Learning High-Quality and General-Purpose Phrase Representations

Lihu Chen, Gaël Varoquaux, Fabian M. Suchanek

Learning High-Quality and General-Purpose Phrase Representations

Lihu Chen[1], **Gaël Varoquaux**[1], **Fabian M. Suchanek**[2]
[1] Inria, Soda, Saclay, France
[2] LTCI, Télécom Paris, Institut Polytechnique de Paris, France
{lihu.chen, gael.varoquaux}@inria.fr
fabian.suchanek@telecom-paris.fr

Abstract

Phrase representations play an important role in data science and natural language processing, benefiting various tasks like Entity Alignment, Record Linkage, Fuzzy Joins, and Paraphrase Classification. The current state-of-the-art method involves fine-tuning pre-trained language models for phrasal embeddings using contrastive learning. However, we have identified areas for improvement. First, these pre-trained models tend to be unnecessarily complex and require to be pre-trained on a corpus with context sentences. Second, leveraging the phrase type and morphology gives phrase representations that are both more precise and more flexible. We propose an improved framework to learn phrase representations in a context-free fashion. The framework employs phrase type classification as an auxiliary task and incorporates character-level information more effectively into the phrase representation. Furthermore, we design three granularities of data augmentation to increase the diversity of training samples. Our experiments across a wide range of tasks show that our approach generates superior phrase embeddings compared to previous methods while requiring a smaller model size. The code is available at https://github.com/tigerchen52/PEARL

1 Introduction

A phrase is a group of words (or a single word) with a special meaning. They may denote recognizable entities: names of people (*Albert Einstein*), organizations (*The New York Times*), dates (*23 February 2008*), and events (*2024 Summer Olympics*). Beyond these typical contexts, phrases also appear as column names in tabular data (*average_wage*), as user queries (*black pant men*), or even as a non-noun phrase in clinical reports (*more than 63kg*). Phrases are thus an important building block in many applications of both data science and natural language processing (NLP), e.g., in tasks such as Entity Alignment (Zhao et al., 2020), Fuzzy Joins (Yu et al., 2016), Question Answering (Lee et al., 2021), Record Linkage (Christen, 2011), and Syntactic Parsing (Socher et al., 2010). Central to these applications is the assessment of the semantic similarity between two distinct phrases. Today, the main tool to assess the similarity of phrases is *phrase embeddings*, i.e., learned vector representations that capture the semantics of the phrases in such a way that phrases with similar meanings are close in representation space.

Input Entity Name: **The New York Times**

Phrase	Phrase-BERT (110 M)	UCTopic (253 M)	PEARL (40 M)
nytimes.com ✓	0.7576 (4)	0.7424 (3)	0.8849 (1)
NYTimes ✓	0.6441 (5)	0.6961 (4)	0.8828 (2)
New-York Daily Times ✓	0.9429 (2)	0.7563 (2)	0.8718 (3)
New York Post ✗	0.9435 (1)	0.8655 (1)	0.8527 (4)
New York ✗	0.7586 (3)	0.5404 (5)	0.6891 (5)

Figure 1: An example of entity retrieval. Given the input entity name *"The New York Times"*, we show the cosine similarity obtained by different models. The ranking of scores is listed in parentheses.

The difficulty of learning such representations arises from the fact that phrases often appear without context (e.g., in user queries), and exhibit diverse morphological variations. For example, given the entity *"The New York Times (Q9684)"*, the knowledge base Wikidata (Vrandečić and Krötzsch, 2014) offers multiple aliases (alternate names)[1]. Three of them are shown in the first rows of Figure 1. The last two rows show names of other entities: *"New York Post (Q211374)"* and *"New York (Q1384)"*. While all five of these phrases look very much alike, only the first three are associated with *"The New York Times"*. This versatility of phrases makes it hard to use rule-based or string-distance methods for semantic similarities. Sentence-BERT (Reimers and Gurevych, 2019)

[1] https://www.wikidata.org/wiki/Q9684

was the first approach to fine-tune pre-trained language models like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) to derive meaningful sentence embeddings. However, Sentence-BERT is given entire sentences during training (no special focus on short texts or phrases), so that its capabilities to embed phrases remain limited. Phrase-BERT (Wang et al., 2021a) was explicitly designed to embed phrases and adopts contrastive learning to fine-tune BERT on lexically diverse phrasal paraphrase pairs and their surrounding context, yielding more powerful phrase embeddings. Another context-aware approach, UCTopic (Li et al., 2022), further improved phrase representations by using cluster-assisted negative sampling i.e., leveraging clustering results as pseudo-labels.

