PARADE: A New Dataset for Paraphrase Identification Requiring Computer Science Domain Knowledge

Yun He, Zhuoer Wang, Yin Zhang, Ruihong Huang, James Caverlee
Department of Computer Science and Engineering, Texas A&M University
{yunhe, wang, zhan13679, huangrh, caverlee}@tamu.edu

Abstract

We present a new benchmark dataset called PARADE for paraphrase identification that requires specialized domain knowledge. PARADE contains paraphrases that overlap very little at the lexical and syntactic level but are semantically equivalent based on computer science domain knowledge, as well as non-paraphrases that overlap greatly at the lexical and syntactic level but are not semantically equivalent based on this domain knowledge. Experiments show that both state-of-the-art neural models and non-expert human annotators have poor performance on PARADE. For example, BERT after fine-tuning achieves an F1 score of 0.709, which is much lower than its performance on other paraphrase identification datasets. PARADE can serve as a resource for researchers interested in testing models that incorporate domain knowledge. We make our data and code freely available.[1]

1 Introduction

Paraphrases are sentences that express the same (or similar) meaning by using different wording (Bhagat and Hovy, 2013). Automatically identifying paraphrases and non-paraphrases has proven useful for a wide range of natural language processing (NLP) applications, including question answering, semantic parsing, information extraction, machine translation, textual entailment, and semantic textual similarity.

Paraphrase identification (PI) is typically formalized as a binary classification problem: given two sentences, determine if they roughly express the same meaning. Traditional paraphrase identification approaches (Mihalcea et al., 2006; Kozareva and Montoyo, 2006; Wan et al., 2006; Das and Smith, 2009; Xu et al., 2014) mainly rely on lexical and syntactic overlap features to measure the semantic similarity between the two sentences. Examples include string-based features (e.g., whether two sentences share the same words), part-of-speech features (e.g., whether shared words have the same POS tags), and dependency-based features (e.g., whether two sentences have similar dependency trees).

s1: the lowest level of code made up of 0s and 1s.
s2: binary instructions used by the cpu.
Label: paraphrase (both describe "Machine Code")

s3: a graph representation that uses a 2d array such that if arr[i][j] == 1, there is an edge between **vertices** i and j
s4: a matrix which records the number of direct links between **vertices**
Label: paraphrase (both describe "Adjacency Matrix")

s5: how **the optimal solution** to **a linear programming problem** changes as the problem data are modified.
s6: how changes in the coefficients of **a linear programming problem** affect **the optimal solution**
Label: non-paraphrase

Table 1: Examples of paraphrases and non-paraphrases from the computer science domain. Judgments are made based on domain knowledge rather than lexical or syntactic features. Overlapping words (other than stop-words) are in bold and key different words are underlined.

However, these shallow lexical and syntactic overlap features may not effectively capture the domain-specific semantics of the two sentences. A typical situation where models based on these overlap features may fail is *a pair of sentences that overlap very little at the lexical and syntactic level but are semantically equivalent* ***based on domain knowledge***. Consider the two paraphrases s1 and s2 in Table 1. Both describe *machine code* though they have very little overlap. In order to correctly identify paraphrases like this pair, it is necessary to have specialized domain knowledge that a processor (CPU) can only understand binary instructions made up of 0s and 1s. On the other hand, *a pair*

[1] https://github.com/heyunh2015/PARADE_dataset

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 7572–7582,
November 16–20, 2020. ©2020 Association for Computational Linguistics

of sentences that overlap greatly at the lexical and syntactic level but are not semantically equivalent ***based on domain knowledge*** can also confuse both non-expert annotators and NLP models. Consider the non-paraphrase of s5 and s6 in Table 1 as an example. Sentence s5 is about a sensitivity analysis between the problem data and the optimal solution while s6 is about a sensitivity analysis between the coefficients and the optimal solution; these two cases are fundamentally different, requiring specialized domain knowledge of linear programming to distinguish the two. These examples highlight the importance of **specialized domain knowledge** for identifying paraphrases and non-paraphrases correctly.

