domain QA. **WKLM (Xiong et al., 2020)** is described in Section 2, which is adopted as the reader model to read multiple paragraphs to predict a single answer. **WKLM + Ranking (Xiong et al., 2020)** is a WKLM paragraph reader plus with a BERT based paragraph ranker with distant-supervised data to assign each paragraph a relevance score.

**Results and Discussion** The results on CosmosQA are shown in Table 3. Compared with BERT-FT$_{RACE+SWAG}$, our RoBERTa significantly achieves 11.89% improvement of accuracy. CosmosQA combines reading comprehension with commonsense reasoning, requires contextual commonsense reasoning over considerably more complex, diverse, and longer context. Compared to RoBERTa, K-ADAPTER (F+L) further improves the accuracy by 1.24%, which indicates that K-ADAPTER can obtain better commonsense inference ability. Moreover, the performance of ablated K-ADAPTER models, i.e., K-ADAPTER (F) and K-ADAPTER (L) are clearly better than RoBERTa, but slightly lose compared with RoBERTa+multitask. It is notable that K-ADAPTER (F+L) makes obvious improvement comparing with RoBERTa+multitask. This demonstrates that the combination of multiple knowledge-specific adapters could achieve better performance.

The results for open-domain QA are shown in Table 3. Our K-ADAPTER models achieve better results on these two datasets as compared to other baselines. This indicates that our K-ADAPTER models can make full use of the infused knowledge and accordingly benefit understanding the retrieved paragraphs to answer the question. Specifically, on

SearchQA, our K-ADAPTER (F+L) makes significant improvement of 4.01% F1 scores, comparing with WKLM where the ranking scores are not used, and even has a slight improvement as compared to WKLM+Ranking. It is worth noting that K-ADAPTER models do not consider the confidence of each retrieved paragraph, while WKLM+Ranking utilizes ranking scores from a BERT based ranker. On the Quasar-T dataset, our K-ADAPTER (F+L) also outperforms WKLM by 2.58% F1 score and slightly outperforms WKLM+Ranking.

## 4.3. Relation Classification

Relation classification aims to determine the correct relation between two entities in a given sentence. We fine-tune and compare our models with several baseline methods on a large-scale relation classification dataset TACRED (Zhang et al., 2017), which covers 42 relation types and contains 106,264 sentences. The statistics of this dataset are shown in the supplementary material. To fine-tune our models for relation classification, we modify the input token sequence by adding special token "@" before and after the first entity, adding "#" before and after the second entity. Then the token representations of the first special token "@" and "#" are concatenated to perform relation classification. We evaluate the models using micro precision, recall and F1, and adopt micro F1 score as the metric to represent the model performance as previous works.

**Baselines** C-GCN (Zhang et al., 2018) employs graph convolutional networks (GCNs) over dependency tree structures to model dependency trees for relation classification. **BERT-large (Baldini Soares et al., 2019)** is a baseline BERT-large model of Baldini Soares et al. (2019) to perform task-specific fine-tuning on TACRED. **BERT+MTB (Baldini Soares et al., 2019)** is a method of training relation representation without supervision from a knowledge base or human annotators by matching the blanks (MTB). Other baseline models, such as BERT-base (Zhang et al., 2019), ERNIE (Zhang et al., 2019), KnowBERT (Peters et al., 2019), KEPLER (Wang et al., 2019) and RoBERTa+multitask are described in Section 2 and 4.1.

**Results and Discussion** Table 4 shows the performances of different models on TACRED. The results indicate that K-ADAPTER models significantly outperform all baselines, which directly demonstrate our models can benefit relation classification. In particular, (1) K-ADAPTER models outperform our RoBERTa, which proves the effectiveness of infusing knowledge into pre-trained model with adapters. (2) K-ADAPTER models gain more improvement compared with *RoBERTa+multitask* which learns tangled knowledge. This directly demonstrates injecting knowledge individually in K-ADAPTER way would help models make full use of

*Table 4.* Results on the TACRED relation classification dataset.

