

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

Ruize Wang^{1*}, Duyu Tang², Nan Duan², Zhongyu Wei¹, Xuanjing Huang¹,
Jianshu Ji³, Guihong Cao³, Daxin Jiang², Ming Zhou²

¹Fudan University, Shanghai, China

²Microsoft, Beijing, China

³Microsoft, Redmond WA, USA

{rzwang18, zywei, xjhuang}@fudan.edu.cn

{dutang, nanduan, jianshuji, gucao, djiang, mingzhou}@microsoft.com

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, the historically injected knowledge would be flushed away. To address this, we propose K-ADAPTER, a framework that retains the original parameters of the pre-trained model fixed and supports the development of versatile knowledge-infused model. Taking RoBERTa as the backbone 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 multiple adapters can be efficiently trained in a distributed way. As a case study, we inject two kinds of knowledge in this work, including (1) factual knowledge obtained from automatically aligned text-triplets on Wikipedia and Wikidata and (2) linguistic knowledge obtained via dependency parsing. Results on three knowledge-driven tasks, including relation classification, entity typing, and question answering, demonstrate that each adapter improves the performance and the combination of both adapters brings further improvements. Further analysis indicates that K-ADAPTER captures versatile knowledge than RoBERTa.¹

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., 2020), have established state-of-the-art performances on various NLP downstream tasks. Despite the huge success of these pre-trained models in empirical studies, recent studies suggest that models learned in such an unsupervised manner struggle to capture rich knowledge. For example, Poerner et al. (2020) suggest that although language models do well in reasoning about the surface form of entity names, they fail in capturing rich factual knowledge. Kassner and Schütze (2020) observe that BERT mostly did not learn the meaning of negation (e.g. “not”). 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., 2020; Peters et al., 2019; He et al., 2020; Xiong et al., 2020). Most previous works (as shown in Table 1) augment the standard language modeling objective with knowledge-driven objectives and update the entire model parameters. Although these methods obtain better performance on downstream tasks, they struggle at supporting the development of versatile models with multiple kinds of knowledge injected (Kirkpatrick et al., 2017). When new kinds of knowledge are injected, model parameters need to be retrained so that previously injected knowledge would fade away. Meanwhile, the resulting models produce entangled representations, so that it is hard to investigate the effect of each kind of knowledge.

In this paper, we propose K-ADAPTER, a flexible and simple framework that supports the infusion of multiple kinds of knowledge into large pre-trained models. K-ADAPTER leaves the original representation of a pre-trained model unchanged and exports different representations for different types of infused knowledge. This is achieved by

*Work is done during internship at Microsoft. Zhongyu Wei and Duyu Tang are corresponding authors.

¹Codes are publicly available at <https://github.com/microsoft/K-Adapter>

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., 2020)	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., 2020)	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

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

the integration of compact neural models dubbed adapters. **Adapters are knowledge-specific models plugged outside of a pre-trained model, whose inputs are the output hidden-states of intermediate layers of the pre-trained model.** 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 and linguistic knowledge obtained by applying off-the-shell dependency parser to web texts. In the pre-training phase, **we train two adapters independently.** 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. Case study and probing experiments indicate that K-ADAPTER captures versatile knowledge than RoBERTa.

2 Related Work

Our work relates to the area of injecting knowledge into pre-trained models. 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. **SenseBERT** (Levine et al., 2020) 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) 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., 2020) integrates fact triples from knowledge graph. For each entity, it sample incoming and outgoing 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,

