A Unified Knowledge Graph Augmentation Service for Boosting Domain-specific NLP Tasks

Ruiqing Ding[1,2], **Xiao Han***[3], **Leye Wang***[1,2]
[1]Key Lab of High Confidence Software Technologies (Peking University), Ministry of Education, China
[2]School of Computer Science, Peking University, Beijing, China
[3]School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai, China
ruiqingding@stu.pku.edu.cn, xiaohan@mail.shufe.edu.cn, leyewang@pku.edu.cn

Abstract

By focusing the pre-training process on domain-specific corpora, some domain-specific pre-trained language models (PLMs) have achieved state-of-the-art results. However, it is under-investigated to design a unified paradigm to inject domain knowledge in the PLM fine-tuning stage. We propose **KnowledgeDA**, a *unified* domain language model development service to enhance the task-specific training procedure with domain knowledge graphs. Given domain-specific task texts input, **KnowledgeDA** can automatically generate a domain-specific language model following three steps: (i) localize domain knowledge entities in texts via an embedding-similarity approach; (ii) generate augmented samples by retrieving replaceable domain entity pairs from two views of both knowledge graph and training data; (iii) select high-quality augmented samples for fine-tuning via confidence-based assessment. We implement a prototype of **KnowledgeDA** to learn language models for two domains, *healthcare* and *software development*. Experiments on domain-specific text classification and QA tasks verify the effectiveness and generalizability of **KnowledgeDA**.

1 Introduction

Although general NLP models such as GPT-3 (Brown et al., 2020) have demonstrated great potential, they may not consistently perform well in domain-specific tasks like healthcare (Kwon et al., 2019) and programming (Liu et al., 2019b). This is because most pre-trained language models are trained on general-domain corpora, e.g., OpenWebText (Radford et al., 2019) and C4 (Raffel et al., 2022). However, the words or knowledge entities frequently used in a specific domain are typically different from those in a general domain. For instance, scientific texts use different words than general texts, with only a 42% overlap (Beltagy et al., 2019). Consequently, general PLMs struggle to capture many important domain entities that rarely appear in general corpora. Therefore, it is necessary to develop a suitable training mechanism for domain-specific NLP tasks.

In general, two steps are needed for domain-specific NLP model development: (i) language model pretraining and (ii) task-specific model training (Gu et al., 2021). Most existing studies focus on pretraining. In particular, to learn domain-specific word embeddings, they retrain PLMs with domain-specific corpora, including ClinicalBERT (Alsentzer et al., 2019), BioBERT (Lee et al., 2020), SciBERT (Beltagy et al., 2019), etc. In contrast, how to improve the second step (i.e., task-specific training) is under-investigated. A common practice is directly fine-tuning the task-specific model with annotated data (Gu et al., 2021). However, *it is difficult to obtain abundant annotated data for a domain-specific task, as labeling often requires domain experts' knowledge* (Yue et al., 2020); without sufficient data, direct fine-tuning may not lead to a satisfactory performance due to overfitting (Si et al., 2020). Some studies propose task-dependent methods to train task-specific models by introducing some types of domain knowledge (Zhu et al., 2022), but they are hard to be generalized to other tasks (Tushev et al., 2022).

Then, a research question appears: **can we introduce domain knowledge to task-specific model training in a *unified* way?** To answer the question, two main issues need to be addressed: (i) *where to find a unified format of domain knowledge?* (ii) *how to improve the task-specific training of various domains' models in a unified way?*

On one hand, the domain knowledge graph (KG) is an effective and standardized knowledge base for a specific domain (Abu-Salih, 2021). KGs have been constructed for various domains such as cybersecurity (Jia et al., 2018), social-impact funding (Li et al., 2020b), and healthcare (Li et al.,

*Corresponding authors

Findings of the Association for Computational Linguistics: ACL 2023, pages 353–369
July 9-14, 2023

2020a; Zhang et al., 2020), which emphasizes the wide availability of domain KGs. Hence, *domain KG could be a feasible source for unified domain knowledge*. On the other hand, data augmentation (DA) is a data-space approach to enrich training data to avoid overfitting regardless of the task-specific model structure. They are often *task-agnostic* (Longpre et al., 2020), i.e., not specified to any particular task. This property inspires us that *it may be possible to design a unified DA process to introduce domain knowledge to task-specific model training*. However, current DA methods in NLP are mostly proposed for general texts (Wei and Zou, 2019), and the performance on domain-specific tasks is limited (Feng et al., 2021). In general, domain-specific DA is still an under-researched direction (Feng et al., 2021).

To fill this research gap, by exploiting domain KGs, we propose **KnowledgeDA**, a novel and unified three-step procedure to perform domain-specific DA: (i) *domain knowledge localization* to map phrases in the text to entities in the domain KG; (ii) *domain knowledge augmentation* to fully utilize the KG and the training data to achieve domain-specific augmentation; and (iii) *augmentation quality assessment* to single out high-quality augmented data for fine-tuning the task-specific model. Specifically: (i) To the best of our knowledge, this is one of the pioneering efforts toward proposing a unified development process for domain-specific NLP models, especially focusing on task-specific model training. (ii) **KnowledgeDA** consists of three core steps, *domain knowledge localization*, *domain knowledge augmentation*, and *augmentation quality assessment*. We implement a prototype of **KnowledgeDA**, which can automatically learn domain-specific models given domain-specific texts, especially in *healthcare* domain. (iii) Experiments are run on text classification and QA tasks (English and Chinese) mainly in healthcare. Results show that **KnowledgeDA** can obtain $\sim 4\%$ improvement compared to direct fine-tuning, and significantly outperform existing DA methods (Wei and Zou, 2019; Yue and Zhou, 2020). The source codes are available[1].

2 Related Work

Domain-specific Knowledge-augmented NLP Methods. To improve domain-specific NLP model development, a general strategy is introducing domain knowledge (Zhu et al., 2022). For zero and few-shot text classification tasks, KPT (Hu et al., 2022) incorporates external knowledge into the projection between a label space and a label word space. For text generation, KG-BART (Liu et al., 2021) proposes a novel knowledge graph augmented pre-trained language generation model to promote the ability of commonsense reasoning. For question answering and dialogue, some work use external knowledge bases to inject commonsense, like KaFSP (Li and Xiong, 2022), KG-FiD (Yu et al., 2022), etc. Besides task-dependent methods, there are also some *unified* training strategies to incorporate knowledge (domain-specific corpora) into PLMs, leading to domain-specific PLMs such as BioBERT (Lee et al., 2020), SciBERT (Beltagy et al., 2019), ClinicalBERT (Alsentzer et al., 2019), and UmlsBERT (Michalopoulos et al., 2021). Also, there are three primary techniques to integrate knowledge graphs and PLMs: (i) pre-training a PLM from scratch by using KG or other structural knowledge/texts (Feng et al., 2022; Huang et al., 2022); (ii) adapting a given PLM to incorporate KG information with new network layers in task-specific training/fine-turning (Zhang et al., 2022b; Yasunaga et al., 2022; Kang et al., 2022); (iii) augmenting training data with KGs during task-specific training/fine-tuning, e.g., PHICON (Yue and Zhou, 2020). Our work also attempts to improve the domain-specific NLP model development in a unified manner. Different from PLM, we focus on task-specific NLP model fine-tuning (Gu et al., 2021). Hence, our proposed **KnowledgeDA** can be used with domain-specific PLMs together to construct NLP models.

