Knowledge-Guided Paraphrase Identification

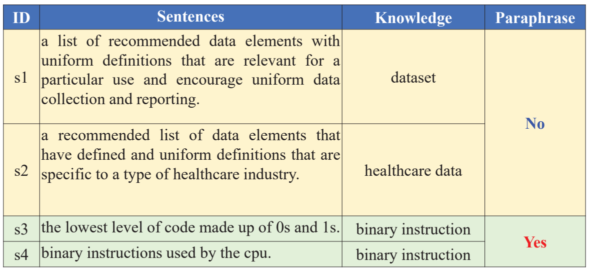
Haoyu Wang[§], Fenglong Ma[†], Yaqing Wang[§] and Jing Gao[§]
[§]Purdue University, West Lafayette, IN, USA
[†]The Pennsylvania State University, University Park, PA, USA
[§]{wang5346,wang5075,jinggao}@purdue.edu, [†]fenglong@psu.edu

Abstract

Paraphrase identification (PI), a fundamental task in natural language processing, is to identify whether two sentences express the same or similar meaning, which is a binary classification problem. Recently, BERT-like pretrained language models have been a popular choice for the frameworks of various PI models, but almost all existing methods consider general domain text. When these approaches are applied to a specific domain, existing models cannot make accurate predictions due to the lack of professional knowledge. In light of this challenge, we propose a novel framework, namely Knowing, which can leverage the external unstructured Wikipedia knowledge to accurately identify paraphrases. We propose to mine outline knowledge of concepts related to given sentences from Wikipedia via BM25 model. After retrieving related outline knowledge, Knowing makes predictions based on both the semantic information of two sentences and the outline knowledge. Besides, we propose a gating mechanism to aggregate the semantic information-based prediction and the knowledge-based prediction. Extensive experiments are conducted on two public datasets: PARADE (a computer science domain dataset) and clinicalSTS2019 (a biomedical domain dataset). The results show that the proposed Knowing outperforms state-of-the-art methods.

1 Introduction

Paraphrase identification (PI) is a classical yet fundamental natural language processing (NLP) task, which aims to determine whether a pair of sentences express the same or similar meaning (Bhagat and Hovy, 2013). Such a task can be used to examine whether a machine learning model really understands the semantic meanings of input sentences and is helpful for many other NLP tasks such as machine translation (Madnani et al., 2012) and question answering (Dong et al., 2017; Rinaldi et al., 2003).



ID	Sentences	Knowledge	Paraphrase
s1	a list of recommended data elements with uniform definitions that are relevant for a particular use and encourage uniform data collection and reporting.	dataset	No
s2	a recommended list of data elements that have defined and uniform definitions that are specific to a type of healthcare industry.	healthcare data	
s3	the lowest level of code made up of 0s and 1s.	binary instruction	Yes
s4	binary instructions used by the cpu.	binary instruction	

Figure 1: Examples of paraphrase identification.

To identify paraphrases automatically, machine learning models have been proposed. Traditional models (Mihalcea et al., 2006; Kozareva and Montoyo, 2006; Islam and Inkpen, 2009; Wan et al., 2006; Xu et al., 2014) focus on leveraging lexical and syntactic features to measure the similarity between two sentences. Recently, deep learning models are introduced and achieve the state-of-the-art performance. These models adopt convolutional neural networks (CNNs) (He et al., 2015; Filice et al., 2015), Long Short-Term Memory (LSTM) (Parikh et al., 2016; Chen et al., 2017; He and Lin, 2016; Nie and Bansal, 2017) or pretrained language models like BERT (Devlin et al., 2019). They directly learn the implicit relation between a pair of input sentences. However, *existing approaches all ignore the importance of knowledge associated with input sentences*.

In fact, each meaningful sentence usually belongs to a certain domain and contains domain-specific knowledge (He et al., 2020a). When domain experts identify whether these two sentences are paraphrases or not, they first read sentences to comprehend the semantic meanings, and then analyze them based on the domain knowledge associated with the sentences, and finally make a decision. As shown in Fig. 1, although S1 and S2 contain several matching words, experts know that they are not paraphrases because the first sentence

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 843–853
November 7–11, 2021. ©2021 Association for Computational Linguistics

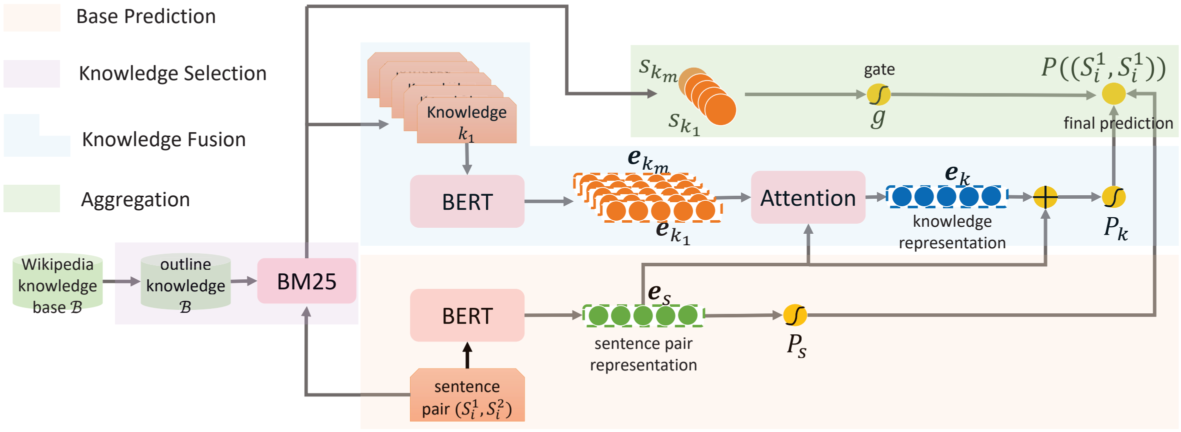


Figure 2: The framework of our proposed method.