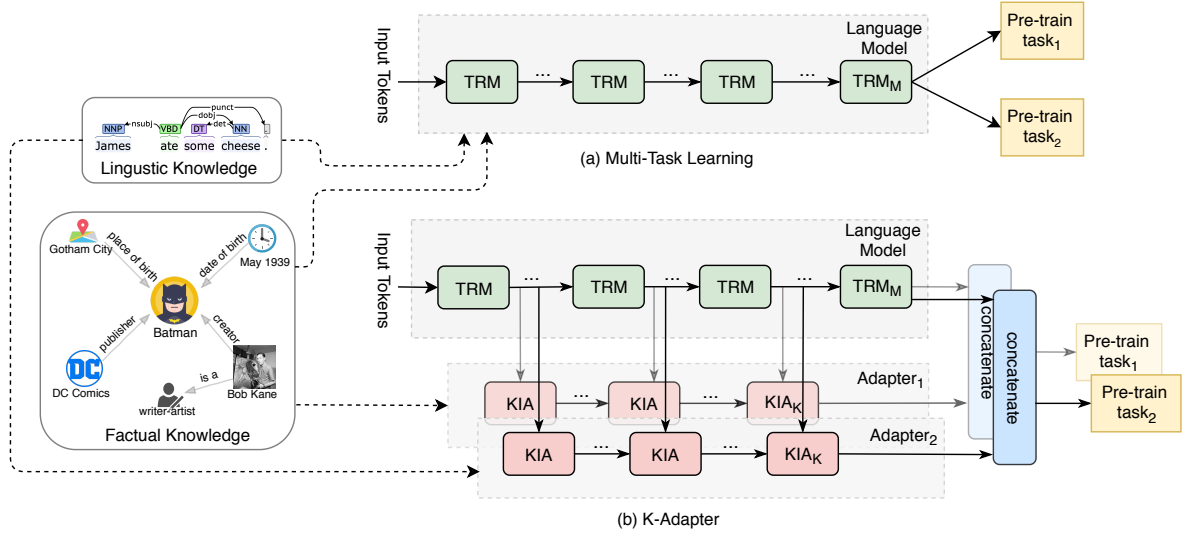


Figure 1: (a) Pre-trained language models inject multiple kinds of knowledge with multi-task learning. Model parameters need to be retrained when injecting new kinds of knowledge, which may result in the catastrophic forgetting (b) Our K-ADAPTER injects multiple kinds of knowledge by training adapters 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.

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 address 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 keeps 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 disen-

tangled representation. Specifically, adapters are knowledge-specific models (with few parameters) plugged outside of a pre-trained model. The inputs of adapters are the output hidden-states of intermediate layers of the pre-trained model. Each adapter is pre-trained independently on different tasks for injecting discriminative 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. In this section, we first describe the structure of our adapter, and then present the process of pre-training knowledge-specific adapters.

3.1 Adapter Structure

In this work, we present a different adapter structure as shown in Figure 2, which is referred to as the knowledge-specific adapter. In contrast to Houlsby et al. (2019) add adapter layers into each transformer layer, our adapter works as outside plug-ins. 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

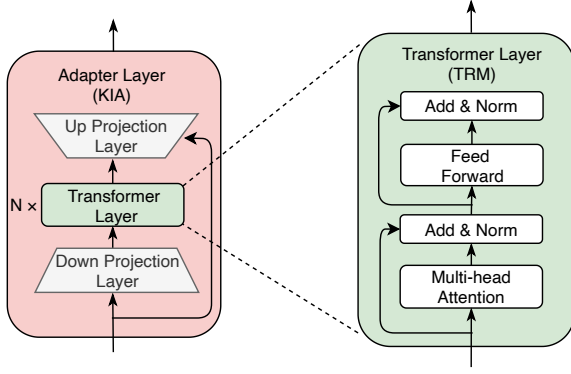


Figure 2: Structure of the adapter layer (left). The adapter layer consists of two projection layers and $N=2$ transformer layers, and a skip-connection between two projection layers.

adapter layers among different transformer layers of the pre-trained model. We concatenate the output hidden feature of the transformer layer in 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 concatenate the last hidden features of the pre-trained model and 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 adapter 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

We use RoBERTa_{LARGE} (L=24, H=1024, A=16, 355M params) implementation by Huggingface² 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 as A_A , the hidden dimension of down-projection and up-projection layers as H_d and 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 lay-

²<https://github.com/huggingface/transformers>

ers 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. It should be noticed that RoBERTa is fixed during training and the parameters of adapters are trainable and initialized randomly. Then we describe how to inject different knowledge into knowledge-specific adapters as below.

3.3 Factual Adapter

Factual knowledge can be described as the basic information that is concerned with facts. In this work, we acquire factual knowledge from the relationships among entities in natural language. We extract a sub-dataset T-REx-rc from T-REx (Eisaihar et al., 2018) which is a large scale alignment dataset between Wikipedia abstracts and Wikidata triples. We discard all relations having less than 50 entity pairs, collecting 430 relations and 5.5M sentences. In order to inject factual knowledge, we propose to pre-train a knowledge-specific adapter called facAdapter on the relation classification task. This task requires a model to classify relation labels of given entity pairs based on context. Specifically, the last hidden features of RoBERTa and facAdapter are concatenated as the input representation, and the pooling layer is applied to the input representations of the given entities. Then, we concatenate two entity representations to perform relation classification.