| Model | P | R | F1 |
|---|---|---|---|
| C-GCN (Zhang et al., 2018) | 69.90 | 63.30 | 66.40 |
| BERT-base (Zhang et al., 2019) | 67.23 | 64.81 | 66.00 |
| ERNIE (Zhang et al., 2019) | 69.97 | 66.08 | 67.97 |
| BERT-large (Baldini Soares et al., 2019) | - | - | 70.10 |
| BERT+MTB (Baldini Soares et al., 2019) | - | - | 71.50 |
| KnowBERT (Peters et al., 2019) | **71.60** | 71.40 | 71.50 |
| KEPLER (Wang et al., 2019) | 70.43 | 73.02 | 71.70 |
| RoBERTa | 70.17 | 72.36 | 71.25 |
| RoBERTa + multitask | 70.18 | 73.11 | 71.62 |
| K-ADAPTER (F+L) | 70.05 | 73.92 | 71.93 |
| K-ADAPTER (F) | 69.39 | 74.59 | 71.89 |
| K-ADAPTER (L) | 68.85 | **75.37** | **71.96** |

knowledge. (3) K-ADAPTER (L) achieves the best performance among all K-ADAPTER models. This demonstrates linguistic knowledge is more useful on TACRED dataset.

## 4.4. Probing Experiments

Although our K-ADAPTER models have shown superior performance on several knowledge-driven downstream tasks, it does not directly provide insights into whether our models infuse richer factual and commonsense knowledge. Thus we utilize a LAMA (LAnguage Model Analysis) probe (Petroni et al., 2019) to examine the ability to memorize factual knowledge. Specifically, the LAMA probing task aims to answer cloze-style questions about relational facts, e.g., "Simon Bowman was born in [MASK]". This task requires the language model to predict a distribution over a limited vocabulary to replace [MASK]. We report mean precision at one (P@1) macro-averaged over relations.

**Settings** We consider several language models including: ELMo (Peters et al., 2018), ELMo5.5B (Peters et al., 2018), Transformer-XL (Dai et al., 2019), BERT$_{LARGE}$ and RoBERTa$_{LARGE}$. We focus on LAMA-GoogleRE and LAMA-T-REx datasets, which are aimed at factual knowledge. We also conduct probe experiments on LAMA-UHN (Poerner et al., 2019), a more factual subset of LAMA-Google-RE and LAMA-T-REx, by filtering out queries that are easy to answer from entity names alone. Different models have different vocabulary sizes. To conduct a more fair comparison experiment, we adopt the intersection of vocabularies and let every language model rank only tokens in this vocabulary following Petroni et al. (2019). For simplicity, we only compare K-APDATER (F) which is infused with factual knowledge, with other baseline models.

**Results and Discussion** Results on LAMA and LAMA-UHN datasets are shown in Table 5. It is surprising that BERT$_{LARGE}$ performs better than RoBERTa$_{LARGE}$.

*Table 5.* P@1 on LAMA and LAMA-UHN across Google-RE and T-REx corpora.

| Corpus | Models | | | | | |
|---|---|---|---|---|---|---|
| | ELMo | ELMo5.5B | TransformerXL | BERT-large | RoBERTa$_{LARGE}$ | K-APDATER |
| LAMA-Google-RE | 2.2 | 3.1 | 1.8 | 12.1 | 4.8 | 7.0 |
| LAMA-UHN-Google-RE | 2.3 | 2.7 | 1.3 | 6.5 | 2.5 | 3.7 |
| LAMA-T-REx | 0.2 | 0.3 | 19.5 | 33.9 | 27.1 | 29.1 |
| LAMA-UHN-T-REx | 0.2 | 0.2 | 12.6 | 26.2 | 20.1 | 23.0 |

*Table 6.* Examples of generation for RoBERTa$_{LARGE}$ and K-ADAPTER. The last column reports the top ranked predicted tokens. Correct predictions are in bold.