describes the concept "dataset" but the second sentence does not. S3 and S4 even do not share a lot of lexical and syntactic features, but they have the same meaning since binary instructions are made up of 0s and 1s, which corresponds to the computer science domain knowledge. Therefore, for the PI task, only relying on lexical and syntactic features is insufficient, and it is indispensable to introduce domain knowledge into the models.

We will meet several technical challenges when introducing domain knowledge into the PI task: (1) **Knowledge selection**. Even in a specific domain, there is a huge volume of domain knowledge. The format of domain knowledge is either structured knowledge base or unstructured text. Thus, using which kind of domain knowledge and how to retrieve related knowledge for each sentence efficiently and accurately are new challenges. (2) **Knowledge fusion**. After we choose appropriate knowledge for each sentence (e.g., a description of a knowledge concept), the challenge is how to automatically incorporate such unstructured knowledge into state-of-the-art identification models to make more accurate predictions.

To solve the aforementioned challenges, in this paper, we propose a **know**ledge-**in**fused **g**ated model (named Knowing), which is shown in Fig. 2. Knowing consists of four main components: a base prediction module, a knowledge selection module, a knowledge fusion module, and an aggregation module. The base module is to encode sentence pairs and make predictions based on their lexical and syntactic information. Then we incorporate domain knowledge, and the first step is to collect related knowledge. The knowledge selection module retrieves top-m related knowledge outlines from Wikipedia via BM25 (Sanderson et al., 2010) for each sentence pair. Then the knowledge fusion module is to encode knowledge via an attention mechanism and get the knowledge-based prediction. In the end, the aggregation module aggregates the lexical and syntactic feature-based prediction and knowledge-based prediction via a novel gate mechanism.

The main contributions can be summarized as follows:

- To the best of our knowledge, we are the first to focus on domain-specific paraphrase identification and the first to infuse unstructured Wikipedia knowledge into BERT for paraphrase identification.
- We propose an effective and efficient way to use unstructured Wikipedia knowledge, which uses the outline of each concept and retrieves them via BM25.
- We propose a novel gated mechanism to automatically aggregate the lexical and syntactic feature-based prediction and knowledge-based prediction.
- The proposed model outperforms state-of-the-art paraphrase identification models on two public domain-specific datasets.

2 Preliminaries

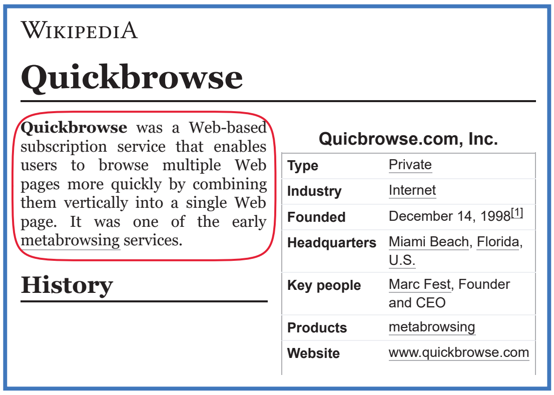
Before formally introducing the proposed model Knowing, we first mathematically define our task and the knowledge that we use in this paper.

2.1 Problem Formulation

Given a sentence pair $(S_i^1, S_i^2) \in \mathcal{X}$ and an external knowledge base $\mathcal{B}$, our goal is to learn a function $F(S_i^1, S_i^2, \mathcal{B}) \rightarrow \{0, 1\}$ to determine whether the two sentences S_i^1 and S_i^2 have the same or similar semantic meaning, where $\mathcal{X} = \{(S_1^1, S_1^2), \cdots, (S_n^1, S_n^2)\}$ denotes the training dataset, and the corresponding labels are $\mathcal{Y} = \{y_1, \cdots, y_n\}$.

2.2 External Knowledge

One major contribution of this work is to incorporate external knowledge to enhance the performance of paraphrase identification. Thus, the selection of knowledge base is crucial. In this paper, we use the most popular knowledge base, i.e., Wikipedia. Each concept in Wikipedia is associated with extensive descriptions, including the outline, the definition, its functions, related concepts, and so on. The length of the whole knowledge description is usually greater than 512, which exceeds the maximum size requirement of BERT-like pre-trained language models. In fact, compared with the other sections of the description, the outline is informative and is the high-level abstraction of the corresponding knowledge. Thus, instead of using the whole description of knowledge, we use the outlines of concepts as the external knowledge.


WIKIPEDIA

Quickbrowse

Quickbrowse was a Web-based subscription service that enables users to browse multiple Web pages more quickly by combining them vertically into a single Web page. It was one of the early metabrowsing services.