3.4 Linguistic Adapter

Linguistic knowledge is implicitly contained in natural language texts, e.g., syntactic and semantic information. In this work, we acquire linguistic knowledge from dependency relationships among words in natural language text. We build a dataset consisting of 1M examples. In particular, we run the off-the-shell dependency parser from Stanford Parser³ on a part of Book Corpus (Zhu et al., 2015). To inject linguistic knowledge, we pre-train another knowledge-specific adapter called linAdapter on the task of dependency relation prediction. This task aims to predict the head index of each token in the given sentence. We concatenate the last hidden features of RoBERTa and linAdapter as the input representation, and then apply a linear layer

³<http://nlp.stanford.edu/software/lex-parser.html>

Model	OpenEntity			FIGER		
	P	R	Mi-F ₁	Acc	Ma-F ₁	Mi-F ₁
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., 2021)	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 (w/o knowledge)	74.47	74.91	76.17	56.93	82.56	77.90
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
K-ADAPTER (F+L)	78.99	76.27	77.61	61.81	84.87	80.54

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

to input representations of each token to perform classification. More training details of facAdapter and linAdapter can be found in the Appendix.

4 Experiments

We evaluate our K-ADAPTER on three knowledge-driven downstream tasks, i.e., entity typing, question answering and relation classification. Furthermore, we conduct detailed analyses with the case study and probing experiments to explore the effectiveness and 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. Implementation details, and statistics of datasets are in the Appendix.

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. We evaluate our models on OpenEntity (Choi et al., 2018) and FIGER (Ling et al., 2015) following the same split setting as Zhang et al. (2019). 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. As for OpenEntity, we adopt micro F_1 score as the final metric to represent the model performance. As for FIGER, we adopt strict accuracy, loose macro, loose micro F_1 scores (Ling and Weld, 2012) for evaluation following the same evaluation criteria used in previous works.

Baselines NFGEC (Shimaoka et al., 2016) employs attentive recursive neural networks to compose context representations. KEPLER (Wang et al., 2021) integrates factual knowledge with the supervision of the knowledge embedding objective. 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. K-ADAPTER (w/o knowledge) consists of a RoBERTa model and an adapter without being injected knowledge. Other baseline models are described in Section 2.

Results and Discussion The results on OpenEntity and FIGER are shown in Table 2. K-ADAPTER (F+L) achieves consistent improvements across these datasets. As for OpenEntity, our RoBERTa achieve better results than other baseline models. K-ADAPTER (F+L) further achieves improvement of 1.38% F_1 over RoBERTa, which means factual knowledge and linguistic knowledge help to predict the types more accurately. As for FIGER, it covers more entity types, and is more fine-grained than OpenEntity. Compared with WKLM, K-ADAPTER (F+L) improves the macro F_1 by 2.88%, micro F_1 by 2.54% and accuracy by 1.60%. This demonstrates that K-ADAPTER (F+L) benefits fine-grained entity typing.

In addition, we further conduct several experiments on our ablated model K-ADAPTER (w/o knowledge), to explore whether the performance gains came from introducing knowledge or additional parameters. Results show that K-ADAPTER (F) significantly outperforms K-ADAPTER (w/o knowledge). Moreover, it is worth noting that on OpenEntity dataset, K-ADAPTER (w/o knowledge) even performs slightly worse than RoBERTa.

Model	SearchQA		Quasar-T		CosmosQA
	EM	F ₁	EM	F ₁	Accuracy
BiDAF (Seo et al., 2017)	28.60	34.60	25.90	28.50	-
AQA (Buck et al., 2018)	40.50	47.40	-	-	-
R ³ (Wang et al., 2018a)	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., 2018b)	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)	61.85	67.17	46.20	52.86	80.93
K-ADAPTER (L)	61.15	66.82	45.66	52.39	80.76
K-ADAPTER (F+L)	61.96	67.31	46.32	53.00	81.83

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

These results demonstrate that our model gains improvement from knowledge instead of more parameters. Thus, for simplicity, we don’t discuss K-ADAPTER (w/o knowledge) in the following experiments.