However, this prior work faces several limitations. First, phrases frequently appear devoid of context cues, especially in tabular data, and are often characterized by short lengths. Consequently, we might not actually need the complex reasoning abilities of large (or deep) language models. A small (or shallow) neural architecture could suffice for the purpose of capturing phrase semantics. Also, we need a model that works well in the absence of context. Second, existing work partially neglects the type information of phrases. For example, although *"The New York Times"* and *"New York"* have a high lexical overlap, a good representation model should distinguish them since the first phrase pertains to an organization while the second is linked to a geopolitical entity. Third, existing sub-word embeddings are not robust against out-of-vocabulary words (Chen et al., 2022), and this vulnerability entails the necessity of using character-level features and morphological information. Indeed, as Figure 1 shows, Phrase-BERT and UCTopic fail to recognize that *"NYTimes"* is an abbreviation of the original phrase, and wrongly rank *"New York Post"* (a different newspaper) as closest to *"The New York Times"*.

In this paper, we present a context-free contrastive learning framework called PEARL[2], which enriches existing language models by incorporating phrase type and character-level features. Additionally, PEARL uses a range of data augmentation techniques to increase training samples. PEARL has the following advantages: First, it is able to discern between phrases that share similar surface forms but are of different semantic types. For example, a model using our framework sees *"New York"* as a poor match for *"The New York Times"* as it is of a different type: a geopolitical entity versus an organization (Figure 1). Second, our approach captures morphology in phrases better. In Figure 1, our method correctly ranks all three positive candidates, including those with acronyms, as *NYTimes*. Third, a PEARL model of relatively small size (40M parameters) can outperform existing larger models (Phrase-BERT and UCTopic) and it learns phrase embeddings in a context-free fashion. This results in shorter training times and less resource consumption, which makes our approach more accessible in low-resource scenarios and reduces its carbon footprint.

We conduct extensive experiments with PEARL across various phrase and short text tasks, including Paraphrase Classification, Phrase Similarity, Entity Retrieval, Entity Clustering, Fuzzy Join, and Short Text Classification. We can show that our method outperforms other competitors across all these tasks – despite a smaller model size.

2 Related Work

Phrases are fundamental linguistic units, pivotal to understanding languages. Hence, learning their representations has attracted quite some attention in the research community. Early works mostly use compositional transformation to obtain phrasal embeddings, i.e., they derive phrase representations from word embddings (Mitchell and Lapata, 2008; Socher et al., 2012; Hermann and Blunsom, 2013; Yu and Dredze, 2015; Zhou et al., 2017). With the advent of large pre-trained models, recent approaches fine-tune transformer models like BERT (Devlin et al., 2019) to obtain generalized text embeddings, e.g. Sentence-Bert (Reimers and Gurevych, 2019) and E5 (Wang et al., 2022). However, a recent study suggests that phrase representations in these language models heavily rely on lexical content while struggling to capture the sophisticated compositional semantics (Yu and Ettinger, 2020). To develop more powerful models dedicated to phrasal representations, Phrase-BERT (Wang et al., 2021a) fine-tunes BERT on lexically diverse datasets by using both phrase-level paraphrases and context sentences around phrases. This allows the production of embeddings that go beyond simple lexical overlap. Another context-aware model, UCTopic (Li et al., 2022),

[2]Phrase Embeddings by Augmented Representation Learning

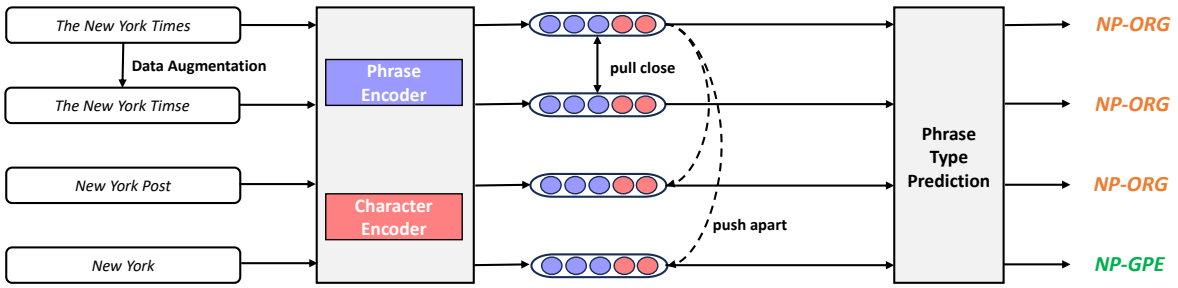


Figure 2: An illustration of PEARL. It uses contrastive learning and an auxiliary task of phrase type prediction for learning phrase embeddings.