History

Quicbrowse.com, Inc.

Type	Private
Industry	Internet
Founded	December 14, 1998[1]
Headquarters	Miami Beach, Florida, U.S.
Key people	Marc Fest, Founder and CEO
Products	metabrowsing
Website	www.quickbrowse.com

Figure 3: The outline of concept Quickbrowse.

We take the concept "Quickbrowse"[1] as an example, which is shown in Fig. 3. The content selected in the red box is the outline that contains most of the important information even with a few words. Such outlines are more suitable for BERT-like pre-trained language models.

3 Methodology

3.1 Overview

The goal of this work is to effectively incorporate external knowledge to further improve the state-of-the-art performance of the paraphrase identification task. Towards this aim, we propose a new model Knowing as shown in Fig. 2, which includes four modules: (1) base prediction module, (2) knowledge selection module, (3) knowledge fusion module, and (4) aggregation module.

The *base prediction module* uses a pre-trained BERT model to encode sentence pairs via their lexical and syntactic features and makes predictions based on the sentence pair representations. However, this base predictor ignores the importance of external knowledge. To empower the effectiveness of knowledge, the *knowledge selection module* is designed to retrieve relevant outline knowledge from Wikipedia for each sentence pair. Since there may be several related outlines, to simultaneously take them into account, the *knowledge fusion module* first encodes each outline using the pre-trained BERT and then uses an attention mechanism to synthesize outline knowledge representation. The sentence pair representation obtained from the base prediction module and the fused knowledge representation learned by the knowledge fusion module are the inputs of the *aggregation module*. In particular, we design a gated function to aggregate them to learn the final representation that is used to identify paraphrases. Next, we provide the details of each module.

3.2 Base Prediction Module

Given a pair of sentences, the simplest way is to first learn a representation for each sentence by extracting lexical and syntactic features and then train a classifier to identify their relations. However, this simple approach may not achieve satisfactory performance since it does not model the interactions at the word level. To address this issue, we propose to use the powerful pre-trained language model BERT, which can model interactions among words between two sentences by directly feeding the sentence pair to BERT, i.e.,

$$\boldsymbol{e}_s = \mathrm{BERT}(S), \qquad (1)$$

where $S = S_i^1 \oplus S_i^2$ and $\oplus$ represents concatenation. The sentence pair representation is further used to identify the paraphrase relation by utilizing a

[1] https://en.wikipedia.org/wiki/Quickbrowse

fully connected layer (FC) followed by the sigmoid function as follows:

$$P_s = \sigma(\text{FC}(\boldsymbol{e}_s)), \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function.

This base prediction module may achieve satisfactory performance but it may still make mistakes on some sentences pairs that are difficult to match or distinguish. To further improve the performance of the PI task, we need to consider the utilization of domain knowledge. Next, we introduce how to select related knowledge for sentence pairs and then describe how to use the selected knowledge.

3.3 Knowledge Selection Module

The goal of the knowledge selection module is to automatically retrieve relevant knowledge for a given sentence pair from the set of knowledge outline $\mathcal{B}$. A straightforward solution is to adopt the pre-trained BERT model to encode sentence pairs and outlines, then calculate their similarity scores, and finally, select the top-m outlines with the highest scores. However, such an approach is inefficient and the computation and space complexity could be high due to the huge number of concepts in Wikipedia–there are about 5,903,527 concepts.

To prevent the complexity bottleneck, we propose to use the classical model BM25 (Sanderson et al., 2010) to estimate the relevance scores between outlines and sentence pairs. BM25 ranks a set of documents based on the query terms appearing in each document. Sentence pairs in a specific domain usually contain some professional terms, and we can treat them as the query terms, which makes it possible for us to use the simple but effective BM25 for knowledge selection. As an example, suppose we have a sentence pair "*a computer that manages web site services, such as supplying a web page to multiple users on demand.*" and "*provides information and services to web surfers.*". If an outline in knowledge base contains terms "web", "services" and "users", it may be useful to determine whether the two sentences talk about the same thing.

Mathematically, given a sentence pair (S_i^1, S_i^2), the knowledge selection module retrieves m relevant outlines $\{k_1, k_2, \cdots, k_m\}$ via BM25, and the corresponding relevance scores of the m outline knowledge are denoted as $\{s_{k_1}, s_{k_2}, \cdots, s_{k_m}\}$.

3.4 Knowledge Fusion Module

There are m relevant outlines selected by the knowledge selection module, and intuitively each of them contains informative knowledge. However, the outlines differ in the amount of useful information they can provide. Thus, we need to automatically learn a relevance score to distinguish the importance of outlines and then use the weighted sum operation to fuse all the outlines for synthesizing outline knowledge representation.