4.2 Question Answering

We conduct experiments on two question answering (QA) tasks, i.e., commonsense QA and open-domain QA. Commonsense QA aims to answer questions with commonsense. We adopt CosmosQA (Huang et al., 2019) dataset to evaluate our models. CosmosQA requires commonsense-based reading comprehension, formulated as multiple-choice questions. To fine-tune our models for CosmosQA, the input token sequence is modified as “<SEP>context</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 QA aims to answer questions using external resources such as collections of documents and webpages. We evaluate our models on two public datasets, i.e., Quasar-T (Dhingra et al., 2017) and SearchQA (Dunn et al., 2017). Specifically, we first retrieve paragraphs corresponding to the question 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. (2017) 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 F_1 (Ling and Weld, 2012) 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. BiDAF (Seo et al., 2017) adopts a bi-directional attention network. AQA (Buck et al., 2018) proposes to re-write questions and aggregate the answers generated by the re-written questions. R³ (Wang et al., 2018a) is a reinforced model making use of a ranker for selecting most confident paragraph. Evidence Agg. (Wang et al., 2018b) proposes making use of the aggregated evidence from across multiple paragraphs. WKLM (Xiong et al., 2020) 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 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. 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 perfor-

Model	P	R	F ₁
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., 2021)	70.43	73.02	71.70
RoBERTa	70.17	72.36	71.25
RoBERTa + multitask	70.18	73.11	71.62
K-ADAPTER (F)	69.39	74.59	71.89
K-ADAPTER (L)	68.85	75.37	71.96
K-ADAPTER (F+L)	70.14	74.04	72.04

Table 4: Results on the relation classification dataset TACRED.

mance 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. K-ADAPTER models achieve better results compared to other baselines. This indicates that 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, K-ADAPTER (F+L) makes significant improvement of 4.01% F_1 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 QuasarT dataset, K-ADAPTER (F+L) also outperforms WKLM by 3.1% F_1 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 adopt a large-scale relation classification dataset TACRED (Zhang et al., 2017). To fine-tune our models for this task, 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 adopt micro F_1 score as the metric to represent the model performance as previous works.

Baselines C-GCN (Zhang et al., 2018) employs graph convolutional networks to model dependency trees. BERT-large (Baldini Soares et al., 2019) is a baseline BERT-large model. BERT+MTB (Baldini Soares et al., 2019) is a method of training relation representation without supervision from a knowledge base by matching the blanks. Other baseline models 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 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. This directly demonstrates injecting knowledge individually in K-ADAPTER way would help models make full use of knowledge.

4.4 Case Study

Table 5 gives a qualitative comparison example between K-ADAPTER and RoBERTa on relation classification dataset TACRED. The results show that, in most cases, the wrongly predicted logit value of RoBERTa and the logit value of the true label are actually quite close. For example, given “*New Fabris* closed down *June 16*”, RoBERTa predicts “no_relation”, but the true label “city_of_birth” ranks in second place. If a model could correctly predict the relationship between “*New Fabris*” and “*June 16*”, then it needs to know that “*New Fabris*” is a company. Thanks to the factual knowledge in K-ADAPTER, it can help the model from predicting “no_relation” to predicting the correct category label.

In addition, we utilize a LAMA (Language Model Analysis) probe (Petroni et al., 2019) to examine models’ ability to memorize factual knowledge. Specifically, the LAMA probing task is under a zero-shot setting, which requires the language model to answer cloze-style questions about relational facts without fine-tuning. For example, given “Simon Bowman was born in [MASK]” as the input, models are asked to predict the correct token

Input	True label	Model	Predicted label	Predicted logits
His former student <u>Mark Devlin</u> of the University of Pennsylvania was co-leader of the other, known as the Microwave Anisotropy Telescope .	schools_attended	K-Adapter	['schools_attended', 'no_relation', 'founded']	[12.6, 9.5, 5.2]
		RoBERTa	['no_relation', 'founded', 'member_of']	[9.1, 6.5, 5.0]
Graham had been in custody in <u>Vancouver</u> , British Columbia, since June .	cities_of_residence	K-Adapter	['cities_of_residence', 'countries_of_residence', 'no_relation']	[13.5, 6.8, 6.6]
		RoBERTa	['countries_of_residence', 'country_of_death', 'alternate_names']	[7.1, 7.0, 6.8]
Vladimir Ladyzhenskiy of Russia died after she suffered a <u>shock</u> in the final of the spa world championship in Heinola, a southern city of Finland, on Saturday .	cause_of_death	K-Adapter	['cause_of_death', 'origin', 'no_relation']	[11.0, 7.6, 7.1]
		RoBERTa	['no_relation', 'cause_of_death', 'origin']	[6.3, 5.9, 5.5]
You can't have a good season unless it starts well," said Bill Martin, co-founder of <u>ShopperTrak</u> , on Saturday .	founded_by	K-Adapter	['founded_by', 'member_of', 'employee_of']	[10.2, 9.3, 7.3]
		RoBERTa	['no_relation', 'founded_by', 'employee_of']	[10.0, 8.5, 5.4]