Text Data Augmentation (DA). DA has received increasing interest, especially low-resource situations(Feng et al., 2021). In general, there are three types of text DA methods: (i) *Rule-based* techniques, e.g., EDA (Wei and Zou, 2019), adopt token-level random perturbation operations including random insertion, deletion, and swap; (ii) *Interpolation-based* techniques, pioneered by MIXUP (Zhang et al., 2018), interpolate the inputs and labels of two or more real examples. Follow-ups include SwitchOut (Wang et al., 2018), MixText (Chen et al., 2020), etc; (iii) *Generator-based* techniques, e.g., LAMBADA (Anaby-Tavor et al., 2020) and GPT3Mix (Yoo et al., 2021), learn generators by fine-tuning the large language generation models (e.g., GPT) on the training data to generate

[1] https://github.com/RuiqingDing/KnowledgeDA

new samples. Basically, three types of methods can be used together as they augment data from diverse perspectives. However, regardless of the type, most existing studies do not explicitly introduce domain knowledge. PHICON (Yue and Zhou, 2020) attempts to use the domain-entity dictionary for text DA, which replaces an entity mention in a sentence with another same-category entity. Compared to PHICON, **KnowledgeDA** further considers relationships in the domain KG; besides, **KnowledgeDA** introduces other newly-designed components, e.g., augmentation quality assessment, to ensure high-quality augmentation.

3 The KnowledgeDA Framework

3.1 Workflow of KnowledgeDA

To facilitate the development of domain-specific NLP models, we propose a unified domain KG service, **KnowledgeDA**, which can achieve explicit domain knowledge injection by domain-specific DA. Challenges to be addressed include:

C1. How to discover domain knowledge in texts? Detecting entities in a text is the first step to link the text with a knowledge base. A domain entity may have multiple expressions, e.g., *lungs* and *pulmonary* share a similar meaning in the healthcare domain. It is important to deal with synonyms.

C2. How to ensure that the augmented texts retain the domain information and are semantically correct? We aim to achieve interpretable data augmentation through explicit domain knowledge injection. The domain information and the semantic correctness of augmented samples are desirable to be kept after data augmentation.

C3. How to ensure the quality of augmented texts? As PLMs grow larger, simple DA method becomes less beneficial (Feng et al., 2021). It is essential to select beneficial samples from all the augmented samples for efficient fine-tuning.

To address the above challenges, we design corresponding modules in **KnowledgeDA** (shown in Figure 1): (i) ***domain knowledge localization***, which locates the mentions of domain KG entities in texts; (ii) ***domain knowledge augmentation***, which incorporates a dual-view DA strategy by considering both domain KG and training data; (iii) ***augmentation quality assessment***, which retains beneficial augmented samples for fine-tuning using a confidence-based strategy. When the task data and the PLM (e.g., BERT) are given, **KnowledgeDA** can automatically conduct data augmentation based on built-in domain KGs and output the final domain task-specific model.

3.2 Module 1: Domain Knowledge Localization

Detecting entities in texts can identify domain-specific objects and the relations between them. Considering that an entity may correspond to multiple mentions (Florian et al., 2004), *exact* string matching will lead to a low matching rate. Although there are some open entity detection tools, like TAGME (Ferragina and Scaiella, 2010) and BLINK (Wu et al., 2020), and some studies achieve supervised non-exact matching of entities and mentions (Hu et al., 2019), the performance on domain-specific entities can not be guaranteed. Then, we use an annotation-free string-similarity-based strategy (Bunescu and Pasca, 2006; Karadeniz and Özgür, 2019) to discover *non-exact but correct* mappings between mentions in the text and entities in the KG. Specifically, we calculate the inner product of word embeddings as string similarity (Wu et al., 2020).

As seen in Figure 1, we follow the NLP preprocessing pipeline and match the processed text with KG. During preprocessing, we add the entity strings in KG to the dictionary of *tokenizer* to avoid word segmentation errors, e.g., 'cerebral embolis' should be treated as a medical term rather than being splitted into two words. Also, we use *POS Tagger* and *Lemmatizer* to convert each token to the canonical form (lemma), e.g., the lemma for 'coughed' is 'cough'. While knowledge localization, we extract the entities' embeddings and the mentions' embeddings from the PLM, and then calculate the similarity between them. We consider the pair of a mention and the most similar KG entity as a match if the similarity score is larger than a threshold λ (0.9 in our implementation).

An example in healthcare is illustrated in Figure 2. Without similarity match, we will ignore that *scour* and *diarrhea* are analogous. Through localization, some relations between entities can be also constructed, e.g., fever and scour are symptoms of pneumonia and respiratory syndrome. These will be used in the next module for data augmentation.

3.3 Module 2: Dual-view Domain Knowledge Augmentation

After locating the domain knowledge, i.e., entity mentions in the text, the next step is to replace these mentions with other *relevant entity words* for

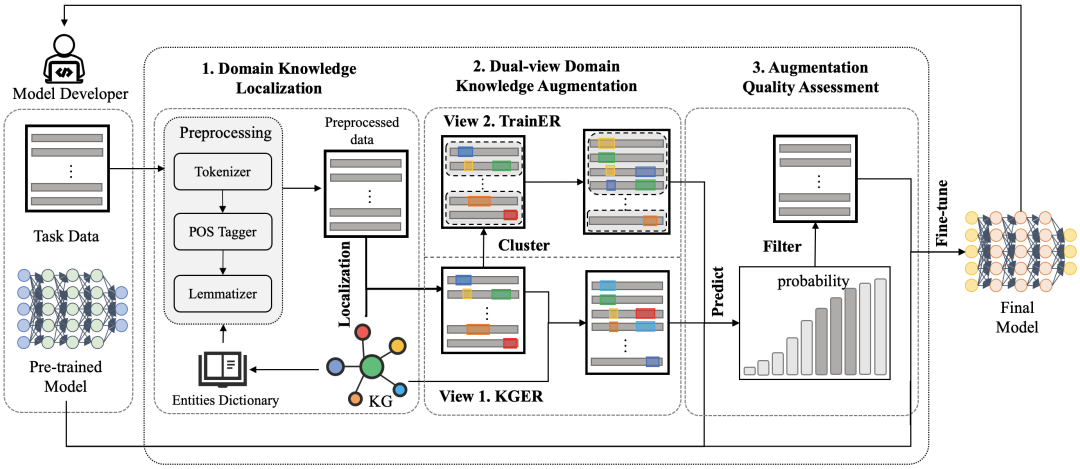

Figure 1: Overview of **KnowledgeDA**: the user only needs to upload the task text data (i.e., training data) and specifies the pre-trained language models (e.g., BERT). All the domain knowledge injection procedures are automatically conducted. Finally, a well-performed domain-specific NLP model will be obtained from **KnowledgeDA**.

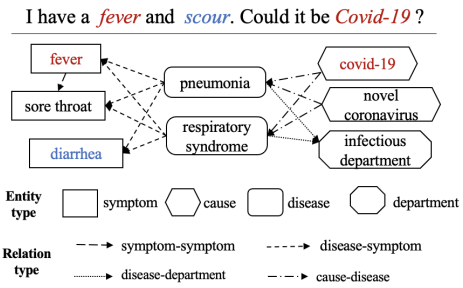

Figure 2: An example of domain knowledge localization in the healthcare domain. The words in red indicate that the entities and the mentions are *exactly* same, and the words in blue indicate that the mention (scour) and the entity (diarrhea) share *similar word embeddings*.

domain-specific data augmentation. Here, we propose a dual-view strategy to conduct the *relevant entity retrieval* by considering both KG and the training text data.

View 1: KG-based Entity Retrieval (KGER)

A direct strategy for domain KG-based DA is to replace the entity with another same-category entity, e.g., replacing 'William' with 'Mike' as both are person names (Yue and Zhou, 2020). However, it may suffer from two pitfalls: (i) Although the original and replaced entities are in the same category, they can be totally different, such as *pneumonia* and *fracture* (both are diseases), which may negatively impact the downstream tasks, e.g., classifying a medical transcription to the relevant department[2]; (ii) When two or more entities appear in a text, they may have certain valuable relationships (e.g., disease and symptom), but replacing these entities separately would ignore this information.

