Parameter-Efficient Domain Knowledge Integration from Multiple Sources for Biomedical Pre-trained Language Models

Qiuhao Lu[1], **Dejing Dou**[1,2], **and Thien Huu Nguyen**[1]
[1]Dept. of Computer and Information Science, University of Oregon, Eugene, OR, USA
[2]Baidu Research
{luqh, dou, thien}@cs.uoregon.edu

Abstract

Domain-specific pre-trained language models (PLMs) have achieved great success over various downstream tasks in different domains. However, existing domain-specific PLMs mostly rely on self-supervised learning over large amounts of domain text, without explicitly integrating domain-specific knowledge, which can be essential in many domains. Moreover, in knowledge-sensitive areas such as the biomedical domain, knowledge is stored in multiple sources and formats, and existing biomedical PLMs either neglect them or utilize them in a limited manner. In this work, we introduce an architecture to integrate domain knowledge from diverse sources into PLMs in a parameter-efficient way. More specifically, we propose to encode domain knowledge via *adapters*, which are small bottleneck feed-forward networks inserted between intermediate transformer layers in PLMs. These knowledge adapters are pre-trained for individual domain knowledge sources and integrated via an attention-based knowledge controller to enrich PLMs. Taking the biomedical domain as a case study, we explore three knowledge-specific adapters for PLMs based on the UMLS Metathesaurus graph, the Wikipedia articles for diseases, and the semantic grouping information for biomedical concepts. Extensive experiments on different biomedical NLP tasks and datasets demonstrate the benefits of the proposed architecture and the knowledge-specific adapters across multiple PLMs.

1 Introduction

In the past few years, large pre-trained language models (PLMs) have demonstrated superior performance over various downstream tasks in natural language processing (NLP), such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2019), GPT-3 (Brown et al., 2020), etc. These PLMs mainly depend on self-supervised pre-training on large amounts of textual data, e.g., Wikipedia, and can be conveniently applied to downstream tasks via fine-tuning. Despite the great success of these general PLMs, their performance over domain-specific texts is relatively poor due to domain shifts (Ma et al., 2019). Consequently, recent studies construct domain-specific PLMs through fine-tuning or pre-training from scratch over domain corpora, such as BioBERT (Lee et al., 2020), ClinicalBERT (Huang et al., 2019), SciBERT (Beltagy et al., 2019), etc.

Since these PLMs are mostly pre-trained on unstructured free texts, a common issue among the aforementioned general and domain-specific PLMs is their lack of specific structured knowledge, which results in their incompetence on knowledge-driven tasks (Rogers et al., 2020). For instance, some studies point out PLMs are insufficient to well capture factual knowledge from text (Poerner et al., 2019; Wang et al., 2020, 2021).

To enrich PLMs with external knowledge, some efforts have been made recently (Yao et al., 2019; Zhang et al., 2019; Kim et al., 2020; Levine et al., 2020; Wang et al., 2021). A common theme among these approaches is the incorporation of an auxiliary knowledge-driven training objective. For instance, KG-BERT (Yao et al., 2019) integrates world/factual knowledge from Wikipedia via knowledge graph completion; KEPLER (Wang et al., 2021) introduces a Knowledge Embedding objective and combine it with the language modeling objective for joint optimization. Despite the improved performance of these knowledge-enriched PLMs over downstream tasks, there are three limitations. First, these approaches, either training from scratch or fine-tuning over off-the-shelf checkpoints, need to optimize the entire model, which is quite expensive. Second, they mostly focus on single-source knowledge incorporation, e.g., an encyclopedia, and neglect knowledge from multiple sources. This limits the utilization of potential knowledge, especially for knowledge-sensitive ar-

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 3855–3865

eas such as the biomedical domain where knowledge is stored in multiple sources and formats (Jin et al., 2019; Lee et al., 2020). Third, most of existing knowledge integration approaches focus on general domain knowledge, while domain knowledge infusion for PLMs is underexplored.

To address these limitations, we propose to perform knowledge integration for PLMs via adapters (Rebuffi et al., 2017; Houlsby et al., 2019; Pfeiffer et al., 2020, 2021; Wang et al., 2020). Basically, adapters are lightweight neural networks that are placed inside PLMs. When fine-tuning a PLM, the original parameters of the PLM are fixed and only the adapters are fine-tuned. This makes adapters a parameter-efficient alternative to full model fine-tuning. Another benefit of adapters is their independent nature, where multiple adapters can be trained independently without interfering with each other. As such, we propose to enrich PLMs with adapters that are independently pre-trained for different sources of domain knowledge.