| Query | Answer | Model | Generation |
|---|---|---|---|
| The official language of Mauritius is [MASK]. | English | RoBERTa | French, **English**, Dutch, Arabic, Portuguese, Spanish |
| | | K-ADAPTER | **English**, French, Dutch, Arabic, Portuguese, Spanish |
| The native language of Mammootty is [MASK]. | Malayalam | RoBERTa | English, Tamil, Hindi, Sanskrit, Arabic, Chinese, spoken |
| | | K-ADAPTER | **Malayalam**, Tamil, Hindi, Mandarin, English, Thai |
| Birds have [MASK]. | feathers | RoBERTa | flown, **feathers**, babies, noticed, gone, changed, come |
| | | K-ADAPTER | feelings, **feathers**, brains, names, souls, eyes, wings |
| Ravens can [MASK]. | fly | RoBERTa | win, play, score, lose, run, drink, **fly**, roll, wait, ide |
| | | K-ADAPTER | **fly**, swim, sing, shoot, kill, go, fish, drink, die, roll |
| Sometimes virus causes [MASK]. | infection | RoBERTa | cancer, death, illness, blindness, paralysis, **infection** |
| | | K-ADAPTER | cancer, illness, death, **infection**, disease, problems, pain |
| Sunshine Coast, British Columbia is located in [MASK]. | Canada | RoBERTa | Florida, California, Texas, Hawaii, Mexico, Arizona |
| | | K-ADAPTER | **Canada**, Vancouver, Victoria, BC, Australia, California |
| iPod Touch is produced by [MASK]. | Apple | RoBERTa | **Apple**, Samsung, Qualcomm, LG, Microsoft, HTC |
| | | K-ADAPTER | **Apple**, HTC, Samsung, Motorola, Intel, Sony |

There is one possible reason: BERT uses a character-level BPE (Gage, 1994) vocabulary, while RoBERTa considers byte-level BPE vocabulary. This finding indicates that, although using bytes makes it possible to learn a subword vocabulary that can encode any text without introducing "unknown" tokens, it might indirectly harm the model's ability to learn factual knowledge, e.g., some proper nouns may be divided into bytes. Thus in the following experiments, we do not take BERT into account.

K-ADAPTER outperforms other models (except for BERT) by a huge margin. On LAMA datasets, compared to RoBERTa$_{LARGE}$, K-ADAPTER obtains 2.2% and 1.2% P@1 improvement across Google-RE and T-REx, respectively. Moreover, compared to RoBERTa$_{LARGE}$, K-ADAPTER still achieves better results on LAMA-UHN. The results demonstrate that K-ADAPTER captures richer factual and commonsense knowledge than RoBERTa. Furthermore, Table 6 shows several examples for the generation of RoBERTa$_{LARGE}$ and K-ADAPTER for LAMA queries. From these examples, we can find that the objects predicted by K-ADAPTER are more accurate.

## 5. Conclusion

In this paper, we propose a flexible and simple approach, called K-ADAPTER, to infuse knowledge into large pre-trained models. K-ADAPTER remains the original parameters of pre-trained models unchanged and supports continual knowledge infusion, i.e., new kinds of injected-knowledge will not affect the parameters learned for old knowledge. Specifically, factual knowledge and linguistic knowledge are infused into RoBERTa with two kinds of adapters, which are pre-trained on the relation classification task and dependency relation prediction task, respectively. Extensive experiments on three knowledge-driven downstream tasks demonstrate that the performance of each adapter achieves a significant improvement individually, and even more together. Probing experiments further suggest that K-ADAPTER captures richer factual and commonsense knowledge than RoBERTa.

# References

Baldini Soares, L., FitzGerald, N., Ling, J., and Kwiatkowski, T. Matching the Blanks: Distributional Similarity for Relation Learning. In *ACL*, pp. 2895–2905, 2019.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. Translating embeddings for modeling multi-relational data. In *NIPS*, pp. 2787–2795, 2013.

Buck, C., Bulian, J., Ciaramita, M., Gesmundo, A., Houlsby, N., Gajewski, W., and Wang, W. Ask the Right Questions: Active Question Reformulation with Reinforcement Learning. In *ICLR*, 2018.