Towards this aim, we first encode the outline knowledge. Similar to the encoding of sentence pairs, we still use BERT to encode the outline knowledge, and the k_i knowledge representation is obtained as follows:

$$\boldsymbol{e}_{k_i} = \text{BERT}(k_i), i = 1, 2, \cdots, m. \tag{3}$$

To distinguish the importance of the m outlines for the prediction, we take advantage of the attention mechanism (Chorowski et al., 2015; Lian et al., 2020; Vaswani et al., 2017) to automatically assign an attention weight to each outline. Formally, the importance can be computed via

$$\boldsymbol{\alpha} = \text{Softmax}(\boldsymbol{e}_s^T \boldsymbol{M}\boldsymbol{E}), \tag{4}$$

where M is a learnable square matrix, and $\boldsymbol{E} = [\boldsymbol{e}_{k_1}, \boldsymbol{e}_{k_2}, \cdots, \boldsymbol{e}_{k_m}]$. Then we represent the whole knowledge via the weighted sum based on the importance values as follows

$$\boldsymbol{e}_k = \boldsymbol{E}\boldsymbol{\alpha}^T. \tag{5}$$

Using the learned knowledge representation $\boldsymbol{e}_k$ and the learned sentence pair representation $\boldsymbol{e}_s$, we can make a prediction. To enable them to fully interact with each other, we propose to use a fully connected layer (FC) followed by a Sigmoid function to get the prediction as follows:

$$P_k = \sigma(\text{FC}([\boldsymbol{e}_s, \boldsymbol{e}_k])). \tag{6}$$

3.5 Aggregation Module

Finally, the aggregation module is to synthesize the prediction P_s and P_k. A direct way is to use P_k or $(P_s + P_k)/2$ as the synthesized result. However, the outline knowledge is retrieved via BM25, so it may not be entirely accurate. Therefore, directly aggregating P_s and P_k as P_k or $(P_s + P_k)/2$ may introduce more noise if the knowledge is not that relevant to the sentence pair. In order to solve this problem, we design a gated mechanism to automatically control the weight of knowledge in the final prediction, i.e., the more relevant the knowledge, the larger the weight, and vice versa. Considering that BM25 also outputs the relevance scores of the knowledge while retrieving it, we use the relevance scores as the gate input to learn the weight of knowledge. Formally, the gate can be represented as:

$$g = \sigma(\boldsymbol{W}_2 \text{ReLU}(\boldsymbol{W}_1 \boldsymbol{s})), \tag{7}$$

where $\boldsymbol{s} = [s_{k_1}, s_{k_2}, \cdots, s_{k_m}]$, and $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are parameters to be learned. Finally, we can get the final prediction

$$P((S_i^1, S_i^2)) = P_s(1-g) + P_k g, \tag{8}$$

and the loss function of the proposed Knowing is

$$\mathcal{L} = - \sum_{(S_i^1, S_i^2) \in \mathcal{X}, y_i \in \mathcal{Y}} (y_i \log P((S_i^1, S_i^2)) + (1-y_i)\log(1 - P((S_i^1, S_i^2)))). \tag{9}$$

4 Experiments

In this section, we empirically validate the effectiveness of the proposed Knowing model. To explore the insights behind Knowing model, we explore the role of Wikipedia knowledge on specific domains and the role of the proposed gated mechanism.

4.1 Experiment Settings

Knowledge base. In this paper, we use Wikipedia as the knowledge base to assist the paraphrase identification task. More specifically, the number of collected knowledge outlines from Wikipedia is 5,903,527.

Datasets. We use two public datasets, PARADE (He et al., 2020a) and clinicalSTS2019 (Wang et al., 2020), to evaluate the model performance. PARADE is a computer science domain benchmark dataset for paraphrase identification, while clinicalSTS2019 belongs to the biomedical domain. For PARADE dataset, we use the same training, validation, testing splits with He et al. (2020a). In clinicalSTS2019, the similarity score of each sentence pair ranges from 0 to 5, where 0 indicates irrelevance, and 5 indicates the equivalence in semantic meanings between the two sentences. Since paraphrase identification is a binary classification task, in order to use this dataset, we need to convert the six classes to two categories. More specifically, we set the labels of instances with scores 0, 1, and 2 as 0, and the remaining ones as 1. Since there is no validation set in the clinicalSTS2019 dataset, we construct one by randomly sampling 10% pairs of its training set. The statistics are shown in Table 1.

Table 1: Statistics of datasets.

Dataset	#Training	#Validation	#Testing
PARADE	7,550	1,275	1,357
clinicalSTS2019	1,478	165	413

Baselines. We compare the proposed Knowing with following state-of-the-art baselines: DecAtt (Parikh et al., 2016), ESIM (Chen et al., 2017), PWIM (He and Lin, 2016), SSE (Nie and Bansal, 2017), BERT (Devlin et al., 2019) and ALBERT (Lan et al., 2019) (BERT and ALBERT use the same backbone with ours). DecAtt (Parikh et al., 2016) is short for the Decomposable Attention Model, which uses attention to model the sentence pairs. ESIM (Chen et al., 2017) uses BiLSTM to encode sentences and models the word pair interactions using the same way as DecAtt. PWIM (He and Lin, 2016) uses LSTM to learn sentence representation and applies dot product, cosine similarity and Euclidean distance together to measure the similarity. SSE (Nie and Bansal, 2017) applies stacked bidirectional LSTM-RNNs with shortcut connections to encode sentences. BERT (Devlin et al., 2019) is considered as the state-of-the-art model for many NLP tasks including paraphrase identification. ALBERT (Lan et al., 2019) compresses the architecture of BERT and achieve better performance in benchmarks. In our paper, we use BERT-base-uncased and ALBERT-base-v2 as two baselines and the backbones of Knowing.