Table 5: A case study for K-ADAPTER and RoBERTa on relation classification dataset TACRED. Underlines and wavy lines highlight the subject entities and object entities respectively. We report the top 3 ranked predictions.

Query	Answer	Model	Generation
The native language of Mammootty is <u>[MASK]</u> .	Malayalam	RoBERTa	English, Tamil, Hindi, Sanskrit, Arabic, Chinese
		K-ADAPTER	Malayalam , Tamil, Hindi, Mandarin, English
Ravens can <u>[MASK]</u> .	fly	RoBERTa	win, play, score, lose, run, drink, fly , roll, wait
		K-ADAPTER	fly , swim, sing, shoot, kill, go, fish, drink, die
Sometimes virus causes <u>[MASK]</u> .	infection	RoBERTa	cancer, death, illness, blindness, paralysis
		K-ADAPTER	cancer, illness, death, infection , disease
Sunshine Coast, British Columbia is located in <u>[MASK]</u> .	Canada	RoBERTa	Florida, California, Texas, Hawaii, Mexico
		K-ADAPTER	Canada , Vancouver, Victoria, BC, Australia
iPod Touch is produced by <u>[MASK]</u> .	Apple	RoBERTa	Apple , Samsung, Qualcomm, LG, Microsoft
		K-ADAPTER	Apple , HTC, Samsung, Motorola, Intel

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

which is masked. 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, which demonstrate that K-ADAPTER captures richer factual knowledge than RoBERTa. More details about this probing experiments can be found in the Appendix A and E.4.

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. Detailed analyses further suggest that K-ADAPTER captures richer factual and commonsense knowledge than RoBERTa, and provide insights on the effectiveness of knowledge infusion. In future work, we will infuse more types of knowledge, and apply our framework to more pre-trained models.

Acknowledgments

This work is partially supported by National Natural Science Foundation of China (No. 71991471), Science and Technology Commission of Shanghai Municipality Grant (No.20dz1200600, 21QA1400600).

References

- Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *ACL*, pages 2895–2905.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795.
- Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Andrea Gesmundo, Neil Houlsby, Wojciech Gajewski, and Wei Wang. 2018. Ask the Right Questions: Active Question Reformulation with Reinforcement Learning. In *ICLR*.
- Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *ACL*, pages 87–96.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL*, pages 2978–2988.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.
- Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. 2017. Quasar: Datasets for question answering by search and reading. In *arXiv preprint arXiv:1707.03904*.
- Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. In *ArXiv preprint arXiv:1704.05179*.
- Hady ElSahar, Pavlos Vougiouklis, Arslan 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 *LREC*.
- Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.
- Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. 2020. **BERT-MK: Integrating graph contextualized knowledge into pre-trained language models**. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2281–2290, Online. Association for Computational Linguistics.
- 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 *ICML*, pages 2790–2799.
- Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *EMNLP*, pages 2391–2401.
- 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*, pages 7811–7818, Online. Association for Computational Linguistics.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and et al. 2017. **Overcoming catastrophic forgetting in neural networks**. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.
- Anne Lauscher, Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2019. Informing unsupervised pretraining with external linguistic knowledge. *arXiv preprint arXiv:1909.02339*.
- Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. **SenseBERT: Driving some sense into BERT**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667, Online. Association for Computational Linguistics.
- Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distantly supervised open-domain question answering. In *ACL*, pages 1736–1745.
- Xiao Ling, Sameer Singh, and Daniel S. Weld. 2015. Design challenges for entity linking. *TACL*, 3:315–328.
- Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22–26, 2012, Toronto, Ontario, Canada*. AAAI Press.
- 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*.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*, pages 2227–2237.