Chen, D. and Manning, C. D. A fast and accurate dependency parser using neural networks. In *EMNLP*, pp. 740–750, 2014.

Choi, E., Levy, O., Choi, Y., and Zettlemoyer, L. Ultra-fine entity typing. In *ACL*, pp. 87–96, 2018.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., and Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL*, pp. 2978–2988, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pp. 4171–4186, 2019.

Dhingra, B., Mazaitis, K., and Cohen, W. W. Quasar: Datasets for question answering by search and reading. In *arXiv preprint arXiv:1707.03904*, 2017.

Dunn, M., Sagun, L., Higgins, M., Güney, V. U., Cirik, V., and Cho, K. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. In *ArXiv preprint arXiv:1704.05179*, 2017.

ElSahar, H., Vougiouklis, P., Remaci, A., Gravier, C., Hare, J. S., Laforest, F., and Simperl, E. T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples. In *LREC*, 2018.

Gage, P. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38, 1994.

He, B., Zhou, D., Xiao, J., Liu, Q., Yuan, N. J., Xu, T., et al. Integrating Graph Contextualized Knowledge into Pre-trained Language Models. In *arXiv preprint arXiv:1912.00147*, 2019.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-Efficient Transfer Learning for NLP. In *ICML*, pp. 2790–2799, 2019.

Huang, L., Bras, R. L., Bhagavatula, C., and Choi, Y. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *EMNLP*, pp. 2391–2401, 2019.

Kassner, N. and Schütze, H. Negated LAMA: Birds cannot fly. *arXiv preprint arXiv:1911.03343*, 2019.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., and et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):35213526, Mar 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114.

Lauscher, A., Vulić, I., Ponti, E. M., Korhonen, A., and Glavaš, G. Informing unsupervised pretraining with external linguistic knowledge. *arXiv preprint arXiv:1909.02339*, 2019.

Levine, Y., Lenz, B., Dagan, O., Padnos, D., Sharir, O., Shalev-Shwartz, S., Shashua, A., and Shoham, Y. Sensebert: Driving some sense into bert. *arXiv preprint arXiv:1908.05646*, 2019.

Lin, Y., Ji, H., Liu, Z., and Sun, M. Denoising distantly supervised open-domain question answering. In *ACL*, pp. 1736–1745, 2018.

Ling, X., Singh, S., and Weld, D. S. Design challenges for entity linking. *TACL*, 3:315–328, 2015.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *NAACL*, pp. 2227–2237, 2018.

Peters, M. E., Neumann, M., Logan, I., Robert, L., Schwartz, R., Joshi, V., Singh, S., and Smith, N. A. Knowledge enhanced contextual word representations. In *EMNLP*, pp. 43–54, 2019.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., and Riedel, S. Language Models as Knowledge Bases? In *EMNLP*, pp. 2463–2473, 2019.

Poerner, N., Waltinger, U., and Schütze, H. BERT is Not a Knowledge Base (Yet): Factual Knowledge vs. Name-Based Reasoning in Unsupervised QA. *arXiv preprint arXiv:1911.03681*, 2019.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. *OpenAI Blog*, 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Rebuffi, S.-A., Bilen, H., and Vedaldi, A. Learning multiple visual domains with residual adapters. In *NeurIPS*, pp. 506–516, 2017.

Seo, M., Kembhavi, A., Farhadi, A., and Hajishirzi, H. Bidirectional Attention Flow for Machine Comprehension. *ArXiv*, abs/1611.01603, 2016.

Shimaoka, S., Stenetorp, P., Inui, K., and Riedel, S. An attentive neural architecture for fine-grained entity type classification. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction(AKBC)*, pp. 69–74, 2016.

Talmor, A., Elazar, Y., Goldberg, Y., and Berant, J. oLMpics–On what Language Model Pre-training Captures. *arXiv preprint arXiv:1912.13283*, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, pp. 5998–6008, 2017.

Wang, S., Yu, M., Guo, X., Wang, Z., Klinger, T., Zhang, W., Chang, S., Tesauro, G., Zhou, B., and Jiang, J. Reinforced reader-ranker for open-domain question answering. *arXiv preprint arXiv:1709.00023*, 2017a.