Evaluation Metrics. Following (He et al., 2020a), we employ Accuracy, Precision, Recall, and F1 score as the evaluation metrics.

Implementation Details. We implement BERT via the hugginface library[2], and the training batch size is set to 8. During training, we set the learning rate for the backbone BERT and ALBERT parameters for Knowing as $2e-5$ and use a different learning rate $1e-4$ for newly added parameters to facilitate training. The optimizer in our experiments is AdamW following Devlin et al. (2019), and the training epoch of Knowing is set as 4. The implementations of DecAtt, ESIM, PWIM and SSE are based on Lan and Xu (2018)[3], and we follow

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

[3] https://github.com/lanwuwei/SPM_toolkit

Table 2: The results of different methods on two datasets. "Acc" and "Prec" mean Accuracy and Precision respectively. "IMP" represents the improvement brought by Knowing. The results of baselines on PARADE are reported from He et al. (2020a).

	PARADE					clinicalSTS2019				
	Acc	Prec	Recall	F1	IMP(%)	Acc	Prec	Recall	F1	IMP(%)
DecAtt	0.5400	0.5190	0.5410	0.5300	38.5	0.7112	0.4901	0.6379	0.5543	57.2
ESIM	0.5950	0.5560	**0.7700**	0.6460	13.6	0.8447	0.7131	0.7500	0.7311	19.2
PWIM	0.7010	0.6890	0.6860	0.6870	6.8	0.8786	0.8000	0.7586	0.7788	11.9
SSE	0.6890	0.6490	0.7640	0.7020	4.5	0.8786	0.7578	0.8362	0.7951	9.6
BERT	0.7290	0.6870	0.7310	0.7080	3.6	0.8956	**0.8842**	0.7241	0.7962	9.4
ALBERT	0.7067	0.6680	0.7708	0.7157	2.5	0.9126	0.9082	0.7672	0.8318	4.8
Knowing (BERT)	0.7369	0.7120	0.7569	**0.7338**	–	**0.9248**	0.8400	**0.9052**	**0.8714**	–
Knowing (ALBERT)	**0.7377**	**0.7311**	0.7154	0.7232	–	**0.9248**	0.8632	0.8707	0.8670	–

their recommended hyper-parameters. We run experiments on a server with one TITAN Xp. We repeat the experiments 5 times and report the average results.

4.2 Performance Comparison

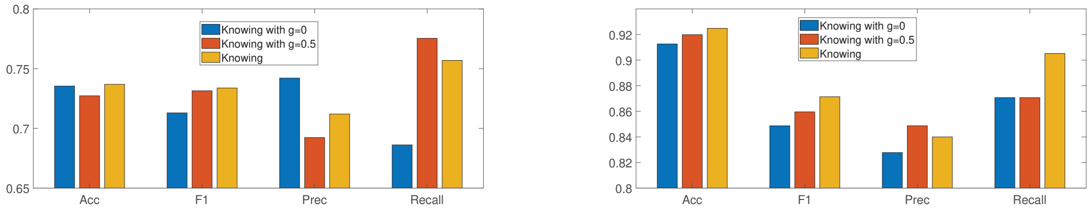
Table 2 shows the performance comparison of all the models on two datasets. Our proposed Knowing improves over baselines largely in terms of F1 score, Accuracy, Precision and Recall on the two datasets. Compared with the baselines BERT , the improvement brought by the proposed Knowing is 3.6% on PARADA dataset and 9.4% on clinicalSTS2019 dataset in terms of F1 score respectively. And compared with the baselines ALBERT , the improvement brought by the proposed Knowing is 2.5% on PARADA dataset and 4.8% on clinicalSTS2019 dataset in terms of F1 score respectively. Although BERT and ALBERT are proven to be effective on general domain datasets such as MSRP (Dolan et al., 2004), it is still difficult for BERT and ALBERT to achieve good performance on specific domain datasets. The main challenge is that professional glossaries in PARADE and clinicialSTS2019 are rarely used in general corpora and such data characteristics make the pre-training for BERT and ALBERT on this task less effective. The observation about the degraded performance of pre-training is validated through the comparison between BERT and other paraphrase identification methods like SSE, where the similar performance between BERT and SSE is observed on both of the datasets. Considering that the corresponding understanding of professional glossaries is usually relied on domain knowledge, we incorporate external domain knowledge into BERT and ALBERT, and propose a new model Knowing, which brings significant improvement on both datasets compared with state-of-the-art baselines. Such an observation confirms the importance of external knowledge for domain specific text.

4.3 Comparison with Methods Pre-trained on Domain Specific Corpora

In this section, we aim to explore how to effectively introduce external knowledge into language models. Besides using external knowledge as knowledge base as in the proposed model Knowing, another option is to pre-train language models on domain specific corpora to store external knowledge in model parameters. To further analyze these two options, we adopt several methods which pre-train BERT on biomedical domain corpora, such as BlueBERT (Peng et al., 2019), BioMedBERT (Chakraborty et al., 2020), SciBERT (Beltagy et al., 2019), and on computer science domain corpora such as SciBERT (pre-trained on Semantic Scholar with 18% of computer science papers and 82% biomedical papers) as baselines for an empirical comparison. The performance comparison is shown in Table 3.