Recent neural models (Nie and Bansal, 2017; Parikh et al., 2016; Chen et al., 2017) that go beyond traditional approaches based on lexical and syntactic features have demonstrated state-of-the-art performance on paraphrase identification. For example, BERT and its variants (Devlin et al., 2018; Liu et al., 2019; Yang et al., 2019; Lan et al., 2019; Raffel et al., 2019) have achieved the best results on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) on two paraphrase identification datasets: the Microsoft Research Paraphrase Corpus (MRPC) and Quora Question Pairs (QQP). Using massive pre-training data and a flexible bidirectional self-attention mechanism, BERT and its variants are able to better model the semantic relationship between sentences. Moreover, two recent studies (Petroni et al., 2019; Davison et al., 2019) observe that BERT without fine-tuning can even capture world knowledge and can answer factual questions like "place of birth" and "who developed the theory of relativity." Naturally, we are curious to know if these neural models can correctly identify paraphrases that require specialized domain knowledge like the examples shown in Table 1.

Hence, our overarching research goal is to create new datasets and enable new models for high-quality *paraphrase identification based on domain knowledge*. Because previous paraphrase datasets (Dolan and Brockett, 2005; Dolan et al., 2004; Xu et al., 2014; Lan et al., 2017; Iyer et al., 2017; Zhang et al., 2019) were not originally designed and constructed from the perspective of domain knowledge, to date there is no such dataset that requires specialized domain knowledge to discern the quality of two candidate sentences as paraphrases. As a first step, we focus in this paper on the computer science domain. Specifically, we require a dataset of paraphrases that overlap very little but are semantically equivalent, and of non-paraphrases that have overlap greatly but are not semantically equivalent based on computer science domain knowledge. Correspondingly, there is a research gap in understanding if modern neural models can achieve exemplary performance on such a dataset, especially in comparison with existing paraphrase identification datasets (that lack such specialized domain knowledge). In sum, this paper makes four contributions:

- First, we propose a novel extensible framework for inexpensively collecting domain-specific sentential candidate paraphrases that are characterized by specialized knowledge. The key idea is to leverage large-scale online collections of *user-generated flashcards*. We treat definitions on each flashcard's back side that correspond to a common entity on the front side (e.g., "machine code") as candidate paraphrases.

- Due to the noise in user-generated flashcards and heterogeneity in the aspects in the candidate paraphrases, our second contribution is a refinement strategy coupled with annotation by domain experts to create a new gold dataset called PARADE (**PARA**phrase identification based on **D**omain knowledg**E**). PARADE contains 4,778 (46.9%) paraphrases and 5,404 (53.1%) non-paraphrases that describe 788 distinct entities from the computer science domain and is the *first publicly available benchmark* for paraphrase identification based on domain knowledge.

- Third, we evaluate the quality of state-of-the-art paraphrase identification models on PARADE and existing paraphrase identification datasets like MRPC and QQP. We find that both state-of-the-art neural models (which have shown strong performance on existing PI datasets) and non-expert human annotators have poor performance on PARADE. For example, BERT after fine-tuning only achieves 0.709 in terms of F1 on PARADE compared to 0.893 on MRPC and 0.877 on QQP. Such a gap indicates the need for new models that can better exploit specialized domain knowledge.

- Finally, we show that incorporating external domain knowledge into the training of models like BERT offers the potential for improvements on PARADE. Concretely, we find that SciBERT – a BERT variant pre-trained on a corpus of computer science papers – improves the accuracy from 0.729 to 0.741. This improvement is encouraging, and suggests the need for further enhancements in incorporating domain knowledge into NLP models.

2 Related Work

Framework for Collecting Paraphrases: The basic idea of collecting a paraphrase dataset is to connect parallel data that are related to the same reference, like different news articles reporting the same event (MRPC) (Dolan and Brockett, 2005; Dolan et al., 2004), multiple descriptions of the same video clip (Chen and Dolan, 2011), multiple phrasal paraphrases on the web to describe the same concept (Hashimoto et al., 2011), different translations of a foreign novel (Barzilay and Elhadad, 2003), and multiple tweets that relate to the same topic (Xu et al., 2014) or contain the same URL (Lan et al., 2017).

In this paper, we propose a novel framework to collect sentential paraphrases from online user-generated flashcards, where different definitions (on the back of flashcards) of the same entity (on the front of flashcards) are probably paraphrases. The main advantage of this framework is that it can easily collect domain-specific paraphrases. Since flashcard websites like Quizlet are mainly used by students to prepare for quizzes and exams, these flashcards are often organized by subject, providing a rich source of domain-specific paraphrases.