- Matthew E Peters, Mark Neumann, IV Logan, L Robert, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. In *EMNLP*, pages 43–54.
- Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language Models as Knowledge Bases? In *EMNLP*, pages 2463–2473.
- Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. [E-BERT: Efficient-yet-effective entity embeddings for BERT](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online. Association for Computational Linguistics.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hananneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.
- Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2016. An attentive neural architecture for fine-grained entity type classification. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction(AKBC)*, pages 69–74.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, pages 5998–6008.
- Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald Tesauro, Bowen Zhou, and Jing Jiang. 2018a. R3: Reinforced ranker-reader for open-domain question answering. In *AAAI*.
- Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018b. Evidence aggregation for answer re-ranking in open-domain question answering. In *ICLR*.
- Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *ACL*, pages 189–198.
- Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. [Kepler: A unified model for knowledge embedding and pre-trained language representation](#). *Transactions of the Association for Computational Linguistics*, 9(0):176–194.
- Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model. In *ICLR*.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. [Xlnet: Generalized autoregressive pretraining for language understanding](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *EMNLP*, pages 2205–2215.
- Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *EMNLP*, pages 35–45.
- Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *ACL*, pages 1441–1451.
- Yukun Zhu, Ryan Kiros, Rich 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 *ICCV*, pages 19–27.

Appendix

A Probing Experiments

Although K-ADAPTER models have shown superior performance on knowledge-driven downstream tasks, it does not directly provide insights into whether our models infuse richer factual 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 is under a zero-shot setting, which requires the language model to answer cloze-style questions about relational facts without fine-tuning, e.g., “Simon Bowman was born in [MASK]”. The model needs 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-TREx, which are aimed at factual knowledge. We also conduct probe experiments on LAMA-UHN (Poerner et al., 2020), a more “factual” subset of LAMA, 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-ADAPTER (F) which is infused with factual knowledge, with other baseline models.

Results and Discussion Results are shown in Table 7. It is surprising that BERT_{LARGE} performs better than RoBERTa_{LARGE}. 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. As for LAMA, com-

pared to RoBERTa_{LARGE}, K-ADAPTER obtains 2.2% and 1.2% P@1 improvement across GoogleRE and TREx, 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.

B Pre-Training Details

B.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 TREx-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.

B.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.

C Applying K-adapter on Downstream Tasks

For the downstream tasks, the key point here is the combination of the pre-trained model’s representations and adapter’s representations, that is to say: leveraging the general information of the pre-trained model on one hand, and the specific knowledge in the adapter on the other. To use K-ADAPTER for downstream tasks is very simple as shown in Figure 5. Usually, when we use pre-trained language models such as BERT and RoBERTa for downstream tasks, we feed the output features from the pre-trained model into the task-specific layer, and then do the corresponding downstream task. As for the K-ADAPTER, we fine-tune it just like what the original BERT or RoBERTa does. We concatenate the output features of the pre-trained model with the features of the adapter, and then feed them to the task-specific layer.

Corpus	Models					
	ELMo	ELMo5.5B	TransformerXL	BERT-large	RoBERTa _{LARGE}	K-ADAPTER
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 7: P@1 on LAMA and LAMA-UHN across Google-RE and T-REx corpora.

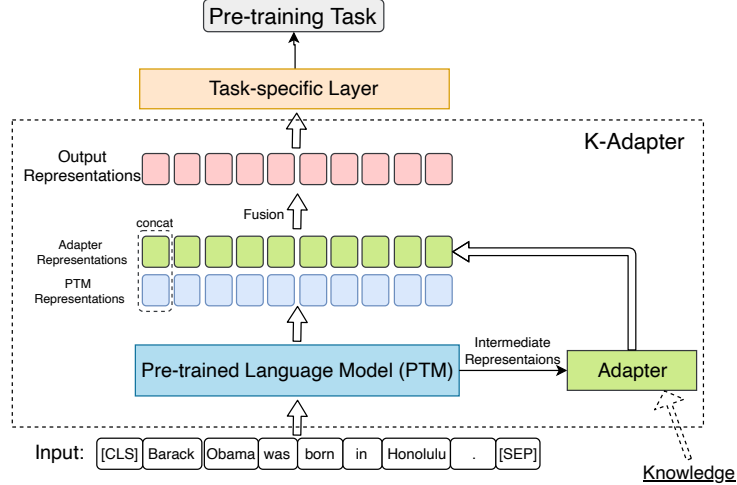


Figure 3: An overview of our K-ADAPTER to inject specific knowledge by training a knowledge-specific adapter on the pre-training task.