In this paper, we propose an architecture that aims to integrate domain knowledge from multiple sources via knowledge-specific adapters to enrich PLMs. We take the biomedical domain as a case study, as it is a knowledge-sensitive area where domain-knowledge is essential for various NLP applications. Specifically, we explore three knowledge-specific adapters for PLMs based on the UMLS Metathesaurus graph, the Wikipedia articles for diseases, and the semantic grouping information for biomedical concepts. We also incorporate an attention-based knowledge controller module that aims to adaptively adjust the activation levels of the adapters, which also brings some explainability as it shows the importance of the adapters for a task. The experimental results show that by equipping PLMs with domain knowledge from multiple sources via the proposed architecture, their overall performance gets consistently improved across tasks and datasets. Moreover, the pre-trained adapters can be directly integrated with multiple PLMs, demonstrating transferability of the architecture.

The contributions of this work can be summarized as follows:

- We propose a novel architecture that incorporates **D**iverse **A**dapters for **K**nowledge **I**ntegration (DAKI) into PLMs. It integrates domain knowledge from multiple sources adaptively via an attention-based knowledge controller. The architecture demonstrates effectiveness, transferability, explainability as well as parameter-efficiency in experiments.

- Taking the biomedical domain as a case study, we specifically investigate and pre-train three knowledge adapters based on the UMLS Metathesaurus graph, the Wikipedia articles for diseases, and the semantic grouping information for biomedical concepts. Such adapters serve as off-the-shelf modules and can be used in a plug-and-play manner via DAKI.

- Extensive experiments on different biomedical NLP tasks and datasets demonstrate the benefits of the proposed knowledge-specific adapters and DAKI.

2 Related Work

This study is essentially related to two lines of research: knowledge integration for PLMs and domain-specific PLMs (biomedical PLMs in particular).

There has been a surge of research on knowledge injection for PLMs in recent years (Yao et al., 2019; Zhang et al., 2019; Peters et al., 2019; Kim et al., 2020; Levine et al., 2020; Lauscher et al., 2020; Pereira et al., 2020; Sun et al., 2020; He et al., 2020a; Wang et al., 2021). These studies aim to integrate knowledge from an external knowledge source, e.g., Wikipedia, into PLMs by augmenting the training objective with a knowledge-driven regularization. As mentioned above, these methods are limited in the sense that they mostly focus on single-source knowledge, and require full model training. K-adapter (Wang et al., 2020) addresses some of these issues by introducing linguistic and factual adapters into RoBERTa, but the adapters are treated equally in their work. Also, general domain knowledge, such as factual knowledge (Zhang et al., 2019; Sun et al., 2020; He et al., 2020a; Wang et al., 2021), commonsense knowledge (Lauscher et al., 2020; Pereira et al., 2020), and linguistic knowledge (Levine et al., 2020) are prioritized in these studies, while domain knowledge is somewhat underexplored (Michalopoulos et al., 2020).

Biomedical NLP continues to be an active area of research in the past few years. There have been several biomedical PLMs proposed and have proven to be successful in various domain tasks (Lee

et al., 2020; Peng et al., 2019; Huang et al., 2019; Alsentzer et al., 2019). As variants of BERT (Devlin et al., 2019) in the biomedical domain, these PLMs are mostly pre-trained on large amounts of domain-specific corpora, such as the PubMed texts (Peng et al., 2019; Lee et al., 2020) and clinical notes (Huang et al., 2019; Alsentzer et al., 2019), and do not explicitly incorporate domain knowledge in the pre-training stage.

This work differs from the aforementioned studies in that we are the first to integrate biomedical domain-specific knowledge from multiple sources into PLMs via an adapter-based architecture. The knowledge integration process is flexible, efficient and transferable.

3 Diverse Adapters for Knowledge Integration (DAKI)

In this section, we introduce a mechanism, i.e., DAKI, that encodes domain knowledge from diverse sources into PLMs via knowledge-specific adapters. We first introduce the adapter module along with the overall architecture of DAKI, and then discuss the knowledge-specific adapters for the biomedical domain. In the end we explain the attention-based knowledge controller that is leveraged to adaptively integrate these adapters.

3.1 Pre-trained Language Models with Adapters

Adapter An *adapter* module is a simple and lightweight neural network placed within a large pre-trained base model, and in NLP the base model is usually a pre-trained language model such as BERT (Devlin et al., 2019). Generally, adapters are placed in or between the intermediate transformer layers in a PLM, and the placement defines two paradigms. One puts the adapters *inside* the intermediate transformer layers (Houlsby et al., 2019; Pfeiffer et al., 2020, 2021), and the other puts the adapter *between* and *outside* the intermediate transformer layers (Wang et al., 2020). In this work, we choose the latter paradigm for its flexibility and extensibility, as shown in Figure 2. Instead of updating the entire language model, only the adapters are updated during fine-tuning on downstream tasks. This strategy demonstrates parameter-efficiency and scalability while achieving similar performance to full fine-tuning, and has been actively explored as an alternative for transfer learning in recent NLP studies (Houlsby et al., 2019; Pfeiffer et al., 2020, 2021; Wang et al., 2020; Rücklé et al., 2020).