Datasets for Paraphrase Identification: To our best knowledge, there are five publicly available sentential paraphrase identification datasets: Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005; Dolan et al., 2004) contains 5,801 pairs of sentences from news articles, PIT-2015 (Xu et al., 2014) contains 18,762 pairs of tweets on 500 distinct topics, Twitter-URL (Lan et al., 2017) contains 51,524 pairs of tweets containing 5,187 distinct URLs, Quora Question Pairs (QQP) (Iyer et al., 2017) contains 400K[2] pairs of question pairs on Quora and PAWS (Zhang et al., 2019) contains 53,402 pairs of sentences by using word scrambling methods based on QQP. These datasets were not originally designed and constructed from the perspective of domain knowledge. Hence, we present PARADE, the first sentential dataset for paraphrase identification based on domain knowledge as shown in Table 1, as a complement to these previous efforts.

Domain-Specific Phrasal Paraphrases: Some previous work aims to extract domain-specific **phrasal** paraphrases (Pavlick et al., 2015; Zhang et al., 2016; Ma et al., 2019), like "head" and "skull" in the Biology domain. In this paper, we focus on sentential paraphrases rather than phrasal paraphrases, which require models that consider context and domain knowledge.

Pre-trained Language Models with Domain Knowledge: Recently, some works have sought to incorporate domain knowledge into pre-trained language models such as BERT. For example, SciBERT (Beltagy et al., 2019) uses the same architecture as BERT-base but is pre-trained over a corpus of 1.14M papers, with 18% of papers from the computer science domain and 82% from the biomedical domain. It has been reported that SciBERT outperforms BERT-base which is pre-trained over Wikipedia and bookscorpus on a variety of tasks like named entity recognition in the both domains.

3 Collecting Domain-Specific Paraphrases from Online Flashcards

In this section, we propose a novel framework that constructs a domain-specific paraphrase corpus from online user-generated flashcards. We choose *computer science* as the target domain in this paper as a first step. The framework can be easily applied to construct datasets of other domains.

Many web platforms provide flashcards like Quizlet, StudyBlue, AnkiWeb, and CRAM. Each flashcard generated by a user is made up of an entity on the front and a definition describing or explaining the entity on the back. The purpose of flashcards is to help users to understand and remember concepts like "machine code."

Our core idea is that two different definitions probably express the same meaning if they have the same entity on the front. Hence, they can be paired as a candidate paraphrase. Our framework can collect arbitrarily many definitions generated by users independently, leading to broad coverage of how native speakers are likely to describe an entity in a specialized domain. By pairing the va-

[2] The size of QQP is much larger than other datasets but its authors claim that the ground-truth labels are not guaranteed to be perfect.

riety of definitions about concepts (like "machine code"), paraphrases and non-paraphrases that requires specialized domain (e.g., computer science) knowledge to discern are generated and collected.

3.1 Collecting Entity-Definition Pairs Related to Specialized Domains

We first collect domain-specific terminology and then collect entity-definition pairs from a popular flashcard website.

Domain-specific terminology: Ren et al. (2014) presented a dataset of 55,171 research papers in the computer science domain, collected from 2,414 conferences or journals, covering sub-fields like artificial intelligence, computer architecture, networking, and so on. Naturally, high document frequency phrases from these papers can be regarded as computer science terminology. Therefore, 3,813 phrases with document frequency higher than 20 are extracted from these papers, where examples are shown in Table 2:

Table 2: Examples of Computer Science Terminology with Document Frequency (DF)

Phrases	DF	Phrases	DF
sensor networks	939	mobile devices	425
information retrieval	688	source code	375
data structures	467	data structure	348
query processing	429	software systems	341

Next, we use these phrases as queries to search flashcards related to computer science from Quizlet, a well-known online flashcards website with a convenient search API.[3] To ensure paraphrases generated from the flashcards are related to the target domain, we only keep the flashcards where the entity on the front is drawn from our computer science terminology set (of size 3,813). Some example flashcards are presented in Table 3.

For each flashcard, we extract the entity from the front and the definition from the back to form an entity-definition pair. Since there are many duplicate entities and definitions on Quizlet flashcards, we merge the same definitions and group unique definitions by entities. Further, we only keep entities and definitions in English and in the form of pure text (some definitions contain images) and remove entities with fewer than 5 unique definitions. Finally, 30,917 unique entity-definition pairs are obtained.