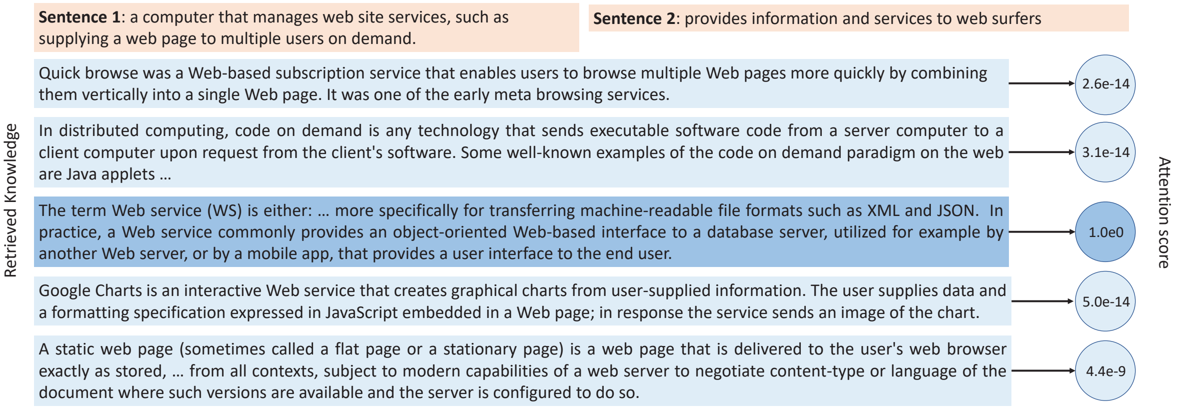
First, we can observe that the variants of BERT pre-trained on domain corpora perform better than vanilla BERT. Comparing BERT with BlueBERT, BioMedBERT, and SciBERT, BlueBERT, BioMedBERT and SciBERT perform better than BERT significantly on clinicalSTS2019, up to 5.5% improvement with respect to F1 score. And SciBERT perform better than BERT up to 2.1% improvement with respect to F1 score on PARADE. It confirms the importance of external knowledge and the effectiveness of pre-training BERT on biomedical domain and computer science domain corpora.

Even though pre-training on domain specific cor-



(a) PARADE (b) clinicalSTS2019

Figure 4: The performance of Knowing and its two variants on two datasets.



Sentence 1: a computer that manages web site services, such as supplying a web page to multiple users on demand.

Sentence 2: provides information and services to web surfers

Quick browse was a Web-based subscription service that enables users to browse multiple Web pages more quickly by combining them vertically into a single Web page. It was one of the early meta browsing services.

In distributed computing, code on demand is any technology that sends executable software code from a server computer to a client computer upon request from the client's software. Some well-known examples of the code on demand paradigm on the web are Java applets ...

The term Web service (WS) is either: ... more specifically for transferring machine-readable file formats such as XML and JSON. In practice, a Web service commonly provides an object-oriented Web-based interface to a database server, utilized for example by another Web server, or by a mobile app, that provides a user interface to the end user.

Google Charts is an interactive Web service that creates graphical charts from user-supplied information. The user supplies data and a formatting specification expressed in JavaScript embedded in a Web page; in response the service sends an image of the chart.

A static web page (sometimes called a flat page or a stationary page) is a web page that is delivered to the user's web browser exactly as stored, ... from all contexts, subject to modern capabilities of a web server to negotiate content-type or language of the document where such versions are available and the server is configured to do so.

Figure 5: An example of Knowing making predictions. The orange parts are given sentences, blue rectangle parts are knowledge, and blue circles are the attention scores.

Table 3: Results of BERT pre-trained on biomedical corpora and Knowing on clinicalSTS2019, and results of BERT pre-trained on computer science corpora and Knowing on PARADE. Results of "*" are taken from (He et al., 2020a).

clinicalSTS2019	Acc	Prec	Recall	F1
BERT	0.8956	0.8842	0.7241	0.7962
BlueBERT	0.8908	0.8142	0.7931	0.8034
BioMedBERT	0.8981	0.8246	0.8103	0.8174
SciBERT	0.9126	0.8635	0.8190	0.8407
Knowing (BERT)	0.9248	0.8400	0.9052	0.8714
Knowing (SciBERT)	0.9296	0.8655	0.8879	0.8766
PARADE	**Acc**	**Prec**	**Recall**	**F1**
BERT	0.7290	0.6870	0.7310	0.7080
SciBERT*	0.7410	0.7070	0.7400	0.7230
Knowing (BERT)	0.7369	0.7120	0.7569	0.7338
Knowing (SciBERT)	0.7362	0.7110	0.7569	0.7332

pora can incorporate external knowledge into parameters, it is less effective compared with the proposed Knowing. Compared to BlueBERT, BioMedBERT and SciBERT, the Knowing(BERT) brings at least 3.7% improvement in terms of F1 score on clinicalSTS2019. Compared to SciBERT, the Knowing(BERT) also brings 1.4% improvement in terms of F1 score on PARADE. Such an improvement demonstrates that it is more effective to treat external knowledge as knowledge base for retrieval instead of pre-training on domain corpora. Besides, Knowing(SciBERT) shows limited gain compared to Knowing(BERT), which demonstrates that Knowing(BERT) can take good advantage of external knowledge while additional pre-training may not bring more benefits.