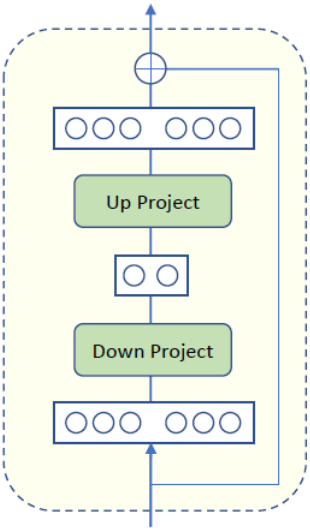

Figure 1: Adapter module.

In this work, we leverage a simple yet effective bottleneck feed-forward network as the adapter module. Essentially, the adapter module consists of a residual connection and two projection layers with `LeakyReLU` as the activation, as shown in Figure 1. The size of adapters is controlled by the bottleneck, and is usually much smaller than that of the base PLM, i.e., $d_{\text{bottleneck}} \ll d_{\text{PLM}}$, where d_{PLM} refers to the dimension of hidden-states in the base PLM. In our case, the bottleneck dimension is set to 128 for all experiments. Note that a more complex adapter is possible, such as two projection layers along with a stack of transformer layers (Wang et al., 2020), but at the cost of efficiency.

Architecture Figure 2 illustrates the overall architecture of DAKI. Essentially, the architecture contains three main components, i.e., the base PLM, the knowledge-specific adapters, and the adapter integration module. DAKI theoretically supports any transformer-based structure as the base PLM, such as BERT (Devlin et al., 2019), ALBERT (Lan et al., 2019), RoBERTa (Liu et al., 2019), etc. Each knowledge-specific adapter contains several adapter modules and they are inserted at certain layers of the base PLM. Each adapter module takes as input the addition of the hidden-states of the transformer layer and the output of the previous adapter module. The adapter modules do not share weights with each other. Motivated by the fact that knowledge from different sources should

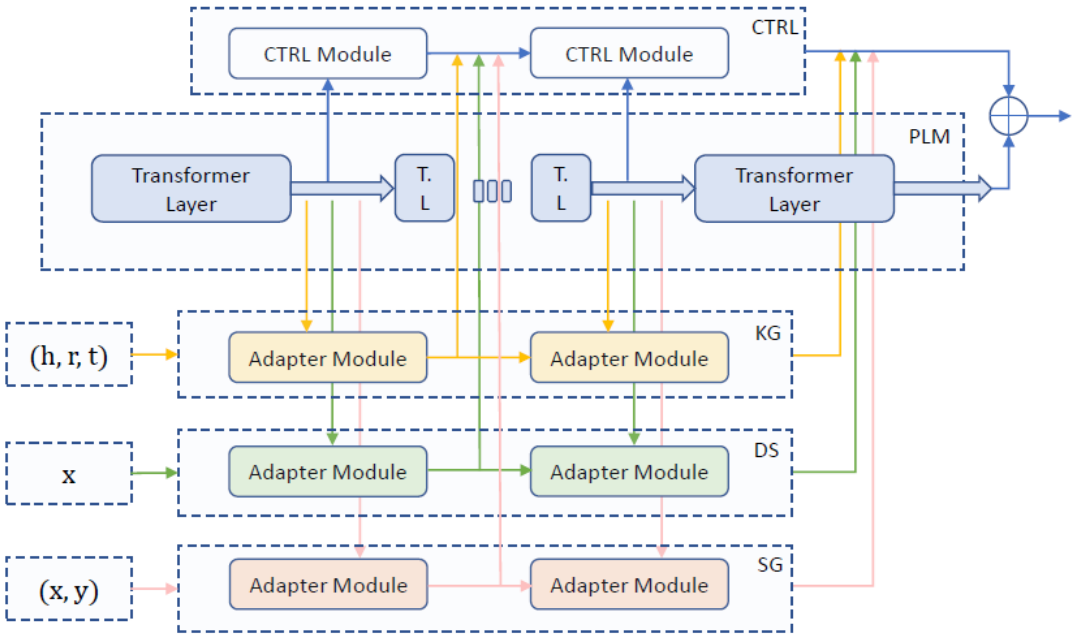

Figure 2: Architecture of DAKI. CTRL refers to the knowledge controller. Linear layers are omitted for simplicity.

have different level of activation over downstream tasks, we incorporate a *knowledge controller* to adaptively integrate the knowledge adapters. Details are explained in Section 3.3.

When pre-training an adapter, we take the addition of the output of the last adapter module and the last-hidden-states of the base PLM as the final output, and use it for the pre-training task. Note that during adapter pre-training, the knowledge controller is dropped and the base PLM is frozen. When applying DAKI to downstream tasks, we take the addition of the output of the knowledge controller and the last-hidden-states of the base PLM as the final output, and use it for the downstream task.