D Dataset statistics

In Table 8, we present the statistics of one relation classification dataset TACRED, and two entity typing datasets OpenEntity and FIGER. In Table 9, we present the statistics of one commonsense QA dataset CosmosQA and two open-domain QA datasets SearchQA and Quasar-T.

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 relation classification dataset TACRED and entity typing datasets, i.e., Open Entity and FIGER.

E Fine-tuning Details

We implement our experiments using Huggingface⁴. For all fine-tuning experiments, we use AdamW as the optimizer. The parameters of adapters are fixed during the fine-tuning process

⁴<https://github.com/huggingface/transformers>

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

Table 9: The statistics of the question answering datasets, i.e., CosmosQA, SearchQA and Quasar-T.

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 reproducibility.

E.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 single 16G P100). For K-ADAPTER (L), the best performance is achieved at batch size=4,

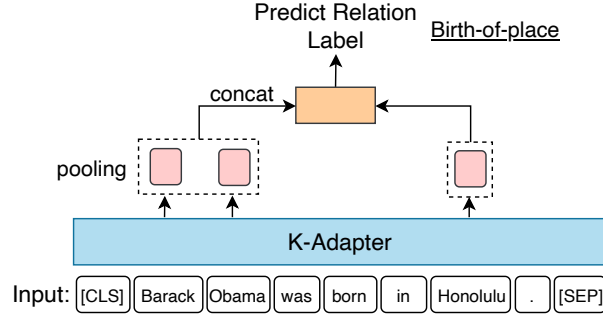


Figure 4: An example of using relation classification as a pre-training task to inject knowledge into K-ADAPTER: given “Barack Obama was born in Honolulu”, and then predicts the relationship between “Barack Obama” and “Honolulu” is “Birth-of-place”.

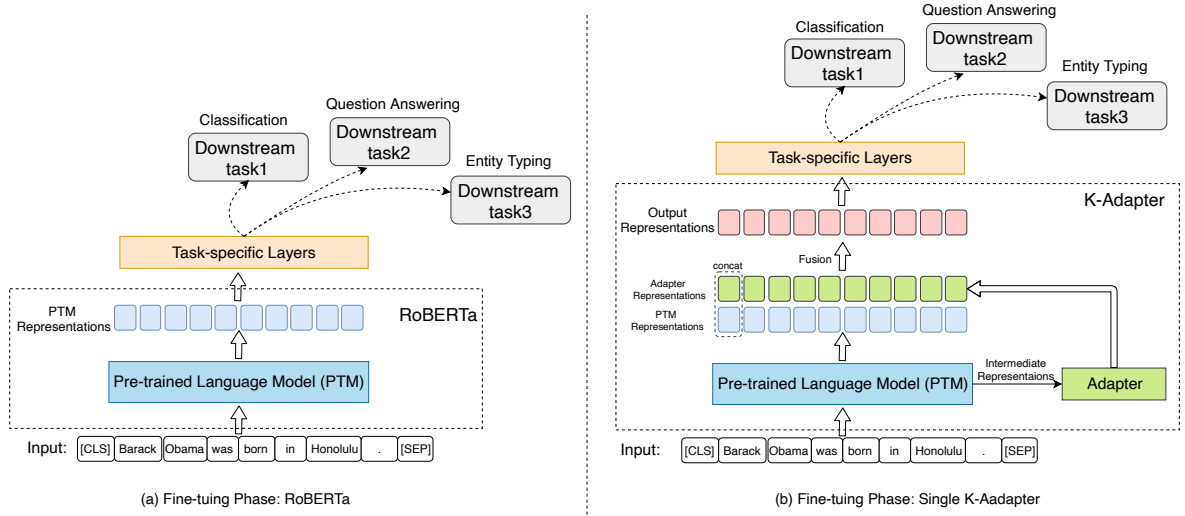


Figure 5: Fine-tuning K-ADAPTER just like what the original RoBERTa or BERT does.

$lr=5e-6$, warmup=1000 (it takes about 2 hours to get the best result running on single 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 single 16G P100). For F1GUR 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.

E.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 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.

E.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.

E.4 Probing Experiments

We implement our probing experiments using LAMA⁵. 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.

⁵<https://github.com/facebookresearch/LAMA>