To address the above issues, we propose two principles for **KGER**: (i) *entity relevance*, refers to ensuring that the retrieved entity is similar to the original entity, not just with the same category; (ii) *relation consistency*, means keeping the relationships unchanged between multiple replaced entities in one text. We formulate a domain KG as $\mathcal{G} = \{E, R, T, C\}$, where E, R, T, and C are the sets of entities, relations, triples, and entities' categories, respectively. Specifically, $T = T^R \bigcup T^C$, where $T^R = \{(h, r, t)|h, t \in E, r \in R\}$ and $T^C = \{(e, BelongTo, c)|e \in E, c \in C\}$.

Given an entity e, we can get its category c, involved triples $T_e = \{(e, r, t) \in T^R\} \bigcup \{(h, r, e) \in T^R\}$, and the adjacent entities $E_e = \{e'|(e, r, e') \in T_e, (e, r, e') \in T_e\}$. To obtain more same-category entities, we further retrieve the involved triples of E_e, named T_{e2}, and put T_e and T_{e2} together as the candidate triples (i.e., 2-hop triples around e).

That is: (1) If only one entity exists, or multiple entities exist but do not have direct KG relations in the text, we randomly select a same-category entity e' from the candidate triples to replace each original entity e. Note that e' must be within 2-hop around e, ensuring the *entity relevance*. (2) If there exist certain pairs of entities with KG relations, we would seek the same relation-type triple from the candidate triples for replacing the pair of entities together, following the *relation consistency*.

For instance, '*I have a **fever** and **scour**. Could it be **pneumonia**?*' (shown in Figure 3), *fever* and *scour* are the symptoms of *pneumonia*. So we need to search for suitable triples to satisfy relation consistency. For instance, *diarrhea* and *sore throat* are the symptoms of *respiratory syndrome*, so the augmented text could be '*I have a **diarrhea** and **sore throat**. Could it be **respiratory syndrome**?*'.

[2] https://www.kaggle.com/datasets/tboyle10/medicaltranscriptions

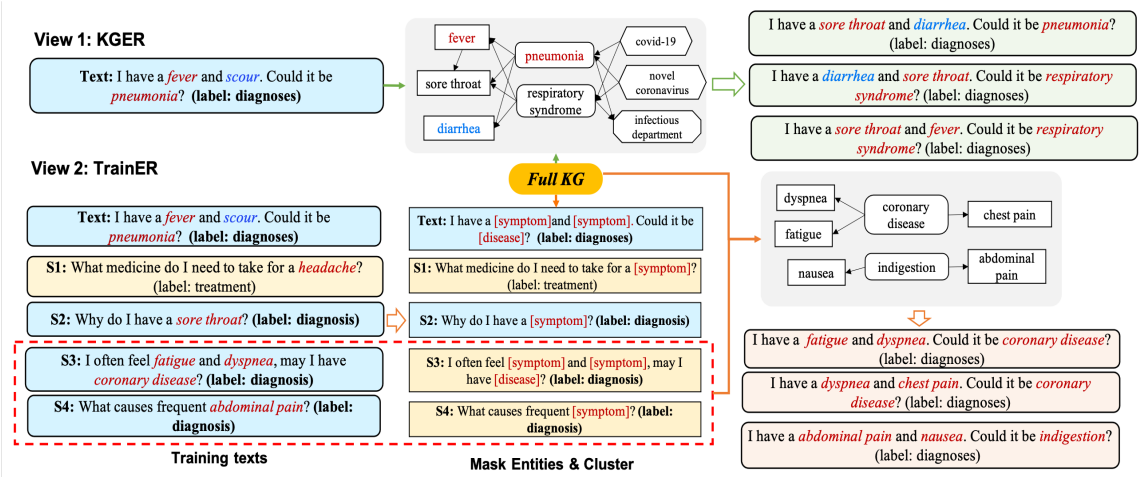

Figure 3: Dual-view knowledge augmentation with a healthcare text as an example. **KGER**: 1. retrieve 2-hop relevant entities in the full KG; 2. replace entities following two principles, 'entity relevance' and 'relation consistency'. **TrainER**: 1. mask entity with the category to represent the expressions pattern; 2. text clustering to select samples with the same label but different clusters (i.e., S3 & S4); 3. collect the candidate triples from selected samples; 4. augment data by replacing relevant entities.

View 2: Training Data-based Entity Retrieval (TrainER)

In View 1, we mainly retrieve relevant entities that are close in the KG. However, entity pairs far away in the KG may be helpful for the specific task if being replaced with each other. For example, for the task to detect the medical query intent, '*blood routine examination*' and '*CT*' is the entity pair that could be replaced with each other for augmentation because they most probably appear in the queries about *diagnosis* and *cause analysis*, but they are distant from each other in the medical KG, like CMedicalKG [3].

To find such task-specific valuable replacement entity pairs which may not be near in the KG, we design a new View 2, *Training Data-based Entity Retrieval (TrainER)*, to retrieve task-specific entity pairs from training data. REINA (Wang et al., 2022a) has verified that retrieving from training data to enrich model inputs (concatenating the original input and retrieved training data) may generate significant gains. Inspired by this idea, TrainER aims to extract gainful entity pairs from the training data for augmentation.

In general, a good entity pair for replacement may satisfy at least two properties: (i) *label consistency*, indicates that the two entities in the pair should be contained in two training texts with the same task label; (ii) *expression diversity*, means that the two texts containing the two entities should have different expression patterns, so as to enrich the training data diversity. Specifically, to reach *label consistency*, for an entity e in a text t, we would retrieve a same-category entity e' from another text t' if t and t' have the same label. To achieve *expression diversity*, we first cluster all the training texts into different clusters with diverse expression patterns. Then, for an entity e in a text t, the replaced entity e' will be retrieved from t' only if t and t' are not in the same cluster. Figure 3 elaborates on the process of TrainER.

To conduct training data clustering to differentiate expression patterns, we first mask entities with their categories to extract the expression templates for each training text. For instance, '*what is* ***pneumonia****?*' and '*what is* ***fracture***' share the same expression template '*what is* ***[disease]***?', as both sentences have the same pattern regardless of the specific entity (i.e., disease). Then, we run a clustering algorithm on the masked texts, i.e., expression templates, to identify diverse expression patterns. The K-means clustering (Arthur and Vassilvitskii, 2007) is applied due to its high efficiency and effectiveness in empirical experiments; the feature of a masked text is represented by TF-IDF vectorization (Jones, 2004). Same as the *relation consistency* principle in KGER, if there are certain entity pair with KG relations in the original text, we will retrieve the entity pair with the same relation from other training texts.

3.4 Module 3: Augmentation Quality Assessment

After Module 1 & 2, we obtain a set of augmented texts. A straightforward way is to fine-tune task-specific models with these texts like most prior studies (Zhang et al., 2015; Wei and Zou, 2019).

[3] https://github.com/liuhuanyong/QASystemOnMedicalKG

Recent work (Zhou et al., 2022) has found that not all the augmented texts are equivalently effective; thus, selecting high-quality ones may further improve the model performance.

Inspired by this finding, **KnowledgeDA** includes a quality assessment module to justify the quality of each augmented text. Prior work (Anaby-Tavor et al., 2020; Zhou et al., 2022) uses the prediction confidence as the quality metric and selects top-K high-confidence augmented samples for fine-tuning, because this ensures the label correctness of augmented texts. However, we argue that it may not significantly improve the model performance since high confidence means that the pattern inside the augmented sample has already been encoded in the original model (without augmentation).