The benefits of this architecture is threefold. First, adapters are independent and do not interact during pre-training, which means they have perfect memory of the knowledge, thus avoiding the forgetting issue in multi-task learning. Second, it demonstrates flexibility and extensibility as it is easy to remove, add or replace the adapters. Third, the usage of DAKI is as simple as a general PLM, since its output can be considered the last-hidden-states of a PLM.

In this work, we use ALBERT-xxlarge-v2 (Lan et al., 2019) as the base PLM. We investigate three knowledge-specific adapters based on the UMLS Metathesaurus graph, the Wikipedia articles for diseases, and the semantic grouping information for biomedical concepts. Details are explained in Section 3.2. Each adapter contains three adapter modules and they are placed at layers $\{0, 5, 11\}$. Note that the number and placement of adapter modules can be flexible, and in this study we follow the same strategy with (Wang et al., 2020) where three modules are distributed at the bottom, middle, and top layer.

Adapter	Source	Size	Format
KG	UMLS Metathesaurus	$1,772,248$	(h, r, t)
DS	Wikipedia	$14,617$	x
SG	Semantic Network	$333,005$	(x, y)

Table 1: Statistics of the datasets for pre-training KG, DS, SG. The formats are triples, passages, and textual definitions with labels, respectively.

3.2 Adapters Pre-training

In this work, we investigate three independent adapters based on three sources of knowledge, i.e., the UMLS Metathesaurus knowledge graph (KG), the Wikipedia articles for diseases (DS), and the semantic grouping information for medical concepts (SG). The statistics of the corresponding datasets for pre-training are shown in Table 1. These knowledge-specific adapters serve as examples for encoding domain knowledge from various sources, and can be easily extended or replaced with alternative knowledge sources. For clarity, we use PLM-KG, PLM-DS and PLM-SG to denote the

model that is used to pre-train the adapters in this section.

3.2.1 Knowledge Graph Adapter (KG)

Knowledge graphs encode real-world knowledge in the form of triples, i.e., *(h, r, t)* where *h* and *t* refer to the head and tail entity and *r* is the relation between them. Knowledge graphs have been actively explored in recent studies of language model pre-training or fine-tuning, as they reveal the relationships between real-world entities that are hidden from surface texts.

To leverage the knowledge encoded in the UMLS Metathesaurus graph[1], we pre-train an adapter that aims to capture the connectivity patterns between medical entities through knowledge graph completion. More specifically, we treat the triples in UMLS as textual sequences and feed them into the PLM-KG encoder. Then the representation of the triple is used as input to a binary classification layer for plausibility prediction.

In particular, given a triple *(h, r, t)*, we first convert it to a textual sequence by concatenating the words in the names of *h*, *r*, and *t*. For example, for a triple *(diffuse adenocarcinoma of the stomach, disease has normal tissue origin, gastric mucosa)*, the constructed input sequence is:

`[CLS]` diffuse adenocarcinoma of the stomach `[SEP]` disease has normal tissue origin `[SEP]` gastric mucosa `[SEP]`

We then use the PLM-KG model to encode the sequence, and use the representation for the `[CLS]` token in the last layer to predict the plausibility of the triple, i.e., determining whether the triple is valid or not. The adapter parameters in this model are optimized with a binary cross-entropy loss:

$$\mathcal{L}_{\text{KG}} = -\sum_{t \in \{\mathcal{T}^+ \cup \mathcal{T}^-\}} (y \log \hat{y}_1 + (1-y) \log \hat{y}_0) \tag{1}$$

where y is the ground-truth label and $\hat{y}_0, \hat{y}_1$ refer to the output prediction probabilities. $\mathcal{T}^+$ and $\mathcal{T}^-$ are the positive and negative triple set. Here, the negative set $\mathcal{T}^-$ is constructed by replacing the head or tail entity in a positive triple with a random entity.

3.2.2 Disease Adapter (DS)

It is crucial to equip pre-trained language models with disease knowledge for medical NLP tasks, as it bridges the gap between disease terms and their textual descriptions. For example, in the medical natural language inference task (NLI), the premise-hypothesis pair *(No history of blood clots or DVTs has never had chest pain prior to one week ago, Patient has angina)* is more likely to be correctly classified as `entailment` if the model specifically knows that angina refers to chest pain.

To leverage the disease knowledge, we pre-train an adapter that aims to infer disease names based on their textual descriptions. More specifically, for each disease, a new passage is formed by collecting the textual content from its Wikipedia article[2]. We then randomly substitute 75% of the *disease terms* in the passage with `[MASK]` in the passage and optimize the PLM-DS model via a masked language modeling (MLM) objective.