Table 3: Examples of Flashcards Related to Computer Science Domain

Entity (Front)	Definition (Back)
Artificial Intelligence	s1: simulating logical thoughts, patterns and responses
Artificial Intelligence	s2: simulates human thinking and behavior, such as the ability to reason and learn
Artificial Intelligence	s3: the ability of a computer or a robot to learn from new information
Artificial Intelligence	s4: machines that can apply and acquire knowledge

3.2 Generating Candidate Paraphrases

For the definitions that describe or explain the same entity, it is not guaranteed that any two of them will form a paraphrase because the definitions might focus on different aspects or facets of the entity. An example is shown in Table 3, where the first two definitions s1 and s2 focus on the aspect of "simulation" while the other two definitions s3 and s4 focus on "learning new knowledge." Two definitions on different aspects of the entity are probably not a paraphrase. As a consequence, a random pair of definitions about the same entity has a low probability of expressing the same meaning.

Clusters of Definitions: Hence, we propose to cluster definitions of each entity to group entity-definition pairs that are likely to be on the same aspect. Intuitively, definitions that focus on the same aspect often share some overlapping terms and are likely to be grouped into the same cluster, and pairs of definitions from the same cluster are more likely to be a paraphrase, like s1 and s2, and s3 and s4 in Table 3. The definitions are first preprocessed with tokenization and lemmatization.[4] K-means is applied to cluster the definitions for each entity, where each definition is represented by the average of 300-dimensional word2vec (Mikolov et al., 2013) token embeddings trained over these definitions. Empirically, we set the number of clusters be half the number of definitions for each entity. Such a large number of clusters is helpful to filter out some noisy data like meaningless or ill-formed definitions because they are likely to be grouped into a single definition's cluster that can be discarded.

Sampling Candidate Paraphrases: Then, every two of the definitions from the same cluster are paired as a candidate paraphrase. Following Lan et al. (2017), we also filter out paraphrases where

[3] https://quizlet.com/subject/sensor-networks/

[4] The tokenizer and lemmatizer are from NLTK.

Table 4: Annotation Criteria

3- Completely equivalent: they clearly describe the same computer science concept with same details;
Example of label 3:
Text 1: its software that is freely available and its source code is also available.
Text 2: typically free software where source code is made freely available
Reason: the two sentences are clearly about the same concept ("open source software") with similar details.

2 - Mostly equivalent: as they clearly describe the same computer science concept but some unimportant information differ. Unimportant information include two categories: (1) some examples to explain the entity; and (2) some details can be inferred (based on computer science knowledge) from the overlapping part of the two texts;
Example of label 2:
Text 1: moves packets between computers on different networks. routers operate at this layer. ip and ipx operate at this layer.
Text 2: osi layer that moves packets between computers on different networks. routers & ip operate at this level.
Reason: they are talking about the same concept: "network layer", only some unimportant information differ (the detail "osi layer" in Text 2 can be inferred based on computer science knowledge: network layer is one of the layers in OSI model).

1 - Roughly equivalent: as they describe the same computer science concept but some important information differs or is missing; Important information here include any details except for the two categories in the previous criterion of label 2;
Example of label 1:
Text 1: term for when a scan fails to find real vulnerabilities. leaves unidentified risk in the code.
Text 2: malicious activity goes undetected.
Reason: the two sentences might be talking about the same concept: "false negatives", but some important information differ (the detail "...risk in the code..." in Text 1 cannot be inferred from Text 2 based on computer science knowledge).

0 - Not equivalent: as they describe two different computer science concepts;
Example of label 0:
Text 1: test without knowledge of system internals.
Text 2: attacker has no knowledge of the network environment (external attack).
Reason: the first sentence is talking about "system test" while the second one is about "system attack."

the two definitions are very similar like they only differ in punctuation or some typos, or one definition is a sub-string of the other. After that, we collect all candidate paraphrases and obtain a dataset with 10,182 pairs.

4 PARADE Dataset

In Section 3, we introduced our framework for generating domain-specific candidate paraphrases. Although each one of the candidate paraphrases is focused on the same topic (entity), we still need to confirm that the two definitions express the same meaning. In this section, we introduce our annotation strategy for candidate paraphrases and formally present the PARADE dataset for paraphrase identification based on domain knowledge.