Hence, we first fine-tune PLM (e.g., BERT) on the task texts; then use this plain fine-tuned model $\mathcal{M}$ to predict the augmented texts and obtain the confidence scores. Instead of selecting top-K confident samples, we pick K augmented samples whose confidence is close to a predefined threshold δ. Note that δ should not be a too small number, as we still want to ensure the correctness of the training labels for augmented texts; meanwhile, δ should not be too large, as a very high-confident sample would contribute little new knowledge to the model. Based on this idea, we design a novel confidence-based data filtering strategy to retain gainful augmented samples.

The task data $D = \{(x_i, y_i)\}_{i=1}^{n}$ and the plain fine-tuned model $\mathcal{M}$ (without augmentation) are known, where x_i is a string of text, and the label $y_i \in \{1, 2, \cdots, q\}$ is the label of x_i among a set of q labels. Through *KGER* and *TrainER*, we can generate the augmented samples $D_i^{aug} = \{x_i^1, x_i^2, \cdots, x_i^m\}$ for the i-th sample, x_i. The prediction confidence (probability) of D_i^{aug} can be calculated as $P_i^{aug} = \{p_i^j\}_{j=1}^{m}$, where $p_i^j = prob(\mathcal{M}(x_i^j) = y_i)$.

We propose a confidence threshold δ to adjust sample selection criteria. Given δ, the sampling weights of D_i^{aug} can be calculated by

$$w_i^1, w_i^2, \cdots, w_i^m = softmax(\xi_i^1, \xi_i^2, \cdots, \xi_i^m) \quad (1)$$

where $\xi_i^j = 1 - |\delta - p_i^j|$. If p_i^j is closer to δ (0.75 in our implementation), we have a higher probability to select this sample. With this confidence-based sampling strategy, we can select augmented samples to further fine-tune the task model $\mathcal{M}$. In general, the selected samples would be relatively confident but not too highly-confident, thus ensuring both *label correctness* and *new knowledge*.

Dataset	Lang.	#Labels	#Samples	#Mentions
CMID	CHI	4	12254	5182
KUAKE-QIC	CHI	11	8886	3369
TRANS	ENG	7	1740	2298
ABS	ENG	5	14438	3808

Table 1: Dataset Statistics

4 Empirical Evaluation

4.1 Text Classification

4.1.1 Setup

Datasets. We conduct experiments on four datasets in healthcare: CMID[4] and KUAKE-QIC (Zhang et al., 2022a)are in Chinese; TRANS[5] and ABS[6] are in English. The basic information is enumerated in Table 1. For Chinese, we use an open-source medical KG, CMedicalKG[7]; for English, we adopt the Unified Medical Language System (UMLS) (Bodenreider, 2004) as KG. The preprocessing of KGs can be found in Appendix A.3.

Baselines. As **KnowledgeDA** focuses on explicit knowledge injection during DA by domain entity replacement, we mainly compare with state-of-the-art rule-based DA methods: **SR** (Vijayaraghavan et al., 2016) uses token-level replacement with synonyms; **EDA** (Wei and Zou, 2019) uses token-level random perturbation operations including random insertion, deletion, and swap; **PHICON** (Yue and Zhou, 2020) uses entity-level replacement with other entities belonging to the same category. For each DA method, we scale up the training data to 5 times the original size and select the best model on the validation set for evaluation. All the methods are based on the same text classifier with the same hyper-parameters (in Appendix A.1). In the main experiments, the base classifier is BERT-base (hereinafter referred to as BERT). We also experiment with domain-specific PLMs as stronger classifiers, discussed in the later part. And the experiment in the software development domain is shown in Appendix C.

4.1.2 Results in Healthcare

Main Results. Table 2 shows the results of different DA methods. We can observe

[4] https://github.com/ishine/CMID
[5] https://www.kaggle.com/datasets/tboyle10/medicaltranscriptions
[6] https://github.com/PoojaR24/Medical-Text-Classification
[7] https://github.com/liuhuanyong/QASystemOnMedicalKG

DA Method	CMID (Chinese)		KUAKE-QIC (Chinese)		TRANS (English)		ABS (English)	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
None	70.25(0.80)	68.21(0.90)	78.82(0.81)	78.57(0.72)	73.10(1.79)	71.50(1.77)	63.95(0.31)	62.84(0.40)
SR	71.90(0.76)	70.97(0.39)	80.52(0.73)	80.10(0.80)	72.38(0.24)	72.52(0.30)	64.14(0.25)	63.13(0.30)
EDA	70.59(0.65)	70.05(1.25)	79.45(0.33)	79.01(0.41)	73.91(0.26)	73.71(0.28)	63.23(0.65)	62.09(0.72)
PHICON	71.95(0.35)	71.14(0.53)	80.52(0.74)	80.23(0.82)	74.53(1.19)	73.10(0.77)	64.17(0.59)	63.35(0.60)
KnowledgeDA	**72.38(0.46)***	**71.94(0.38)**	**81.67(0.41)*****	**81.31(0.44)***	**75.66(0.58)****	**75.37(0.72)**	**64.97(0.29)**	**64.18(0.28)***

Table 2: Performance of baselines and **KnowledgeDA** with BERT in healthcare: 1. Values in '( )' denote the standard deviation of five repeated experiments' results; 2. **Bold** denotes the best-performed ones of the task; 3. *, **, *** denote that the t-test significance p-value $< 0.1, 0.05, 0.01$ when comparing the results of **KnowledgeDA** and the best baseline.

Method	CMID	KUAKE-QIC	TRANS	ABS
SR	14.80%	17.89%	33.56%	24.33%
EDA	11.18%	15.85%	26.66%	25.07%
PHICON	35.84%	29.11%	76.45%	74.07%
KnowledgeDA	**40.67%**	**35.36%**	**79.33%**	**78.37%**

Table 3: Novel entity coverage for healthcare datasets

Method	CMID		TRANS	
	Acc.	F1	Acc.	F1
None	73.10(0.32)	71.71(0.37)	75.88(0.41)	75.22(0.57)
SR	73.49(0.19)	72.08(0.22)	75.38(1.11)	75.01(1.43)
EDA	72.93(0.46)	71.89(0.89)	75.40(0.79)	74.91(0.80)
PHICON	73.51(0.68)	72.49(0.63)	75.37(0.59)	75.18(0.72)
KnowledgeDA	**73.60(0.33)**	**72.61(0.31)**	**76.52(0.59)****	**76.54(0.91)***

Table 4: Performance with domain-specific PLMs, where CMID/TRANS use eHealth/ClinicalBERT as the domain-specific PLM.

that **KnowledgeDA** achieves the best performance among all the methods in both accuracy and F1 score on four datasets. At the same time, **PHICON** also outperforms **SR** and **EDA** in most cases, verifying the effectiveness of domain-specific knowledge. Specifically, on two Chinese datasets, CMID and KUAKE-QIC, **KnowledgeDA** improves the accuracy by 3.03% and 3.62%, respectively, over the fine-tuned model without augmentation. Moreover, compared to the best baseline, **PHICON**, **KnowledgeDA**'s improvements on accuracy are still statistically significant. Similar results are also observed on two English datasets. In a nutshell, the results suggest that domain-specific entity replacement can facilitate text classification in healthcare. Compared to **PHICON** which only considers entity categories, **KnowledgeDA** selects entities from dual views and accounts for the KG relations between them, which further improves the quality of the augmented text and thus achieves a better performance. To further quantitatively verify that **KnowledgeDA** can introduce more domain knowledge, following Wang et al. (2022b), we calculate *Novel Entity Coverage*, the percentage of the *novel* entities in the test data covered by augmented texts (*novel* means

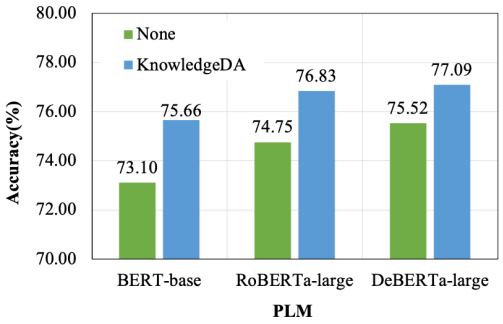

Figure 4: Performances on TRANS with larger PLMs.

not appearing in the training data). As illustrated in Table 3, **KnowledgeDA** has the highest coverage, which also explains the effectiveness.