Formally, let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_K\}$ denote the indexes of the masked tokens in the passage T, where K is the number of masked tokens. Then T_{Π} and $T_{\neg\Pi}$ represent the set of masked and observed tokens in the passage, respectively. Then the training objective for the adapter parameters is described as:

$$\mathcal{L}_{\text{DS}} = \mathcal{L}_{\text{mlm}}(T_{\Pi}|T_{\neg\Pi}) = -\frac{1}{K}\sum_{k=1}^{K} \log p(t_{\pi_k}|T_{\neg\Pi}) \tag{2}$$

where $p(t_{\pi_k}|T_{\neg\Pi})$ is the probability of predicting t_{π_k} given the unmasked tokens $T_{\neg\Pi}$, estimated by a softmax layer.

3.2.3 Semantic Grouping Adapter (SG)

To provide a proper and consistent categorization of concepts in the Metathesaurus, the UMLS Semantic Network groups concepts according to the semantic types that have been assigned to them. Each concept is assigned to at least one semantic type from a total of 127 semantic types. For certain purposes, however, a coarser-grained categorization is desirable, and hence the semantic types are aggregated into 15 semantic groupings (McCray et al., 2001). Such aggregation ensures the semantic coherence between concepts in the same group[3]. This property would help pre-trained language models capture the connectivity between medical concepts, as well as between their descriptive texts.

To leverage the semantic grouping information, we pre-train an adapter that aims to predict the se-

[1]The data is available at https://www.nlm.nih.gov/research/umls.

[2]This data is proposed by (He et al., 2020b).

[3]The data is available at https://semanticnetwork.nlm.nih.gov.

Datasets	MEDIQA-2019			TRECQA-2017			MEDNLI	BC5CDR	NCBI
Metrics(%)	Accuracy	MRR	Precision	Accuracy	MRR	Precision	Accuracy	F1	F1
BERT	64.95	82.72	66.49	74.61	56.17	52.55	75.95	83.09	85.14
ClinicalBERT	67.30	84.78	70.59	77.00	52.56	56.62	81.50	84.90	87.25
SciBERT	68.47	84.47	68.07	77.23	54.57	57.54	80.94	86.16	87.24
BioBERT	68.29	83.61	72.78	77.12	49.84	57.25	81.86	85.99	87.70
diseaseBERT	66.40	83.33	68.94	75.33	56.41	54.01	77.29	83.47	86.81
umlsBERT	62.87	83.91	63.62	70.20	54.17	46.69	81.65	84.54	86.23
RoBERTa	72.49	86.74	74.67	75.33	51.76	54.01	81.65	83.04	85.83
ALBERT	76.54	88.46	81.41	75.09	58.57	53.03	85.48	84.28	87.56
diseaseALBERT	79.49	90.00	84.02	80.10	57.21	**62.40**	86.15	84.71	87.69
DAKI-BERT	69.47	85.06	70.17	77.95	54.65	58.27	77.85	83.43	85.67
DAKI-BioBERT	72.54	87.33	77.46	78.55	54.17	59.04	83.41	**86.51**	**89.01**
DAKI-RoBERTa	73.98	89.22	76.39	77.23	51.92	58.48	81.65	83.36	86.01
DAKI-ALBERT	**80.22**	**91.22**	**84.36**	**80.33**	**58.65**	62.31	**86.85**	84.86	87.86

Table 2: Performance of DAKI over downstream tasks QA, NLI and NER.

mantic groupings of concepts in UMLS based on their textual definitions. More specifically, for a UMLS concept with corresponding textual definition, we encode the definition with the PLM-SG model and feed the `[CLS]` representation into a linear layer for classification. The model is optimized with cross-entropy loss:

$$\mathcal{L}_{\text{SG}} = -\sum_{i=1}^{15} y_i \log \hat{y}_i \tag{3}$$

where y_i is the ground-truth label and $\hat{y}_i$ refers to the output prediction probabilities.

3.3 Knowledge Controller

The *knowledge controller* is essentially a separate adapter with additional linear layers, which is distributed at the same layers with the knowledge adapters, as shown in Figure 2. This module aims to adaptively integrate the knowledge adapters by assigning them different importance weights, as opposed to simple concatenation of the outputs of adapters (Wang et al., 2020). At each layer i where a adapter module is placed, three linear transformation modules are employed, i.e., Q_i, K_i, V_i, as motivated by (Vaswani et al., 2017). Essentially, Q_i takes the hidden-states of the controller as the input, and the output is considered as the *query* signal. K_i in contrast takes the hidden-states of the adapters as the input, and the output serves as the *key* signal. The *value* signal is the hidden-states of the adapters. Then the attention weights are computed for each adapter and the weighted sum of the hidden-states of adapters are fed into V_i, the output of which is regarded as the final output of the knowledge controller at layer i:

$$\begin{aligned}
\mathbf{Q}_i &= \mathbf{W}_{Q_i}\mathbf{H}_{C_i} + \mathbf{b}_{Q_i} \\
\mathbf{K}_i &= \mathbf{W}_{K_i}\mathbf{H}_{D_i} + \mathbf{b}_{K_i} \\
\mathbf{A}_i &= \text{softmax}(\mathbf{Q}_i\mathbf{K}_i^T)\mathbf{H}_{D_i} \\
\mathbf{Z}_i &= \mathbf{W}_{V_i}\mathbf{A}_i + \mathbf{b}_{V_i}
\end{aligned} \tag{4}$$

where $\mathbf{H}_{C_i}$ are the hidden-states of the controller and $\mathbf{H}_{D_i}$ are the concatenation of the hidden-states of the adapters at layer i. $\mathbf{W}_{Q_i}, \mathbf{b}_{Q_i}, \mathbf{W}_{K_i}, \mathbf{b}_{K_i}, \mathbf{W}_{V_i}, \mathbf{b}_{V_i}$ are trainable parameters of the linear modules at each layer.

4 Experiments

In this section, we evaluate the DAKI architecture over three knowledge-driven downstream tasks in biomedical NLP, where we aim to show the effectiveness of the knowledge integration method. We also investigate some desirable properties of the architecture.

4.1 Setup

4.1.1 Downstream tasks

We perform evaluation over three knowledge-driven biomedical NLP tasks, i.e., Question Answering (QA), Natural Language Inference (NLI) and Named Entity Recognition (NER)[4].

[4]The datasets for downstream tasks are available at https://github.com/heyunh2015/diseaseBERT.

Datasets	MEDIQA-2019			TRECQA-2017			MEDNLI	BC5CDR	NCBI	A.P	C.P
Metrics(%)	Acc	MRR	Pre	Acc	MRR	Pre	Acc	F1	F1		
DAKI	80.22	91.22	84.36	80.33	58.65	62.31	86.85	84.86	87.86	79.63	-
w/o ctrl	78.32	88.27	81.68	79.38	56.09	61.19	86.78	84.58	86.99	78.14	-1.49
w/o KG	79.49	90.72	85.42	80.45	57.85	62.74	85.94	83.93	87.43	79.33	-0.30
w/o DS	78.86	89.61	82.37	79.62	57.85	61.53	85.86	83.99	87.82	78.61	-1.02
w/o SG	73.15	86.33	80.77	79.26	57.61	60.43	85.37	84.29	86.87	77.12	-2.51
w/o ctrl,DS,SG	78.14	89.61	80.11	79.86	59.13	62.11	86.29	83.76	87.37	78.48	-1.15
w/o ctrl,KG,SG	77.78	89.44	83.54	79.98	57.45	61.96	84.18	83.46	87.33	78.34	-1.29
w/o ctrl,KG,DS	77.51	89.83	83.44	80.69	58.01	64.01	86.51	84.25	87.26	79.05	-0.58
ALBERT	76.15	84.67	83.19	77.12	57.93	56.68	86.01	85.38	86.81	77.10	-2.53

A.P means average of performance and C.P means change of performance. ALBERT means removing everything.

Table 3: Ablation analysis.

QA We conduct the medical QA experiments on MEDIQA-2019 (Abacha et al., 2019) and TRECQA-2017 (Abacha et al., 2017), where the task is cast as a regression problem. Essentially, for an given question-answer pair, a numerical score ranging from -2 to 2 is assigned by experts, indicating the quality of the answer to the question, and the task is to predict the score. We use a simple prediction model, where each pair is encoded with a PLM or DAKI, and the representation for `[CLS]` is fed into a linear layer on top for prediction.

NLI We conduct the medical NLI experiments on MEDNLI (Romanov and Shivade, 2018), where the task is to classify a given premise-hypothesis pair into a class of `entailment`, `neutral`, or `contradiction`. Similarly, each pair is encoded with a PLM or DAKI, and the `[CLS]` representation is fed into a classification head on top.

NER We conduct the medical NER experiments on NCBI (Doğan et al., 2014) and BC5CDR-disease (Wei et al., 2016), where the task is to classify tokens of sentences into a class of `B`, `I`, or `O` (Peng et al., 2019; He et al., 2020b), with a PLM or DAKI as the encoder.

Note that our models for downstream tasks QA, NLI, and NER follow those in diease-BERT/diseaseALBERT (He et al., 2020b) to be comparable. We also inherit the hyper-parameters for such models from (He et al., 2020b). In particular, we employ AdamW as the optimizer and set learning rates of {1e-5, 1e-5, 5e-5}, and the batch sizes of {8, 16, 16} respectively for the tasks.

Baselines We take three PLMs, i.e., BERT-base-uncased (Devlin et al., 2019), RoBERTa-base (Liu et al., 2019), ALBERT-xxlarge-v2 (Lan et al., 2019), as well as their main biomedical variants as the baselines, including ClinicalBERT (Alsentzer et al., 2019), SciBERT (Beltagy et al., 2019), BioBERT-v1.1 (Lee et al., 2020), umls-BERT (Michalopoulos et al., 2020) and disease-BERT/diseaseALBERT (He et al., 2020b).