4.1 Annotators with Domain Expertise

As discussed in Section 1, candidate paraphrases in our dataset can not be annotated correctly without specialized domain knowledge. Hence, unlike most previous works (Lan et al., 2017; Xu et al., 2015, 2014; Chen and Dolan, 2011) that hire workers from crowdsourcing platforms like Amazon Mechanical Turk, we invited students majoring in computer science as the annotators for this dataset. The 40 invited annotators include 5 Ph.D. students, 18 masters students, and 17 upper-level undergraduates. All have finished courses that cover almost all of the entities (topics) introduced in Section 3.1.

4.2 Annotation Criteria

Since the annotators have domain expertise, we expect them to provide more specific judgments than just true paraphrase or not. The annotation criteria are presented in Table 4: Completely equivalent (3), Mostly equivalent (2), Roughly equivalent (1), and Not equivalent (0). Labels of 3 and 2 are considered paraphrases, while 0 and 1 are non-paraphrases.

4.3 Annotation Quality Control

Annotators are asked to carefully read the annotation criteria before starting annotations. Each pair is randomly assigned to three annotators; the final ground-truth is decided by majority vote. We evaluate annotation quality of each annotator via Cohen's Kappa score (Artstein and Poesio, 2008) against the ground-truth. The average Cohen's Kappa score of the annotators is 0.65. Following Lan et al. (2017), we re-assign the data instances that were assigned to 2 annotators with low annotation quality (Cohen's Kappa score$<$0.4) to the best 5 annotators (Cohen's Kappa score$>$0.75) and ask them to re-label (give labels without seeing old labels) these data instances.

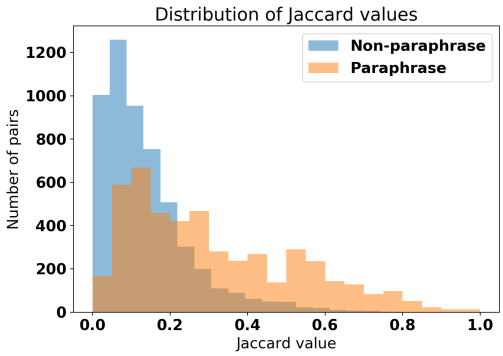

Figure 1: Distributions of Jaccard similarity for paraphrases and non-paraphrases in PARADE.

4.4 Dataset Description

Finally, we construct the first gold dataset for paraphrase identification based on domain knowledge, with 10,182 pairs of definitions that describe 788 distinct entities in the computer science domain. Among them, 4,778 (46.9%) are paraphrases and 5,404 (53.1%) are non-paraphrases. The average length of the definitions is 17.1 words and the maximum length is 30. An example from PARADE is shown in Table 5. Note that entities like "machine code" are also provided with definitions. However, these entities are not used in training and testing models for paraphrase identification tasks; otherwise the models will just learn the answers.

Table 5: An example of PARADE

Entity: Machine Code
Definition 1: the lowest level of code made up of 0s and 1s.
Definition 2: binary instructions used by the cpu.
Label: paraphrase

We calculate the Jaccard similarity for each pair to measure the lexical overlap[5] between the two definitions. In Figure 1, we illustrate the distributions of Jaccard similarity for paraphrases and non-paraphrases. It can be observed that PARADE contains lots of paraphrases that overlap very little at the lexical level but are semantically equivalent. PARADE also contain a few non-paraphrases that overlap a lot but are not semantically equivalent.

In Section 5, we present a qualitative analysis of PARADE on the cases where BERT give wrong predictions, which indicates that PARADE is truly enriched with domain knowledge.

5 Experiments

In this section, we present experiments that aim to answer the following research questions (RQs):

- RQ1: How do BERT and other neural models perform on PARADE? Better or worse than their performance on traditional PI datasets?
- RQ2: What kinds of domain knowledge are captured by PARADE? And how well do non-experts identify paraphrases that contain this domain knowledge?
- RQ3: Can we achieve high-quality identification by augmenting BERT-like models with a collection of domain-specific resources?

5.1 Experimental Setup

We first introduce our experimental setup here, including paraphrase identification models, other PI datasets and their partition and reproducibility.