Domain-specific PLMs as the Base Classifiers. Domain-specific PLMs contain domain knowledge by pre-training with domain corpus. To confirm that **KnowledgeDA** is still beneficial with a domain-specific PLM, we use eHealth (Wang et al., 2021) and ClinicalBERT (Alsentzer et al., 2019) as the PLMs for Chinese and English datasets, respectively. According to Table 4, the improvement brought by the domain-specific PLMs is evident (comparing with the results of BERT in Table 2). Consistent with the survey (Feng et al., 2021), we discover that when using the domain-specific PLMs, baseline DA methods may not generate an obvious performance improvement and even have a negative effect compared to no-augmentation. For instance, on TRANS, **EDA** improves the performance over BERT (increasing F1 score from 71.50 to 73.71); while **EDA** worsens the performance when a domain-specific PLM is used (reducing F1 score from 75.22 to 74.91). However, even with domain-specific PLMs, **KnowledgeDA** can still improve the domain NLP task performance consistently. Note that for TRANS, **KnowledgeDA** is the only DA method with positive improvement (and this improvement is also statistically significant).

Larger PLMs as the Base Classifiers. In addi-

Method	CMID		TRANS	
	Acc.	F1	Acc.	F1
KnowledgeDA	**72.38(0.46)**	**71.94(0.38)**	**75.66(0.58)**	**75.37(0.72)**
w.o. SimMatch	71.75(0.38)	71.26(0.41)	74.05(1.11)	74.23(0.78)
w.o. KGER	71.82(0.49)	71.59(0.35)	73.92(0.66)	74.67(0.85)
w.o. TrainER	71.88(0.57)	71.48(0.34)	74.36(0.63)	74.74(0.65)
w.o. Assess	72.00(0.61)	70.84(0.56)	74.78(0.60)	74.65(0.73)

Table 5: Effectiveness of each module in **KnowledgeDA**

Time(min)	None	SR	EDA	PHICON	KnowledgeDA
CMID	5.05	22.23	20.63	30.77	36.33
TRANS	5.43	9.54	8.22	18.46	25.82

Table 6: Time Consumption of DA & Model Fine-tuning.

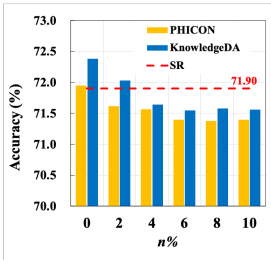

Figure 5: Performance on CMID under n% CMedicalKG disturbance.

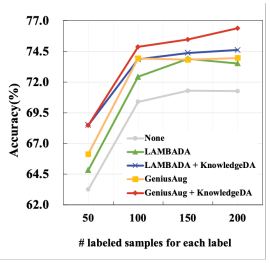

Figure 6: Combining with LAMBADA and GeniusAug (TRANS).

tion to using domain-specific PLMs with the same parameter size as BERT-base (110 million parameters), we further take RoBERTa-large (Liu et al., 2019a) and DeBERTa-large (He et al., 2021) with more than 350 million parameters as base classifiers on the TRANS dataset. As shown in Figure 4, with increasing parameters, RoBERTa-large and DeBERTa-large achieve better accuracy than BERT without DA. However, there are still notable improvements of 2.78% and 2.08% in accuracy with **KnowledgeDA** on RoBERTa-large and DeBERTa-large, demonstrating the generalizability of **KnowledgeDA**.

Ablation Study. To validate the effectiveness of each module of **KnowledgeDA**, we design corresponding ablation experiments: *KnowledgeDA w.o. SimMatch* removes the similarity-based non-exact matching and only uses exact string matching in Module 1 (Sec. 3.2); *KnowledgeDA w.o. KGER* removes the KGER (view 1) in Module 2 (Sec. 3.3); *KnowledgeDA w.o. TrainER* removes the TrainER (view 2) in Module 2 (Sec. 3.3); *KnowledgeDA w.o. Assess* removes the quality assessment module, i.e., Module 3 (Sec. 3.4). Table 5 shows the results. **KnowledgeDA** outperforms all the other methods that remove certain components. This verifies the validity of each module of **KnowledgeDA**.

Time Consumption. Table 6 reports the time consumption of all DA methods on CMID and TRANS. The time required for fine-tuning without augmentation is short ($\sim$ 5 minutes). As **PHICON** and **KnowledgeDA** need to retrieve the entity mentions and then replace them, time consumption is increased.In particular, **KnowledgeDA** takes more time because it considers the relations between entities in the KG. In general, the learning process can be completed in about half an hour for **KnowledgeDA** (also not much longer than competitive baselines like **PHICON**).

Impact of KG Errors. Considering that the KG quality may affect the quality of the augmented texts (Kang et al., 2022), we randomly change the categories of $n\%$ entities and the relation types of $n\%$ triples in CMedicalKG, and test the performance of **KnowledgeDA** in CMID dataset. As shown in Figure 5, when we adjust n from 0 to 10, the accuracy is between 71.0 and 72.5, with a slight decline. When $n \geq 4$, **SR**, the KG-independent DA method, performs better. This illustrates the importance to ensure the KG quality for KG-based DA methods, which is consistent with the findings of other KG-based applications (Hu et al., 2022). In the future, we will explore how to identify potential KG errors so as to improve the robustness of **KnowledgeDA**.

Different Strategies for Augmented Data Quality Assessment and Selection. We compare two strategies for augmented data quality assessment and selection: δ-*K* is proposed in Sec. 3.4; *Top-K* (Anaby-Tavor et al., 2020; Zhou et al., 2022) selects the top K augmented samples with the highest confidence for each original sample. Table 7 shows the results of different strategies, as well as the results of *KnowledgeDA without quality assessment*. δ-*K* and *Top-K* both outperform *KnowledgeDA without assessment*, verifying the necessity to select high-quality samples for augmentation. And δ-*K* performs better than *Top-K*. This empirically validates our intuition that an augmented sample with a not-too-high confidence may bring more new knowledge to the NLP model, as discussed in Sec. 3.4.

Combine with Generator-based Augmentation Techniques. **KnowledgeDA** provides a unified framework for domain-specific knowledge augmentation, which may be combined with other DA tech-

Method	CMID		TRANS	
	Acc.	F1	Acc.	F1
w.o. Assess	72.00(0.61)	70.84(0.56)	74.78(0.60)	74.65(0.73)
Top-K	71.95(0.52)	71.53(0.43)	75.01(0.81)	74.89(0.69)
δ-K	**72.38(0.46)**	**71.94(0.38)**	**75.66(0.58)**	**75.37(0.72)**

Table 7: Different quality assessment strategies in **KnowledgeDA**

niques. Here, we use generator-based augmentation methods as an example. Specifically, we generate augmented samples with two methods, LAMBADA (Anaby-Tavor et al., 2020) and GeniusAug (Guo et al., 2022); based on these augmented samples, we leverage **KnowledgeDA** to acquire more augmented samples. Since generator-based methods are mostly applied to few-shot tasks (Anaby-Tavor et al., 2020), we randomly select 50 to 200 samples for each task label in the TRANS dataset. LAMBADA and GeniusAug both generate 200 more samples for each label. Figure 6 shows the results. As expected, the performance of each method goes up as the number of labeled samples increases. More importantly, combining **KnowledgeDA** with LAMBADA or GeniusAug both can achieve higher accuracy. This demonstrates the general utility of **KnowledgeDA** to combine with generator-based DA methods to improve the few-shot NLP tasks.