proposes cluster-assisted contrastive learning for inducing phrasal representations for topic mining. McPhraSy (Cohen et al., 2022) incorporates context information into phrase embeddings during inference. Although these methods can effectively generate semantically meaningful phrasal representations, they ignore the phrase type and morphological information, which are crucial for understanding phrases. In this paper, we show that our approach can outperform these models with a much smaller model.

In the field of data science, a task closely related to phrase representation is string matching. It is widely used across diverse applications, including Fuzzy Join (Yu et al., 2016), Entity Resolution (Papadakis et al., 2020) or Alignment (Zhao et al., 2020), and Ontology Matching (Otero-Cerdeira et al., 2015). A simple yet effective solution for this task is similarity functions such as the Edit Distance and Jaccard similarity, which assess either token-level or character-level (or n-gram) similarity. More refined methods resort to word embeddings like GloVe (Pennington et al., 2014) and Fasttext (Bojanowski et al., 2017) to better capture lexical meaning. In this work, we show that models trained by our framework can be used for a series of database or knowledge base related tasks and achieve competitive results at little cost.

3 Our Approach

Our objective is to learn representations for arbitrary input phrases. For this, we design a novel contrastive-learning framework named PEARL, as shown in Figure 2. The input for PEARL is ***context-free phrases***. This is different from other existing models like Phrase-BERT (Wang et al., 2021a) and UCTopic (Li et al., 2022) which take phrases with context as input. Given a specific phrase, PEARL first applies data augmentation in order to obtain similar phrases that will serve as positive samples. For example, *"The New York Times"* becomes *"The New York Timse"* by using a character-level augmentation (character swap). Next, embeddings are generated by both phrase-level and character-level encoders. We then learn embeddings with the help of contrastive loss, which aims to pull close positive pairs while pushing apart in-batch negative samples. In order to learn more expressive representations, we add a certain number of hard negatives to each batch. For example, *"New York Post"* and *"New York"* can serve as hard negatives, given their high lexical overlap with the original phrase coupled with very distinct semantics. To integrate phrase structural information into the representations, we force the framework to assign tags of a lexical class and a named entity type to each phrase. For example, the framework learns to assign a `NP-ORG` tag to the phrase *"The New York Times"*, meaning that the phrase is a noun phrase associated with an organization. The negative sample *"New York"*, in contrast, receives a `NP-GPE` tag, meaning that the phrase is a noun phrase linked to a geopolitical entity. This augmentation with entity type information allows the model to distinguish *"The New York Times"* and *"New York"* in the representation space.

3.1 Data Augmentation

The positive pairs in contrastive learning are generated by data augmentation, and we use three different granularity methods to create training samples, as shown in Figure 3.

Character-level Augmentation aims to add morphological perturbations to the characters inside a single word. The goal is to make the representations robust against variations so that phrases that

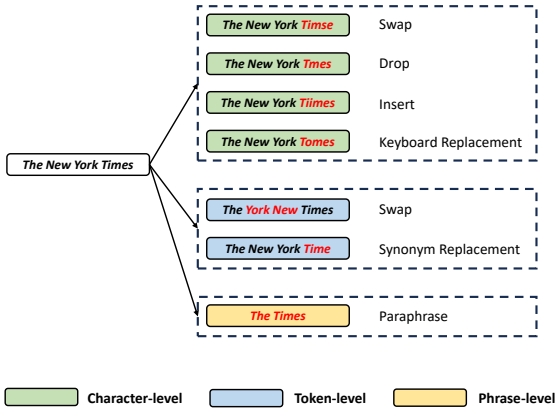


Figure 3: Different levels of granularity for the data augmentation methods on *"The New York Times"*.

have the same meaning but slightly different surface forms (e.g., misspellings) can be pulled close in the representation space. We adopt four types of character-level augmentations, inspired by Out-of-Vocabulary models (Pruthi et al., 2019; Chen et al., 2022): (1) Swap two consecutive characters, (2) drop a character, (3) insert a new character, (4) replace a character according to keyboard distance.