4.1.2 Pre-training Adapters

When pre-training the adapters KG, DS, SG, we use the ALBERT-xxlarge-v2 (Lan et al., 2019) as the base PLM, and set the adapter size to 128. We use Adam as the optimizer and set learning rates of {1e-6, 2e-4, 1e-5}, batch sizes of {256, 16, 256}, maximum sequence lengths of {16, 256, 128} and training epochs of {2, 10, 1}, respectively for the corresponding adapters.

4.2 Results

Table 2 shows the performance of our proposed architecture, i.e., DAKI, over three biomedical NLP tasks across five datasets. Generally, one main observation from the table is that equipping PLMs with DAKI significantly improve their performance on these biomedical tasks, as reflected in DAKI-BERT, DAKI-RoBERTa and DAKI-ALBERT, demonstrating the effectiveness of the architecture. Moreover, although DAKI-BERT outperforms BERT across all metrics, it only performs comparably or poorer than ClinicalBERT, SciBERT and BioBERT. We conjecture that it is due to lack of the knowledge in their pre-training data, i.e., the MIMIC-III clinical notes (Johnson et al., 2016), the Semantic Scholar papers (Ammar et al., 2018), and the PubMed articles, respectively.

Transferability Another advantage of DAKI is transferability, due to its flexible architecture and implementation. In this work, we have three

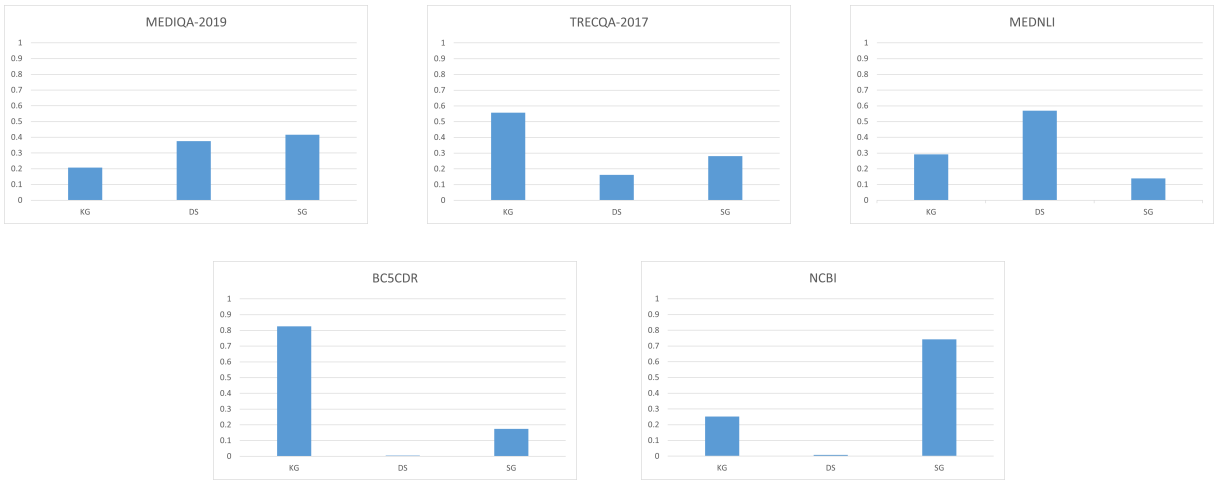

Figure 3: Activation levels of the adapters KG, DS, SG over the downstream tasks. We calculate the softmax activations in the last layer for each adapter, and the activations are averaged over all instances in the test set.

adapters and they are all pre-trained with ALBERT as the base PLM. All the DAKI variants in Table 2 are the corresponding PLMs equipped with such pre-trained adapters (based on ALBERT). As such, the performance gain of the DAKI variants show that the knowledge in the adapters is transferable across BERT versions, making it possible to use adapters as off-the-shelf modules in a a plug-and-play manner. Interestingly, even for the knowledge-augmented BioBERT, incorporating DAKI yields a performance boost over all tasks, which further demonstrates the transferability of the architecture.

Ablation Study To investigate the influence of each component of DAKI, we perform an ablation study and show the results in Table 3. We first remove the knowledge controller from DAKI, and take the addition of the outputs of adapters, without adaptive adjustment. Then we remove each adapter while keeping the controller. Finally we apply accumulative ablation by removing both of them. Essentially, the results of the ablated versions demonstrate varying degrees of performance drop, indicating the necessity of each component.

Explainability We expect the *knowledge controller* to bring some explainability, as it adaptively activates the adapters when fine-tuning over the downstream tasks. We show the average softmax attention weights of the adapters in Figure 3, which we assume to reflect the activation levels of them. Basically, the activations of adapters are different across tasks and datasets, except that KG and SG seem to have more impact on BC5CDR and NCBI.