Models for Binary Paraphrase Identification: We test seven different approaches on PARADE. The Decomposable Attention Model (**DecAtt**, 380K parameters) (Parikh et al., 2016) is one of the earliest models to apply attention for modeling sentence pairs. It computes the word pair interaction between the two sentences in a candidate paraphrase. The Pairwise Word Interaction Model (**PWIM**, 2.2M parameters) (He and Lin, 2016) uses Bi-LSTM to model the context of each word and then uses cosine similarity, Euclidean distance and dot product together to model word pair interactions. The Enhanced Sequential Inference Model (**ESIM**, 7.7M parameters) (Chen et al., 2017) first encodes sentences by using Bi-LSTM and then also calculates the word pair interaction between the two sentences like DecAtt. The Shortcut-Stacked Sentence Encoder (**SSE**, 140M parameters) (Nie and Bansal, 2017) applies a stacked Bi-LSTM with skip connections as the sentence encoder. Recently, the Bidirectional Encoder Representations from Transformer (**BERT**) (Devlin et al., 2018) obtains the state-of-the-art performance on many NLP tasks, including paraphrase identification. We evaluate **BERT-base** (12 layers and 768 hidden embedding size with 108M parameters) and **BERT-large** (24 layers and 1024 hidden embedding size with 334M parameters) on PARADE. We also adopt **ALBERT**, which compresses the architecture of BERT by factorized embedding parameterization

[5] Stopwords and punctuation were removed; words were stemmed.

and cross-layer parameter sharing, to obtain a substantially higher capacity than BERT. We choose the maximum version ALBERT-xxlarge (12 layers and 4096 hidden embedding size with 235M parameters).

Datasets and Their Partition: For PARADE, we randomly split it by entities into three parts: 7,550 with 560 distinct entities in the training set, 1,275 with 110 distinct entities in the validation set and 1,357 with 118 distinct entities in the testing set. For paraphrase datasets MRPC[6], PAWS[7], Twitter-URL[8] and PIT-2015[9], we follow the data partitioning strategy of their authors. For QQP[10], the labels for its test set at GLUE are private, so we treat its validation set at GLUE as the test set and sample another part from its training set as the validation set. Details of these previous PI datasets can be found in Section 2.

Reproducibility: PARADE and its split in this paper is released.[11] For BERT, we use a widely used pytorch implementation[12] and Adam optimizer with batch size 32 and learning rate 2e-5. We fine-tuned BERT for 20 epochs. We selected the BERRT hyper-parameters from the range as recommended in Devlin et al. (2018) and based on the performance in terms of F1 on the validation set. The implementations[13] of the other neural models are from Lan and Xu (2018), and we use the same hyper-parameters as recommended by Lan and Xu (2018).

5.2 RQ1: Paraphrase Identification Comparison

We first present the performance of BERT-large on PARADE and previous PI datasets in Table 6. Compared to datasets that lack domain knowledge, we observe that BERT yields the lowest performance on PARADE across all metrics. For example, BERT obtains 0.709 in terms of F1, which is much lower than its performance on the other datasets. Both the precision and the recall are relatively low, which indicates that identifying paraphrases in PARADE is non-trivial even for BERT.

Table 6: Performance of BERT on paraphrase identification datasets

BERT-large	Accuracy	F1	Precision	Recall
MRPC	0.853	0.893	0.866	0.922
QQP	0.908	0.877	0.866	0.889
PWAS	0.939	0.933	0.923	0.944
Twitter-URL	0.905	0.770	0.728	0.817
PIT-2015	0.901	0.746	0.803	0.697
PARADE	0.736	0.709	0.669	0.753

Table 7: Performance of Neural Models on PARADE

	Accuracy	F1	Precision	Recall
DecAtt	0.540	0.530	0.519	0.541
ESIM	0.595	0.646	0.556	0.770
PWIM	0.701	0.687	0.689	0.686
SSE	0.689	0.702	0.649	0.764
BERT-base	0.729	0.708	0.687	0.731
BERT-large	0.736	0.709	0.669	0.753
ALBERT-xxlarge	0.753	0.741	0.738	0.745

Additionally, we present the results of BERT-base, BERT-large, ALBERT-xxlarge and other neural models on PARADE in Table 7. We observe that the other neural models have lower performance than BERT-family models. Among the BERT-family models, BERT-large is slightly better than BERT-base, and ALBERT-xxlarge is the best due to its large learning capacity. However, the best performance is still relatively low on this dataset.

A possible reason is that these general neural net models do not sufficiently capture specialized knowledge of the computer science domain. BERT is pre-trained on two corpora: BooksCorpus (800M words) (Zhu et al., 2015) and English Wikipedia (2,500M words), which leads to some world knowledge learned as reported in Petroni et al. (2019); Davison et al. (2019). However, BooksCorpus[14] does not contain computer science books. While Wikipedia does contain articles on computer science, BERT may not pay enough attention to this subject since Wikipedia is such a huge corpus and computer science is just one branch.