4.4 Effectiveness of the Gated Mechanism

In this section, we explore the role of the proposed gated mechanism by analyzing two variants: (1) we set the gate value as 0 and correspondingly $P((S_i^1, S_i^2)) = P_k$; and (2) we aggregate P_s and P_k via the average operation, i.e., $P((S_i^1, S_i^2)) = (P_s + P_k)/2$. We report the performance comparison between these two variants and our proposed Knowing in Fig. 4.

First, we compare our proposed model Knowing against the model with $g = 0$. When $g = 0$, the model makes predictions solely based on external knowledge and thus the performance is degraded in term of Accuracy and F1 score compared with Knowing according to Fig. 4. It shows that accurate predictions cannot be achieved solely based on knowledge without taking semantic information into account.

Second, we compare our proposed model Knowing with another variant that is based on average operation (Knowing with $g = 0.5$). The proposed Knowing brings improvements in terms of Accuracy and F1 score on two datasets, especially on clinicalSTS2019. Such an observation illustrates the necessities of adaptive combination between external knowledge and semantic information of sentences instead of using a fixed ratio. These results clearly show that the proposed gated mechanism is able to automatically tune how much external knowledge is incorporated and this mechanism further improves the model performance.

4.5 Case Study

In this section, we use a concrete example from the PARADE dataset to show how the Knowing works, which is shown in Fig. 5. The sentence pair is "*a computer that manages web site services, such as supplying a web page to multiple users on demand.*" and "*provides information and services to web surfers*". PARADE dataset contains the computer science topic attribute for each sentence pair. Since such an attribute is not very common for other datasets, we do not take the topic attribute as a part of inputs to our model, but we can use this information to verify the knowledge selection for analysis purpose. The topic attribute of the given example is *Web Service*. The attention mechanism of our model Knowing assigns the largest score to the third knowledge concept "Web Service". Such a selection aligns well with the given topic attribute value *Web Service*. Correspondingly, the gate value for external knowledge is 0.6402, which indicates that the selected knowledge concept is helpful in making the final prediction, which also aligns well with our observation.

5 Related Work

5.1 Paraphrase Identification

Traditional methods for paraphrase identification (PI) are based on word or string similarity measurements. VBS (Mihalcea et al., 2006) applies cosine similarity with tf-idf weighting. STS (Islam and Inkpen, 2009) and KM (Kozareva and Montoyo, 2006) measure the similarity based on both semantic and string similarity. MCS (Mihalcea et al., 2006) obtains the similarity scores based on multiple word similarity computation methods.

Recently, deep learning methods advance the performance for PI. REL-TK (Filice et al., 2015), L.D.C Model (Wang et al., 2016) and Multi-Perspective CNN (He et al., 2015) employ convolutional neural network (CNN) to extract features for similarity measurement. SAMS-RecNN (Cheng and Kartsaklis, 2015) and SHPNM (Socher et al., 2011) model sentence representations via recursive neural networks. ESIM (Chen et al., 2017), PWIM (He and Lin, 2016) and SSE (Nie and Bansal, 2017) apply LSTM to learn sentence representations for predictions. Both DecAtt (Parikh et al., 2016) and ESIM (Chen et al., 2017) employ attention to learn the interactions between two sentences.

BERT (Devlin et al., 2019) and other pre-trained language models (Liu et al., 2019) achieve state-of-the-art performance on PI. However, existing works do not incorporate domain knowledge for PI, and hence, cannot achieve satisfactory performance on domain specific PI task. Different from existing works, the proposed method exploits the unstructured knowledge and applies a novel gating mechanism to automatically aggregate the lexical and syntactic information for predictions.

5.2 Knowledge-enhanced Language model

Incorporating external knowledge into language model is effective for downstream tasks and recently attracts lots of attentions. Recent works (Zhang et al., 2019; Liu et al., 2020; Xiong et al., 2019; Peters et al., 2019; Cui et al., 2020; Song et al., 2021; Hu et al., 2019; Ye et al., 2019) explore how to introduce knowledge graphs to enhance language models for downstream tasks. However, knowledge graph may be not available for each domain since its construction needs lots of human efforts. Moreover, a structured knowledge graph contains entities and relations, but the knowledge associated with each entity may be incomplete, which may be difficult to provide enough help for paraphrase identification.

To take advantage of unstructured knowledge, a lot of works (Chakraborty et al., 2020; He et al., 2020b; Beltagy et al., 2019; Peng et al., 2019; Huang et al., 2019; Lee et al., 2020) propose to pre-train language models on domain specific text. However, the pre-training objective function is usually not designed to capture knowledge concepts and their explanations, and only leads to limited improvement with intensive computation costs. Compared with the existing works, our proposed method can leverage external knowledge effectively to achieve significant improvements without a computationally expensive pre-training stage.

6 Conclusions

In this paper, we investigated the important and challenging task of domain-specific paraphrase identification. Since domain-specific text is difficult to be understood without domain knowledge, we proposed to incorporate Wikipedia knowledge into our model. However, there are two major challenges: (1) how to select knowledge, and (2) how to incorporate the selected knowledge into state-of-the-art paraphrase identification models automatically.