Token-level Augmentation modifies tokens in phrases for constructing positive samples. One method is to swap the order of two adjacent tokens, as in *"New York"* → *"York New"*. Another method is Synonym Replacement, which substitutes a token in a phrase with a synonymous one from a lexical dictionary. For example, *"New York newspaper"* can be transformed to *"NYC newspaper"*. We use two methods to retrieve synonyms: First, we draw synonyms from the lexical database WordNet (Miller, 1992). Second, we use the word embeddings of FastText (Bojanowski et al., 2017). We regard word pairs whose vector cosine similarity is greater than a certain threshold as synonyms.

Phrase-level Augmentation paraphrases an input phrase for generating completely diverse samples. Specifically, we employ a text-to-text paraphraser called Parrot (Damodaran, 2021). For instance, consider the input phrase *"The New York Times"*. Through the usage of Parrot, an alternative name such as *"The Times"*[3] can be generated as output. This augmentation stands distinct from character and token methods, thereby broadening the diversity of positive samples.

[3] *"The Times"* is an ambiguous name, and it can also mean a British daily national newspaper based in London.

3.2 Encoder

Phrases that are semantically similar can differ both on the token level (as in *"adult male"* vs. *"grown man"*) and on the character level (as in *adult* vs. its typo *adlut*). To cater to both variations, we feed the input phrase into both a phrase-level encoder and a character-level encoder and concatenate the two embeddings.

Phrase Encoder. We use E5 (Wang et al., 2022) as our phrase encoder. E5 is a general-purpose text embedding model pre-trained on curated large-scale (270 million) text pairs. It is able to transfer to a wide range of tasks requiring a single-vector representation of texts such as classification, retrieval, and clustering.

Character Encoder. We take inspiration from LOVE (Chen et al., 2022), a lightweight out-of-vocabulary model, to generate character-level embeddings. LOVE can produce word embeddings for arbitrary unseen words such as misspelled words, rare words, and domain-specific words, and it learns the behavior of pre-trained embeddings using only the surface form of words. We feed the vector obtained by LOVE to a fully connected layer to reduce its dimension.

3.3 Phrase Type Classification

The semantic type of a phrase is an important piece of information for distinguishing phrases that share similar surface forms but possess different meanings (such as *"The New York Times"* and *"New York"*). To integrate the phrase type into the learning framework, we design an auxiliary training task, Phrase Type Classification, which aims to predict the tags of the lexical phrase class and entity types for an input phrase. We use the following lexical tags during training: Noun Phrase (`NP`), Verb Phrase (`VP`), Prepositional Phrase (`PP`), Adverb Phrase (`ADVP`), and Adjective Phrase (`ADJP`). As for the entity type, we use the named entity labels defined in OntoNotes (Hovy et al., 2006): `CARDINAL`, `DATE`, `PERSON`, `NORP`, `GPE`, `LAW`, `PERCENT`, `ORDINAL`, `MONEY`, `WORK_OF_ART`, `FAC`, `TIME`, `QUANTITY`, `PRODUCT`, `LANGUAGE`, `ORG`, `LOC`, and `EVENT`. We add an `OTHER` for phrases that do not belong to any of them. We combine the two sets in a Cartesian product so that we obtain a label set $\mathcal{Y}$ with 95 phrase types in total. For example, the label `NP-GPE` signifies a noun phrase related to a geopolitical name (*"the United States"*), a label

VP-ORG corresponds to a verb phrase associated with an organization (*"Bring Me the Horizon"*), and a label PP-QUANTITY identifies a propositional phrase linked to a quantity (*"between 1500 to 2000 ft"*), which might be useful for numerical reasoning tasks.

Now suppose that we have an m-dimensional vector $\mathbf{u} \in \mathbb{R}^m$ and an n-dimensional vector $\mathbf{v} \in \mathbb{R}^n$ generated by the phrase and character encoder, respectively. We concatenate them and apply a softmax layer with a trainable weight $\mathbf{W} \in \mathbb{R}^{(m+n)\times|\mathcal{Y}|}$:

$$\mathbf{o}^{et} = \mathrm{softmax}((\mathbf{u}, \mathbf{v})\mathbf{W}) \tag{1}$$

Here, $\mathcal{Y}$ is the label set and $\mathbf{o}^{et} \in \mathbb{R}^{|\mathcal{Y}|}$ is the final output for predicting the entity type.