5.3 RQ2: Domain Knowledge

As discussed in Section 5.2, BERT and other neural models face key challenges in paraphrase identi-

[6] https://gluebenchmark.com/tasks

[7] https://github.com/google-research-datasets/paws

[8] https://github.com/lanwuwei/Twitter-URL-Corpus

[9] https://cocoxu.github.io/#publications

[10] https://gluebenchmark.com/tasks

[11] https://github.com/heyunh2015/PARADE_dataset

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

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

[14] This corpus has 11,038 books like *American Psycho* and *No Country for Old Men.*

fication with domain knowledge as in PARADE. A possible reason is that PARADE has a lot of domain knowledge, which is beyond the lexical, syntactic features, or even commonsense knowledge captured by these models. To confirm the presence of domain knowledge, we first conduct a qualitative analysis of PARADE.

Table 8: A case where BERT predict incorrectly

Entity: Type Inference Definition 1: variables don't need explicit statements about their type unlike in **java**. **haskell** can automatically tell that **1** is of type **int**. Definition 2: allows the **compiler** to deduce the proper type for you automatically, instead of you having to say it.
BERT Prediction: non-paraphrase Ground-truth: paraphrase

Qualitative Analysis: We qualitatively analyzed 277 cases (171 paraphrase and 106 non-paraphrases) where BERT predicts the wrong results. From the perspective of domain knowledge, we count the occurrences of each phenomenon in the following categories: **Specialized Terminology.** Examples in the computer science domain include java, haskell and compiler in Table 8. **Acronyms and Abbreviations:** Examples include "int" for integer in Table 8, OS (operating system), OSI (Open Systems Interconnection) model and so on. **Numbers and Equations:** These have special meaning like "port: 80", "arr[i][j] == 1" and "an m-ary tree with m = 2". **Inference:** These non-overlapping sentences may be paraphrases based on domain-specific inference. For example in Table 8, definition 1 does not mention "compiler" in definition 2 but domain experts can infer that based on context and domain knowledge that the compiler is responsible for identifying the type. **Examples:** Non-overlapping paraphrases use examples to support the main idea, like the example of "haskell" in the definition 1 in Table 8. Although definition 2 does not have this example, they still express the concept of "type inference."

A typical example of the cases where these phenomena occur together is shown in Table 8. We report the number of occurrences of each phenomenon in Table 9 and observe that the cases where BERT fails have a high frequency of these domain knowledge phenomena, further supporting the assertion that PARADE is enriched with domain knowledge.

Table 9: Number of domain knowledge phenomena in the 277 cases where BERT mis-labels

Phenomenon	Count	Frequency
Specialized Terminology	150	0.54
Acronyms and Abbreviations	30	0.11
Numbers and Equations	31	0.11
Inference	114	0.41
Examples	78	0.28
Cases that have one phenomenon at least	197	0.71

Table 10: Performance of Non-Experts on Paraphrase Identification

Human	Accuracy	F1	Precision	Recall
MRPC	0.70	0.75	0.77	0.74
QQP	0.74	0.61	0.50	0.77
PAWS	0.90	0.88	0.86	0.90
Twitter-URL	0.90	0.71	0.67	0.75
PIT-2015	0.90	0.76	0.80	0.73
PARADE	0.62	0.56	0.45	0.73

Performance of Non-Experts without Domain Knowledge: To further confirm the presence of domain knowledge, we invite three college students who are **not** majoring in computer science to label PARADE and other datasets. Before evaluation, we ask the students to carefully read the annotation criteria of each dataset and 100 sampled cases with labels from the training set from each dataset. After that, 100 cases without ground-truth are randomly sampled from the test set of each dataset for evaluating the quality of non-expert annotators.

The results are presented in Table 10. We observe that non-experts without domain knowledge obtain abysmal performance on PARADE like 0.56 in terms of F1. However, on other datasets, these non-experts can achieve much better results like 0.88 in terms of F1 on PAWS. By interviewing these students, we believe they can correctly identify paraphrases based on lexical, syntactic and commonsense knowledge on all datasets except for PARADE, where the lack of specialized domain knowledge made the task too challenging.