Wang, S., Yu, M., Jiang, J., Zhang, W., Guo, X., Chang, S., Wang, Z., Klinger, T., Tesauro, G., and Campbell, M. Evidence aggregation for answer re-ranking in open-domain question answering. In *ICLR*, 2018.

Wang, W., Yang, N., Wei, F., Chang, B., and Zhou, M. Gated self-matching networks for reading comprehension and question answering. In *ACL*, pp. 189–198, 2017b.

Wang, X., Gao, T., Zhu, Z., Liu, Z., Li, J., and Tang, J. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *arXiv preprint arXiv:1911.06136*, 2019.

Xiong, W., Du, J., Wang, W. Y., and Stoyanov, V. Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model. In *ICLR*, 2020.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237*, 2019.

Zhang, Y., Zhong, V., Chen, D., Angeli, G., and Manning, C. D. Position-aware Attention and Supervised Data Improve Slot Filling. In *EMNLP*, pp. 35–45, 2017.

Zhang, Y., Qi, P., and Manning, C. D. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *EMNLP*, pp. 2205–2215, 2018.

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., and Liu, Q. ERNIE: Enhanced Language Representation with Informative Entities. In *ACL*, pp. 1441–1451, 2019.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *ICCV*, pp. 19–27, 2015.

# Supplementary Material

## A. Pre-Training Details

### A.1. Factual Adapter

The pre-trained model is fixed during training and the parameters of the factual adapter are trainable and initialized randomly. The model is trained with cross-entropy loss. To accelerate the training process, we set the max sequence length as 64 as the average sequence length of T-REx-rc is only 22.8. We train the model for 5 epochs using a batch size of 128. We use AdamW to optimize our models with the initial learning rate of 2e-5. We train the model with 4 16G NVIDIA V100 GPUs.

### A.2. Linguistic Adapter

Same as the training process of the factual adapter, the pre-trained model is fixed during training and the parameters of the linguistic adapter are trainable and initialized randomly. The model is trained with BCEWithLogits loss. We set the max sequence length as 128. We train the model for 10 epochs using a batch size of 256. We use AdamW with the initial learning rate of 1e-5. We train the model with 4 16G NVIDIA V100 GPUs.

## B. Dataset statistics

In Table 7, we present the statistics of one relation classification dataset TACRED, and two entity typing datasets OpenEntity and FIGER. In Table 8, we present the statistics of one commonsense QA dataset CosmosQA and two open-domain QA datasets SearchQA and Quasar-T.

*Table 7.* The statistics of the relation classification dataset TACRED and entity typing datasets, i.e., Open Entity and FIGER.

| Dataset | Train | Dev | Test | Relation/Type |
|---|---|---|---|---|
| TACRED | 68,124 | 22,631 | 15,509 | 42 |
| Open Entity | 2,000 | 2,000 | 2,000 | 6 |
| FIGER | 2,000,000 | 10,000 | 563 | 113 |

*Table 8.* The statistics of the question answering datasets, i.e., CosmosQA, SearchQA and Quasar-T.

| Dataset | Train | Dev | Test |
|---|---|---|---|
| CosmosQA | 25,588 | 3,000 | 7,000 |
| SearchQA | 99,811 | 13,893 | 27,247 |
| Quasar-T | 28,496 | 3,000 | 3,000 |

## C. Fine-tuning Details and Hyperparameters

We implement our experiments using Huggingface[3]. For all fine-tuning experiments, we use AdamW as the optimizer. The parameters of adapters are fixed during the fine-tuning process and the parameters of RoBERTa are trainable and initialized from Huggingface checkpoint. We select the best hyperparameters on the validation set. For all experiments, we set the random seed to be 42 for reproductibility.