3.4 Objective and Training

Loss Function. There are two training tasks in our framework: Contrastive Learning and Phrase Type Classification. We adopt the widely-used contrastive loss (Hjelm et al., 2019; Chen et al., 2020) for training, which encourages learned representations for positive pairs to be close while pushing apart representations of negative samples. The loss function can be written as:

$$\mathcal{L}_{\mathrm{CL}} = -\log \frac{e^{\mathrm{sim}(\mathbf{h}^\top \mathbf{h}^+)/\tau}}{e^{\mathrm{sim}(\mathbf{h}^\top \mathbf{h}^+)/\tau} + \sum e^{\mathrm{sim}(\mathbf{h}^\top \mathbf{h}_i^-)/\tau}} \tag{2}$$

Here, τ is a temperature parameter that regulates the level of attention given to difficult samples, $\mathrm{sim}(\cdot)$ is a similarity function such as cosine similarity, and $(\mathbf{h}, \mathbf{h}^+)$, $(\mathbf{h}, \mathbf{h}^-)$ are positive pairs and negative pairs, respectively (assuming that all vectors are normalized). During training, we apply one data augmentation randomly to the original phrase for obtaining positive pairs while negative examples are the other samples in the mini-batch. This training process encourages the model to learn representations that are invariant against variations.

As for the task of Phrase Type Classification, we use a standard cross-entropy loss:

$$\mathcal{L}_{\mathrm{CE}} = -\sum_{i=1}^{|\mathcal{Y}|} y_i \log o_i^{et} \tag{3}$$

Finally, the overall learning objective is:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CL}} + \mathcal{L}_{\mathrm{CE}} \tag{4}$$

Training Corpus. We use Wikipedia to construct our training samples. We parse the articles with the Berkeley Neural Parser (Kitaev and Klein, 2018) and collect five lexical types of phrases (NP, VP, PP, ADVP, ADJP). We remove phrases that appear less than two times and obtain around 3.8 million phrases in total (NP: 60.1%, VP: 0.4%, PP: 26.1%, ADVP: 11.0%, ADJP: 2.4%). To obtain the entity types, we employ a Named Entity Recognition (NER) model. We use DeBERTa (He et al., 2021) fine-tuned on OntoNotes (Hovy et al., 2006). The entity type distribution is shown in Figure A1.

Hard Negatives. Conventional contrastive learning regards other samples in the same batch as negatives (in-batch negatives) (Hjelm et al., 2019; Chen et al., 2020), which is simple and effective. However, these negative samples might be easy to distinguish by a model. For example, *"The New York Times"* and *"two years after"* can be in the same batch during training, but this negative pair contributes less to the parameter optimization process. Hence, we introduce *hard negatives* into each batch, i.e., samples that have a surface form similar to the original phrase, but a different semantics – as in *"The New York Times"* and *"New York City"*. For each phrase in the training set, we first retrieve candidates that have a small edit distance with the original phrase. Next, all the candidates are encoded by the E5 text embedding. Finally, the candidates with a low cosine similarity are selected as the hard negatives. During training, a certain number of hard negatives are added to each batch.

Weight Average. We found that there is a catastrophic forgetting problem (McCloskey and Cohen, 1989) after fine-tuning, i.e., the model forgets previously learned information upon learning new information. To avoid this, we average the weights of the original and fine-tuned models, which is simple yet effective.

4 Experiments

4.1 Datasets

To evaluate our framework, we use tasks of phrase and short text in experiments. In total, there are six types of tasks, which cover both the field of data science and of natural language processing. We briefly introduce tasks and datasets used in experiments and you can see more details in the appendix A.1.

For phrase datasets, we consider five tasks:

(1) **Paraphrase Classification**. We use two paraphrase classification datasets used by PhraseBERT (Wang et al., 2021a): *PPDB* and *PPDB-filtered*. (2) **Phrase Similarity**. We use two datasets, *Turney* (Turney, 2012) and *BIRD* (Asaadi et al., 2019). (3) **Entity Retrieval**. We construct two entity retrieval datasets by using a general knowledge base *Yago* (Pellissier Tanon et al., 2020) and a biomedical terminology *UMLS* (Bodenreider, 2004), respectively. (4) **Entity Clustering**. We use the general-purpose *CoNLL 03* (Tjong Kim Sang, 2002) benchmark and the biomedical *BC5CDR* (Li et al., 2016) benchmark. (5) **Fuzzy Join**. We use the *AutoFJ* benchmark (Li et al., 2021), which contains 50 diverse fuzzy-join datasets derived from DBpedia (Lehmann et al., 2015).