Compare with GPT-3.5. Recently, ChatGPT has shown powerful text generation capabilities. To explore the performance of this large language model on domain-specific tasks, we use the OpenAI API [8] to query text-davinci-003 (the most powerful GPT-3.5) by the prompt, *'decide which label the following text belongs to, {label names}: \n Text:{sentence} \n Label: '*. It can be seen as a zero-shot manner to response directly. For TRANS(English), the test accuracy is 66.67% ($\sim$ 10% lower than **KnowledgeDA** with BERT). It performs even worse on CMID(Chinese) with an accuracy of only 32.32%, perhaps due to the limited exposure to relevant texts and knowledge. Therefore, more effective prompt engineering or fine-tuning of GPT (especially for non-English languages) is still necessary for domain-specific tasks, which may be potential future work.

4.2 QA Tasks

Setup. The **CMedQA**(Chinese) (Zhang et al., 2017) and **PubMedQA**(English) (Jin et al., 2019) are used for the QA task. Both datasets give the label of each question-answer pair (i.e., match or mismatch). For CMedQA, we sample 1000 question-answer pairs from the original dataset. For PubMedQA, we keep the original data size (429 samples). In **KnowledgeDA**, we take the question and answer pair as input and retrieve the entity mentions together. While fine-tuning, we feed questions and answers, separated by the special [SEP] token to BERT (Jin et al., 2019). The KGs and other settings are the same as classification tasks.

[8] https://platform.openai.com

Method	**CMedQA**(Chinese)		**PubMedQA**(English)	
	Acc.	F1	Acc.	F1
None	85.00(3.96)	82.60(7.06)	66.00(6.87)	57.65(10.46)
SR	88.46(0.84)	87.91(0.73)	72.68(1.97)	68.99(1.47)
EDA	88.66(1.18)	88.37(1.00)	72.72(1.57)	68.69(1.71)
PHICON	88.56(1.17)	87.83(1.39)	73.96(1.88)	69.67(1.98)
KnowledgeDA	**89.16(0.58)**	**88.58(0.56)**	**74.64(0.83)***	**70.98(0.70)**

Table 8: Performance of QA Tasks.

Results. Table 8 compares the performance of different DA methods based on BERT. It is obvious that using any data augmentation strategy can make the performance more stable under different seeds (i.e. smaller standard deviation). Also, **KnowledgeDA** outperforms all the baselines.

5 Conclusions

In this paper, we present **KnowledgeDA**, a unified knowledge graph service to boost domain-specific NLP tasks. The intrinsic technical novelty is a three-step framework of task-specific data augmentation process based on domain KGs. The experiments on healthcare-related texts both in English and Chinese verify the effectiveness and generality of **KnowledgeDA**. We also confirm that it can be flexible and effective to incorporate other generator-based DA methods on few-shot tasks. In the future, we can further investigate how to better combine **KnowledgeDA** and generator-based DA methods and add KG quality inspection methods to avoid the negative impact of errors in KG.

Limitations

Domain KGs are the premise of **KnowledgeDA**, while open and high-quality domain KGs may be rare in some domains. Therefore, the method will be limited in the domains without suitable KGs. Besides, we use a similarity-based method to map entity mentions in the text to the corresponding entities in the KG. Although this method performs efficiently, it ignores the problem of entity ambigu-

ity (Vretinaris et al., 2021). For instance, the abbreviation, CAT, can stand for '*catalase*' or '*COPD Assessment Test*' in healthcare. To address this problem, it is necessary to use contextual information to clarify the specific meaning of the mention (Phan et al., 2017; Orr et al., 2021; Vretinaris et al., 2021). Last but not least, **KnowledgeDA** may be not good at tasks of paragraph-level texts and the efficiency will reduce. Because long texts probably contain more entity mentions and have more complex syntax, it is more difficult to retrieve the entities and acquire their relations from the KG.

Ethics Statement

This paper proposes a unified framework, **KnowledgeDA**, for text augmentation based on domain KGs for domain-specific NLP tasks. All the experimental datasets and KGs are publicly available, and the related papers and links have been listed in the paper. Also, though PLM and KG are publicly available, there may still be several ethical considerations to bear in mind when applying **KnowledgeDA** in real-world scenarios. For instance, it is crucial to check whether KG contains biases concerning race, gender, and other demographic attributes.

Acknowledgements

This research was supported by National Key R&D Program of China (2020AAA0109401) and NSFC Grants no. 61972008, 72071125, and 72031001.

References

Bilal Abu-Salih. 2021. Domain-specific knowledge graphs: A survey. *Journal of Network and Computer Applications*, 185:103076.

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

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *AAAI*, pages 7383–7390.

David Arthur and Sergei Vassilvitskii. 2007. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1027–1035. SIAM.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: Pretrained language model for scientific text. In *EMNLP*.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Res.*, 32(Database-Issue):267–270.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Razvan C. Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *EACL*.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *ACL*, pages 2147–2157.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Shangbin Feng, Zhaoxuan Tan, Wenqian Zhang, Zhenyu Lei, and Yulia Tsvetkov. 2022. Kalm: Knowledge-aware integration of local, document, and global contexts for long document understanding. *arXiv preprint arXiv:2210.04105*.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of ACL*, pages 968–988.

Paolo Ferragina and Ugo Scaiella. 2010. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM*, pages 1625–1628. ACM.

Radu Florian, Hany Hassan, Abraham Ittycheriah, Hongyan Jing, Nanda Kambhatla, Xiaoqiang Luo, Nicolas Nicolov, and Salim Roukos. 2004. A statistical model for multilingual entity detection and tracking. In *NAACL-HLT*, pages 1–8.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Biyang Guo, Yeyun Gong, Yelong Shen, Songqiao Han, Hailiang Huang, Nan Duan, and Weizhu Chen. 2022. Genius: Sketch-based language model pre-training via extreme and selective masking for text generation and augmentation.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding enhanced bert with disentangled attention. In *ICLR*.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *ACL*, pages 2225–2240.

Shengze Hu, Zhen Tan, Weixin Zeng, Bin Ge, and Weidong Xiao. 2019. Entity linking via symmetrical attention-based neural network and entity structural features. *Symmetry*, 11(4):453.

Ningyuan Huang, Yash R Deshpande, Yibo Liu, Houda Alberts, Kyunghyun Cho, Clara Vania, and Iacer Calixto. 2022. Endowing language models with multimodal knowledge graph representations. *arXiv preprint arXiv:2206.13163*.

Yan Jia, Yulu Qi, Huaijun Shang, Rong Jiang, and Aiping Li. 2018. A practical approach to constructing a knowledge graph for cybersecurity. *Engineering*, 4(1):53–60.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP-IJCNLP*, pages 2567–2577.

Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation*, 60(5):493–502.

Minki Kang, Jinheon Baek, and Sung Ju Hwang. 2022. KALA: knowledge-augmented language model adaptation. In *NAACL-HLT*, pages 5144–5167.

Ilknur Karadeniz and Arzucan Özgür. 2019. Linking entities through an ontology using word embeddings and syntactic re-ranking. *BMC bioinformatics*, 20:1–12.

Bum Chul Kwon, Min-Je Choi, Joanne Taery Kim, Edward Choi, Young Bin Kim, Soonwook Kwon, Jimeng Sun, and Jaegul Choo. 2019. Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):299–309.

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.