### C.1. Entity typing

For Open Entity dataset, we set the max sequence length to be 256 and select the hyperparameters from batch size: {4, 8}, learning rate: {2e-5, 1e-5, 5e-6} and warmup step: {0, 200, 500, 1000, 1200}. For K-ADAPTER (F), the best performance is achieved at batch size=4, lr=5e-6, warmup=500 (it takes about 2 hours to get the best result running on singe 16G P100). For K-ADAPTER (L), the best performance is achieved at batch size=4, lr=5e-6, warmup=1000 (it takes about 2 hours to get the best result running on singe 16G P100). For K-ADAPTER (F+L), the best performance is achieved at batch size=4, lr=5e-6, warmup=1000 (it takes about 3 hours to get the best result running on singe 16G P100). For FIGER dataset, we run experiments on 4 16G P100 for 3 epochs, set the max sequence length to be 256, and select the hyperparameters from batch size: {64, 512, 2048}, learning rate: {2e-5, 1e-5, 5e-6} and warmup step: {0, 200, 500, 1000, 1200}. For K-ADAPTER (F), the best performance is achieved at batch size=2048, lr=5e-6, warmup=500. For K-ADAPTER (L), the best performance is achieved at batch size=2048, lr=5e-6, warmup=200. For K-ADAPTER (F+L), the best performance is achieved at batch size=2048, lr=5e-6, warmup=1000.

### C.2. Question Answering

For CosmosQA dataset, we run experiments on one single 16G P100 for 3 epochs, set the max sequence length to be 256, and select the hyperparameters from batch size: {16, 32, 64, 128}, learning rate: {2e-5, 1e-5, 5e-6} and warmup step: {0, 200, 500, 800, 1000}. For K-ADAPTER (F+L) and its ablated models, the best performance is achieved at batch size=64, lr=1e-5, warmup=0 (it takes about 8 hours to get the best result).

For SearchQA dataset, we run experiments on one single 16G P100 for 2 epochs, set the max sequence length to be 128, and select the hyperparameters from batch size: {2, 4, 8, 16}, learning rate: {5e-5, 2e-5, 1e-5, 5e-6} and warmup step: {0, 500, 1000}. For K-ADAPTER (F+L) and its ablated models, the best performance is achieved at batch size=8, lr=5e-6, warmup=0. For Quasar-T dataset, we run experiments on one single 16G P100 for 5 epochs, set the

---

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

max sequence length to be 256, and select the hyperparameters from batch size: {2, 4, 8, 16}, learning rate: {5e-5, 2e-5, 1e-5, 5e-6} and warmup step: {0, 500, 1000}. For K-ADAPTER (F+L) and its ablated models, the best performance is achieved at batch size=16, lr=1e-5, warmup=0.

### C.3. Relation Classification

For TACRED dataset, we run experiments on 4 16G P100 for 5 epochs, set the max sequence length to be 184, and select the hyperparameters from batch size: {4, 8, 16, 32}, learning rate: {2e-5, 1e-5, 5e-6, 1e-6} and warmup step: {0, 200, 500, 800, 1000, 1200}. For K-ADAPTER (F), the best performance is achieved at batch size=32, lr=1e-5, warmup=500. For K-ADAPTER (L), the best performance is achieved at batch size=32, lr=1e-5, warmup=200. For K-ADAPTER (F+L), the best performance is achieved at batch size=32, lr=5e-6, warmup=1000.

## D. Probing Experiments

We implement our probing experiments using LAMA[4]. LAMA probe aims to answer cloze-style questions about relational facts, e.g., "Simon Bowman was born in [MASK]". This task requires the language model to predict a distribution over a limited vocabulary to replace [MASK]. When we infuse knowledge into knowledge-specific adapters, we do not change the original parameters of the pre-trained model and thus do not adopt the masked language model (MLM) as a pre-training task. Therefore, before we conduct probing experiments, we need to add and train a linear layer as the mlm layer for predicting the [MASK] entities. Specifically, we fix all the parameters of K-ADAPTER and only update the parameters of the mlm layer using a masked language modeling (MLM) loss. We adopt the raw WikiText-2 dataset (181M). We train the mlm layer with one single 16G P100 for 2 epochs. We set the max sequence length to be 512, batch size to be 1024 and warmup step to be 0.

---

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