5.4 RQ3: Incorporating Domain Knowledge

As shown in Section 5.2 and Section 5.3, both widely used neural models and non-expert human annotators have poor performance on PARADE. To corroborate the importance and possibility to enhance a model for PARADE by incorporating specialized domain knowledge, we ran an off-the-shelf model, SciBERT[15] (Beltagy et al., 2019), that

[15] https://huggingface.co/allenai/scibert_scivocab_uncased

uses the same architecture as BERT-base and is pretrained using 1.14M papers from Semantic Scholar (Ammar et al., 2018) with 18% of papers from the computer science domain and 82% from the biomedical domain. As shown in Table 11, SciBERT outperforms BERT consistently over all the metrics. This experiment shows that simply using corpora of a target domain for model training does lead to some improvements on PARADE. Further improvements may be achieved by methods that can more effectively infuse domain knowledge into NLP models.

Table 11: Results of Enhancing BERT by incorporating domain knowledge

	Accuracy	F1	Precision	Recall
BERT-base	0.729	0.708	0.687	0.731
SciBERT	0.741↑	0.723↑	0.707↑	0.740↑

6 Conclusion and Future Work

We have presented PARADE, a new dataset for sentential paraphrase identification requiring domain knowledge. We conducted extensive experiments and analysis showing that both state-of-the-art neural models and non-expert human annotators perform poorly on PARADE. In the future, we will continue to investigate effective ways to obtain domain knowledge and incorporate it into enhanced models for paraphrase identification. In addition, since PARADE provides entities like "machine code" for definitions, this new dataset could also be useful for other tasks like entity linking (Shen et al., 2014), entity retrieval (Petkova and Croft, 2007) and entity or word sense disambiguation (Navigli, 2009).

Acknowledgments

This work is supported in part by NSF (#IIS-1909252).

References

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. Construction of the literature graph in semantic scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 84–91, New Orleans - Louisiana. Association for Computational Linguistics.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Regina Barzilay and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 25–32.

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

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

David L Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 190–200. Association for Computational Linguistics.

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, Vancouver, Canada. Association for Computational Linguistics.

Dipanjan Das and Noah A Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 468–476. Association for Computational Linguistics.

Joe Davison, Joshua Feldman, and Alexander Rush. 2019. Commonsense knowledge mining from pretrained models. 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 1173–1178, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bill 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, Geneva, Switzerland. COLING.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Chikara Hashimoto, Kentaro Torisawa, Stijn De Saeger, Jun'ichi Kazama, and Sadao Kurohashi. 2011. Extracting paraphrases from definition sentences on the web. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1087–1097. Association for Computational Linguistics.

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.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs. *data. quora. com*.

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, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. *arXiv preprint arXiv:1708.00391*.

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, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1909.11942*.

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*.

Danni Ma, Chen Chen, Behzad Golshan, and Wang-Chiew Tan. 2019. Essentia: Mining domain-specific paraphrases with word-alignment graphs. *arXiv preprint arXiv:1910.00637*.

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.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):10.

Yixin Nie and Mohit Bansal. 2017. Shortcut-stacked sentence encoders for multi-domain inference. *arXiv preprint arXiv:1708.02312*.

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, Austin, Texas. Association for Computational Linguistics.

Ellie Pavlick, Juri Ganitkevitch, Tsz Ping Chan, Xuchen Yao, Benjamin Van Durme, and Chris Callison-Burch. 2015. Domain-specific paraphrase extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 57–62.

Desislava Petkova and W Bruce Croft. 2007. Proximity-based document representation for named entity retrieval. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 731–740. ACM.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Xiang Ren, Jialu Liu, Xiao Yu, Urvashi Khandelwal, Quanquan Gu, Lidan Wang, and Jiawei Han. 2014. Cluscite: Effective citation recommendation by information network-based clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 821–830. ACM.

Wei Shen, Jianyong Wang, and Jiawei Han. 2014. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Stephen Wan, Mark Dras, Robert Dale, and Cécile Paris. 2006. Using dependency-based features to take thepara-farceout of paraphrase. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 131–138.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Wei Xu, Chris Callison-Burch, and Bill Dolan. 2015. Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 1–11.

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.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Lilin Zhang, Zhen Weng, Wenyan Xiao, Jianyi Wan, Zhiming Chen, Yiming Tan, Maoxi Li, and Mingwen Wang. 2016. Extract domain-specific paraphrase from monolingual corpus for automatic evaluation of machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 511–517, Berlin, Germany. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.