Junzhuo Li and Deyi Xiong. 2022. KaFSP: Knowledge-aware fuzzy semantic parsing for conversational question answering over a large-scale knowledge base. In *ACL*, pages 461–473.

Linfeng Li, Peng Wang, Jun Yan, Yao Wang, Simin Li, Jinpeng Jiang, Zhe Sun, Buzhou Tang, Tsung-Hui Chang, Shenghui Wang, et al. 2020a. Real-world data medical knowledge graph: construction and applications. *Artificial intelligence in medicine*, 103:101817.

Ying Li, Vitalii Zakhozhyi, Daniel Zhu, and Luis J Salazar. 2020b. Domain specific knowledge graphs as a service to the public: Powering social-impact funding in the us. In *SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2793–2801.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and S Yu Philip. 2021. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6418–6425.

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

Zhongxin Liu, Xin Xia, Christoph Treude, David Lo, and Shanping Li. 2019b. Automatic generation of pull request descriptions. In *ASE*, pages 176–188.

Shayne Longpre, Yu Wang, and Chris DuBois. 2020. How effective is task-agnostic data augmentation for pretrained transformers? In *EMNLP-Findings*, pages 4401–4411.

George Michalopoulos, Yuanxin Wang, Hussam Kaka, Helen Chen, and Alexander Wong. 2021. Umls-BERT: Clinical domain knowledge augmentation of contextual embeddings using the Unified Medical Language System Metathesaurus. In *NAACL-HLT*, pages 1744–1753.

Laurel J. Orr, Megan Leszczynski, Neel Guha, Sen Wu, Simran Arora, Xiao Ling, and Christopher Ré. 2021. Bootleg: Chasing the tail with self-supervised named entity disambiguation. In *CIDR*.

Minh C. Phan, Aixin Sun, Yi Tay, Jialong Han, and Chenliang Li. 2017. Neupl: Attention-based semantic matching and pair-linking for entity disambiguation. In *CIKM*, pages 1667–1676. ACM.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. Exploring the limits of transfer learning with a unified text-to-text transformer. 21(1).

Shijing Si, Rui Wang, Jedrek Wosik, Hao Zhang, David Dov, Guoyin Wang, and Lawrence Carin. 2020. Students need more attention: Bert-based attention model for small data with application to automatic patient message triage. In *Proceedings of the 5th Machine Learning for Healthcare Conference*, volume 126 of *Proceedings of Machine Learning Research*, pages 436–456.

Prabhnoor Singh, Rajkanwar Chopra, Ojasvi Sharma, and Rekha Singla. 2020. Stackoverflow tag prediction using tag associations and code analysis. *Journal of Discrete Mathematical Sciences and Cryptography*, 23(1):35–43.

Miroslav Tushev, Fahimeh Ebrahimi, and Anas Mahmoud. 2022. Domain-specific analysis of mobile app reviews using keyword-assisted topic models. In *ICSE*, pages 762–773.

Prashanth Vijayaraghavan, Ivan Sysoev, Soroush Vosoughi, and Deb Roy. 2016. DeepStance at SemEval-2016 task 6: Detecting stance in tweets using character and word-level CNNs. In *SemEval*, pages 413–419.

Alina Vretinaris, Chuan Lei, Vasilis Efthymiou, Xiao Qin, and Fatma Özcan. 2021. Medical entity disambiguation using graph neural networks. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, page 2310–2318. Association for Computing Machinery.

Quan Wang, Songtai Dai, Benfeng Xu, Yajuan Lyu, Yong Zhu, Hua Wu, and Haifeng Wang. 2021. Building chinese biomedical language models via multi-level text discrimination. *CoRR*, abs/2110.07244.

Shuohang Wang, Yichong Xu, Yuwei Fang, Yang Liu, Siqi Sun, Ruochen Xu, Chenguang Zhu, and Michael Zeng. 2022a. Training data is more valuable than you think: A simple and effective method by retrieving from training data. In *ACL*, pages 3170–3179.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. Switchout: an efficient data augmentation algorithm for neural machine translation. In *EMNLP*, pages 856–861.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022b. Promda: Prompt-based data augmentation for low-resource NLU tasks. In *ACL*, pages 4242–4255.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP*, pages 6382–6388.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *EMNLP*, pages 6397–6407.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D Manning, Percy S Liang, and Jure Leskovec. 2022. Deep bidirectional language-knowledge graph pretraining. *Advances in Neural Information Processing Systems*, 35:37309–37323.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woo-Myoung Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. In *EMNLP*, pages 2225–2239.

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022. KG-FiD: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In *ACL*, pages 4961–4974.

Xiang Yue, Bernal Jimenez Gutierrez, and Huan Sun. 2020. Clinical reading comprehension: A thorough analysis of the emrqa dataset. In *ACL*, pages 4474–4486.

Xiang Yue and Shuang Zhou. 2020. PHICON: Improving generalization of clinical text de-identification models via data augmentation. In *ClinicalNLP-EMNLP Workshop*, pages 209–214.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *ICLR*. OpenReview.net.

Ningyu Zhang, Mosha Chen, Zhen Bi, Xiaozhuan Liang, Lei Li, Xin Shang, Kangping Yin, Chuanqi Tan, Jian Xu, Fei Huang, Luo Si, Yuan Ni, Guotong Xie, Zhifang Sui, Baobao Chang, Hui Zong, Zheng Yuan, Linfeng Li, Jun Yan, Hongying Zan, Kunli Zhang, Buzhou Tang, and Qingcai Chen. 2022a. CBLUE: A chinese biomedical language understanding evaluation benchmark. In *ACL*, pages 7888–7915.

Sheng Zhang, Xin Zhang, Hui Wang, Jiajun Cheng, Pei Li, and Zhaoyun Ding. 2017. Chinese medical question answer matching using end-to-end character-level multi-scale cnns. *Applied Sciences*, 7(8):767.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022b. Greaselm: Graph reasoning enhanced language models. In *ICLR*.

Yong Zhang, Ming Sheng, Rui Zhou, Ye Wang, Guangjie Han, Han Zhang, Chunxiao Xing, and Jing

Dong. 2020. Hkgb: an inclusive, extensible, intelligent, semi-auto-constructed knowledge graph framework for healthcare with clinicians' expertise incorporated. *Information Processing & Management*, 57(6):102324.

Jing Zhou, Yanan Zheng, Jie Tang, Li Jian, and Zhilin Yang. 2022. Flipda: Effective and robust data augmentation for few-shot learning. In *ACL*, pages 8646–8665.

Chenguang Zhu, Yichong Xu, Xiang Ren, Bill Yuchen Lin, Meng Jiang, and Wenhao Yu. 2022. Knowledge-augmented methods for natural language processing. In *ACL*, pages 12–20.

A Implementation Details

A.1 Experiment Platform & Settings

Our experiment platform is a server with AMD Ryzen 9 3900X 12-Core Processor, 64 GB RAM and GeForce RTX 3090. We use Python 3.6 with pytorch 1.8 on Ubuntu 20.04 for algorithm implementation.

For the text classification task, we feed the [CLS] representation into the output layer when BERT-base as the classifier (Devlin et al., 2019). We split the dataset into training set, validation set, and test set as 8:1:1. When fine-tuning PLMs, we set batch size to 32, learning rate to 1e-5, and training epoch to 10. It will early stop if the loss of the validation set does not decrease in 500 iterations. *Accuracy* and *micro-F1* are used as the metrics in text classification and QA tasks. We repeat each experiment 5 times and record the average results.