Parameter-efficiency An advantage of using DAKI for incorporating knowledge is that only one version of the PLM is needed to accommodate multiple knowledge sources. In particular, without adapters, fine-tuning a PLM with one knowledge source will produce a new version of PLM. For three knowledge sources in our work, we will need to have $3 \times N_{\text{PLM}}$ parameters. With DAKI, this number is reduced to $N_{\text{PLM}} + 3 \times N_{\text{adapter}} + N_{\text{ctrl}}$. Considering ALBERT as an example, this amount to a reduction of $2 \times N_{\text{PLM}} - 3 \times N_{\text{adapter}} - N_{\text{ctrl}} \approx 2 \times 223M - 4M = 442M$ parameters.

5 Conclusion

In this paper, we propose DAKI, an adapter-based architecture that adaptively integrates knowledge from multiple sources into pre-trained language models. We take the biomedical domain as a case study, and specifically explore three different sources of biomedical knowledge and integrate them with DAKI. The experimental results prove the effectiveness of the architecture, and also show that the architecture demonstrates parameter-efficiency, transferability, and explainability to some degree. The objective of this work is not to update state-of-the-art results on the benchmarks, but to provide an alternative method of domain knowledge integration for PLMs, especially from multiple sources of knowledge.

Acknowledgements

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112

and the NSF grant CNS-1747798 to the IUCRC Center for Big Learning. This research is also based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, ODNI, IARPA, the Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

References

Asma Ben Abacha, Eugene Agichtein, Yuval Pinter, and Dina Demner-Fushman. 2017. Overview of the medical question answering task at trec 2017 liveqa. In *Proceedings of the Text Retrieval Conference (TREC)*.

Asma Ben Abacha, Chaitanya Shivade, and Dina Demner-Fushman. 2019. Overview of the mediqa 2019 shared task on textual inference, question entailment and question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task (BioNLP)*, pages 370–379.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop (ClinicalNLP)*, pages 72–78.

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, et al. 2018. Construction of the literature graph in semantic scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3, Industry Papers (NAACL-HLT)*, pages 84–91.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3606–3611.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (NAACL-HLT)*, pages 4171–4186.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Bin He, Xin Jiang, Jinghui Xiao, and Qun Liu. 2020a. Kgplm: Knowledge-guided language model pre-training via generative and discriminative learning. *arXiv preprint arXiv:2012.03551*.

Yun He, Ziwei Zhu, Yin Zhang, Qin Chen, and James Caverlee. 2020b. Infusing disease knowledge into bert for health question answering, medical inference and disease name recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4604–4614.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2790–2799.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.

Qiao Jin, Bhuwan Dhingra, William Cohen, and Xinghua Lu. 2019. Probing biomedical embeddings from language models. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP (RepEval)*, pages 82–89.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Bosung Kim, Taesuk Hong, Youngjoong Ko, and Jungyun Seo. 2020. Multi-task learning for knowledge graph completion with pre-trained language models. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pages 1737–1743.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

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 (ACL)*, pages 4656–4667.

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*.

Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2019. Domain adaptation with bert-based domain classification and data selection. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo)*, pages 76–83.

Alexa T McCray, Anita Burgun, and Olivier Bodenreider. 2001. Aggregating umls semantic types for reducing conceptual complexity. *Studies in health technology and informatics*, 84(0 1):216.

George Michalopoulos, Yuanxin Wang, Hussam Kaka, Helen Chen, and Alex Wong. 2020. Umls-bert: Clinical domain knowledge augmentation of contextual embeddings using the unified medical language system metathesaurus. *arXiv preprint arXiv:2010.10391*.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task (BioNLP)*, pages 58–65.

Lis Pereira, Xiaodong Liu, Fei Cheng, Masayuki Asahara, and Ichiro Kobayashi. 2020. Adversarial training for commonsense inference. In *Proceedings of the 5th Workshop on Representation Learning for NLP (Repl4NLP)*, pages 55–60.

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

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of The 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Association for Computational Linguistics.

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

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2019. Bert is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised qa. *arXiv preprint arXiv:1911.03681*.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pages 506–516.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics (TACL)*, 8:842–866.

Alexey Romanov and Chaitanya Shivade. 2018. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1586–1596.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2020. Adapterdrop: On the efficiency of adapters in transformers. *arXiv preprint arXiv:2010.11918*.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuan-Jing Huang, and Zheng Zhang. 2020. Colake: Contextualized language and knowledge embedding. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pages 3660–3670.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pages 6000–6010.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.

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 (TACL)*, 9:176–194.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Jiao Li, Thomas C Wiegers, and Zhiyong Lu. 2016. Assessing the state of the art in biomedical relation extraction: overview of the biocreative v chemical-disease relation (cdr) task. *Database*, 2016.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1441–1451.