To solve these challenges, we introduced Wikipedia as external knowledge base and proposed a knowledge-infused gated model named Knowing to fuse Wikipedia knowledge with BERT. The Knowing contains four modules: a base prediction module, a knowledge selection module, a knowledge fusion module, and an aggregation module. The base prediction module is to learn sentence pair representations and make predictions only based on sentence pair themselves. The knowledge selection module is to retrieve relevant knowledge from Wikipedia, and then knowledge fusion module applies an attention mechanism to synthesize the knowledge representations. Finally, the aggregation module is to aggregate the based module's predictions and the knowledge-based predictions via a gate function. The experiments on two public domain-specific datasets show that the proposed Knowing outperforms state-of-the-art baselines.

Acknowledgement

This work is supported in part by the US National Science Foundation under grant NSF IIS-1747614 and NSF IIS-2141037. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

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 3615–3620.

Rahul Bhagat and Eduard Hovy. 2013. What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

Souradip Chakraborty, Ekaba Bisong, Shweta Bhatt, Thomas Wagner, Riley Elliott, and Francesco Mosconi. 2020. Biomedbert: A pre-trained biomedical language model for qa and ir. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 669–679.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.

Jianpeng Cheng and Dimitri Kartsaklis. 2015. Syntax-aware multi-sense word embeddings for deep compositional models of meaning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1531–1542.

Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 577–585.

Leyang Cui, Sijie Cheng, Yu Wu, and Yue Zhang. 2020. Does bert solve commonsense task via commonsense knowledge? *arXiv preprint arXiv:2008.03945*.

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

William B Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886.

Simone Filice, Giovanni Da San Martino, and Alessandro Moschitti. 2015. Structural representations for learning relations between pairs of texts. In *Proceedings of the 53rd Annual Meeting of the Association* *for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1003–1013.

Hua He, Kevin Gimpel, and Jimmy Lin. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1576–1586.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 conference of the north American chapter of the Association for Computational Linguistics: human language technologies*, pages 937–948.

Yun He, Zhuoer Wang, Yin Zhang, Ruihong Huang, and James Caverlee. 2020a. Parade: A new dataset for paraphrase identification requiring computer science domain knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7572–7582.

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.

Yidan Hu, Gongqi Lin, Yuan Miao, and Chunyan Miao. 2019. Commonsense knowledge+ bert for level 2 reading comprehension ability test. *arXiv preprint arXiv:1909.03415*.

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

Aminul Islam and Diana Inkpen. 2009. Semantic similarity of short texts. *Recent Advances in Natural Language Processing V*, 309:227–236.

Zornitsa Kozareva and Andrés Montoyo. 2006. Paraphrase identification on the basis of supervised machine learning techniques. In *International conference on natural language processing (in Finland)*, pages 524–533. Springer.

Wuwei Lan and Wei Xu. 2018. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3890–3902.

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 *International Conference on Learning Representations*.

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.

Defu Lian, Haoyu Wang, Zheng Liu, Jianxun Lian, Enhong Chen, and Xing Xie. 2020. Lightrec: A memory and search-efficient recommender system. In *Proceedings of The Web Conference 2020*, pages 695–705.

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

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

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–190.

Rada Mihalcea, Courtney Corley, Carlo Strapparava, et al. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Aaai*, volume 6, pages 775–780.

Yixin Nie and Mohit Bansal. 2017. Shortcut-stacked sentence encoders for multi-domain inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 41–45.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255.

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*, pages 58–65.

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.

Fabio Rinaldi, James Dowdall, Kaarel Kaljurand, Michael Hess, and Diego Mollá. 2003. Exploiting paraphrases in a question answering system. In *Proceedings of the second international workshop on Paraphrasing*, pages 25–32.

Mark Sanderson, D Christopher, Hinrich Manning, et al. 2010. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100.

Richard Socher, Eric H Huang, Jeffrey Pennington, Andrew Y Ng, and Christopher D Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *NIPS*, volume 24, pages 801–809.

Dandan Song, Siyi Ma, Zhanchen Sun, Sicheng Yang, and Lejian Liao. 2021. Kvl-bert: Knowledge enhanced visual-and-linguistic bert for visual commonsense reasoning. *Knowledge-Based Systems*, page 107408.

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 *Advances in neural information processing systems*, pages 5998–6008.

Stephen Wan, Mark Dras, Robert Dale, and Cécile Paris. 2006. Using dependency-based features to take the'para-farce'out of paraphrase. In *Proceedings of the Australasian language technology workshop 2006*, pages 131–138.

Yanshan Wang, Sunyang Fu, Feichen Shen, Sam Henry, Ozlem Uzuner, and Hongfang Liu. 2020. The 2019 n2c2/ohnlp track on clinical semantic textual similarity: Overview. *JMIR Medical Informatics*, 8(11):e23375.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence similarity learning by lexical decomposition and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1340–1349.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *International Conference on Learning Representations*.

Wei Xu, Alan Ritter, Chris Callison-Burch, William B Dolan, and Yangfeng Ji. 2014. Extracting lexically divergent paraphrases from twitter. *Transactions of the Association for Computational Linguistics*, 2:435–448.

Zhi-Xiu Ye, Qian Chen, Wen Wang, and Zhen-Hua Ling. 2019. Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models. *arXiv preprint arXiv:1908.06725*.

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*, pages 1441–1451.