A.2 Algorithms

In this part, we summarize the detailed implementations of *domain knowledge localization* (i.e., Module 1) and *domain knowledge augmentation & augmentation quality assessment* in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1: Domain Knowledge Localization

Input: A text x, the entities list E, words embeddings dictionary $Embeds$, and similarity threshold θ
Output: A matched pair list $Matchs$ of mentions in x and entities in E

Initialize $Matchs$ as an empty list ;
Preprocess x with NLP preprocessing pipeline to get word list $words$;
Construct entity embedding matrix E_{emb} and embedding matrix W_{emb} by searching for E and $words$ from $Embeds$;
Compute similarity matrix $Sim = W_{emb} \times E_{emb}.T$;
Query the maximum similarity sim_values between each word and entity;
if $sim\_value \geqslant \theta$ **then**
    Find the index of sim_values in Sim ;
    Get the pair $(word, entity)$ according to index ;
    Add $(word, entity)$ to $Matchs$;
end
Return $Matchs$;

A.3 KGs Preprocessing

Healthcare is a field with rich professional knowledge. There are also publicly available knowledge graphs, e.g., the Unified Medical Language System (UMLS) (Bodenreider, 2004). We take such open medical KGs for healthcare

Algorithm 2: Domain Knowledge Augmentation & Augmentation Quality Assessment

Input: Train Data $D = \{(x_i, y_i)_{i=1}^n\}$; the KG $\mathcal{G} = \{E, R, T, C\}$; the pre-trained language model PLM; a confidence threshold δ.
Output: The selected augmented samples D^{aug}

Fine-tune without augmentation $\mathcal{M}$ = *fine-tune*(PLM, D) ;
for x_i *in* $\{x_i\}_{i=1}^n$ **do**
    Get the $Matches_i$ in x_i by Algorithm 1;
    Generate augmented samples D_i^{aug} with $\mathcal{G}$ following the steps in Figure 3;
    Initialize the prediction probabilities of D_i^{aug} as P_i^{aug};
    for x_i^j *in* D_i^{aug} **do**
        Calculate the prediction probability $p_i^j = prob(\mathcal{M}(x_i^j) = y_i)$;
        Add p_i^j to P_i^{aug};
    end
    Calculate the sampling weights of D_i^{aug} according to Eq. 1 ;
    Sample 5 samples from D_i^{aug} by weights and add them to D^{aug};
end
return D^{aug};

KnowledgeDA. UMLS Metathesaurus is a compendium of many biomedical terminologies with the associated information, including synonyms, categories, and relationships. It groups semantically equivalent or similar words into the same concept, for example, the words 'flu', 'syndrome flu' and 'influenza' are mapped to the same concept unique identifier (CUI) *C0021400*, which belongs to the category, *disease or syndrome*. There are 127 semantic types in biology, chemistry, and medicine, consisting of 4,441,326 CUIs (16,132,273 terminologies) in the UMLS 2021AA version. Since the size of the KG is too large to affect the speed of retrieval, we only screen out entities that belong to the type of medicine (e.g., *body part, organ, or organ component*, *disease or syndrome*, etc.). At the same time, we also delete non-English strings. Finally, we keep 1,145,062 CUIs (16 semantic types), 502 types of relationships and 4,884,494 triples. Although there are Chinese medical terminologies in UMLS, the number is limited. Hence, we use an open-source Chinese medical KG, CMedicalKG,[9] which includes 44,111 entities (7 categories), 10 types of relationships, and 294,149 triples.

B Case Study

Fig. 7 shows the examples in English and Chinese with various DA methods. We can observe

[9]https://github.com/liuhuanyong/QASystemOnMedicalKG

English Task	
Origin	Patients with sequelae of cerebral hemorrhage are paralyzed, aphasia, and vomiting, how to improve?
EDA	aphasia with sequelae of cerebral hemorrhage are paralyzed patients and vomiting, how to improve?
SR	Patients with sequelae of cerebral shed blood are paralyzed, aphasia, and vomiting, how to improve?
PHICON	Patients with sequelae of a traumatic brain injury frequently deal with paralyzed, stomach pain, constipation, how to improve?
KnowledgeDA	Patients with sequelae of stroke may face paralysis, speech deficits, and nausea, how to improve?

Chinese Task	
Origin	心悸和早搏是一回事吗？
SR	气喘和早搏是一回事吗？
EDA	心悸和早搏真的是一回事吗？
PHICON	急性肺损伤和视力障碍是一回事吗？
KnowledgeDA	咳嗽和感冒是一回事吗？

Figure 7: Examples of data augmentation in English and Chinese. Text chunks in blue are the entities in the original sentence and text chunks in red are the modified words/entities by DA methods.

that the sentence augmented by **KnowledgeDA** has a high quality as it can introduce more domain entities and the whole sentence has a good semantic meaning.

C Text Classification in Software Development

C.1 Dataset

We use a open data, *SO-PLC*[10], which is a Stack Overflow dataset for 4 programming language classification: python, C#, java, and javascript.

C.2 KG for Software Development

There is little research on building KGs for software development NLP tasks, and thus we decide to build one from scratch.

To build the KG, we refer to the software developer forum *Stack Overflow* to obtain raw text data.[11] *Stack Overflow* is one of the biggest forums for professional and enthusiastic software developers. Various technical questions are covered on the platform and marked with appropriate tags. These tags are usually programming-specific terminologies and can be beneficial to learn about tech ecosystems and the relationships between technologies (Singh et al., 2020). To build a KG from tags, we follow the existing KG construction process (Li et al., 2020a):

Step 1. Data Collection: We use programming languages (e.g., python, C#, java, and javascript) as keywords to search for related questions on Stack Overflow, and sort them according to 'most frequency'; then crawl the tags that appeared in the top 7,500 related questions (i.e., the first 150 pages).

Step 2. Entity Recognition: A tag is a word or phrase that mainly describes the key information of the question, which is usually a programming-specific terminology (Singh et al., 2020). Hence, we directly treat tags as the entity names in the KG.

Step 3. Relation Formation: There is usually more than one tag in one question. When multiple tags co-appear at the same question, we link them in the KG. Afterward, there is still a lack of entity types and edge types, and we use the community detection algorithm, Louvain (Blondel et al., 2008), to automatically classify tags, and the edge type is defined by the types of the two connected entities.

Finally, we get TagKG, which includes 6,126 entities (11 categories), 56 types of relationships, and 41,227 triples.

Method	SO-PLC	
	Acc.	F1
None	84.78(0.48)	84.65(0.50)
SR	84.72(0.32)	84.69(0.31)
EDA	84.78(1.08)	84.71(1.08)
PHICON	85.63(1.17)	85.60(1.19)
KnowledgeDA	**86.82(0.90)***	**86.83(0.88)****

Table 9: Performance on SO-PLC dataset (*BERT* as PLM)

C.3 Result

As illustrated in Table 9, there are almost no improvements or even slight decreases with EDA and SR, meaning these general DA methods are not suitable for the texts in software development forums. With the help of our constructed TagKG, **PHICON** achieves some performance gains by replacing same-category programming entities; this indicates that the category identified by the community detection algorithm in TagKG is effective for understanding software development related texts. By leveraging TagKG more comprehensively, **KnowledgeDA** works even better and improves accuracy and F1 score by 2.42% and 2.58%, respectively, compared with no-augmentation. It also implies that the construction of TagKG is valid.

[10] http://storage.googleapis.com/download.tensorflow.org/data/stack_overflow_16k.tar.gz

[11] https://stackoverflow.com/

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations

☑ A2. Did you discuss any potential risks of your work?
Limitations and Ethics Statement

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Sec. 1 (Introduction)

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Sec. 4

☑ B1. Did you cite the creators of artifacts you used?
Sec. 4

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Sec. 4

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Sec 4.1, Sec 4.2, Appendix A.1

C ☑ Did you run computational experiments?

Sec. 4.1.2

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Sec. 4.1.2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Sec. 4.1.1, Sec. 4.2, Appendix A.1

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Sec. 4

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.