For short text datasets, we consider two tasks: (1) **Sentiment Analysis**. We use a Twitter corpus [4] for this goal due to its short length. Two datasets are constructed based on this corpus: Twitter-S and Twitter-L, which contain 10,000 short Twitter sentences and 20,000 long Twitter sentences, respectively. (2) **Intent Classification**. We use the ATIS (Airline Travel Information Systems) dataset (Hemphill et al., 1990), which consists of 5400 queries with 8 intent categories. We constructed two subsets, ATIS-S and ATIS-L, based on the length of query sentences.

4.2 Implementation Details

All approaches are implemented with PyTorch (Paszke et al., 2019) and HuggingFace (Wolf et al., 2020). We use three NVIDIA Tesla V100S PCIe 32 GB for all experiments. We test two versions of PEARL, PEARL-small and PEARL-base, initialized by E5-small and E5-base (Wang et al., 2022), respectively. We then fine-tune them on our constructed phrase dataset for two epochs. The hyperparameters are selected by using grid search (see Figure 5). The batch size is 512 (the maximum capacity for a single GPU), and we use Adam (Kingma and Ba, 2015) with a learning rate of $3e-5$ for optimization. The learning rate is exponentially decayed for every 2000 steps with a rate of 0.98. The temperature τ is the default value of 0.07 and the number of hard negatives is 2 for each mini-batch. Each data augmentation method is randomly used during training. We fine-tune PEARL three times with different seeds and report the average score.

[4] https://huggingface.co/datasets/carblacac/twitter-sentiment-analysis

4.3 Competitors

We compare our approach to the following competitors: **String Distance** uses the Jaccard similarity of n-gram characters to compare two strings. **FastText** (Bojanowski et al., 2017) and **GloVe** (Pennington et al., 2014) are two popular word embedding methods, and we average word embeddings in order to obtain phrasal representations. **Sentence-BERT** (Reimers and Gurevych, 2019) fine-tuned BERT on SNLI (Bowman et al., 2015) sentence pairs. **Phrase-BERT** (Wang et al., 2021a) is a dedicated model for phrase representation fine-tuned on lexically diverse datasets. **UCTopic** (Li et al., 2022) is an unsupervised contrastive learning framework for context-aware phrase representations and topic mining. **E5** (Wang et al., 2022) is a powerful text embedding model that can transfer to a wide range of tasks. E5 offers three model sizes: E5_{small}, E5_{base}, and E5_{large}, initialized from MiniLM (Wang et al., 2021b), BERT_{base}, and BERT_{large}. We do not compare to McPhrasy (Cohen et al., 2022) because it is not publicly available.

5 Results

5.1 Overall Performance

Table 1 shows the experimental results across five phrase tasks. We first note that PEARL-base achieves the best performance on average, obtaining the best score on 6 of 9 datasets. Second, our framework brings significant improvements to the corresponding backbone language models. Specifically, PEARL-base improves E5-base by 3.7 absolute percentage points on average and the corresponding improvement of PEARL-small is 6.1 absolute percentage points. Moreover, PEARL-small with 40 million parameters outperforms other competitors, and this result validates our claim that a small model can obtain competitive results with a big model for short text representations.

Apart from these phrase tasks, we conduct experiments on short text classification to show a practical usage of our PEARL model and the results are shown in Table 2. While PEARL is able to outperform other phrase models like PhraseBERT and UCTopic, there is no statistical difference compared to other sentence models like SimCSE (*BERT-unsup*) and E5. It is worth mentioning that our model brings a benefit on very short texts (Twitter-S and ATIS-S).

Model	Size	Paraphrase Classification		Phrase Similarity		Entity Retrieval		Entity Clustering		Fuzzy Join	Avg
		PPDB	PPDB filtered	Turney	BIRD	YAGO	UMLS	CoNLL 03	BC5CDR	AutoFJ	
Length		(2.5)	(2.0)	(1.2)	(1.7)	(3.3)	(4.1)	(1.5)	(1.4)	(3.8)	(2.4)
String Distance	-	-	-	-	-	-	-	-	-	64.7	-
GloVe (2014)	-	95.5	50.6	31.5	53.1	20.9	18.8	21.2	7.8	50.6	38.9
FastText (2017)	-	94.4	61.2	59.6	58.9	16.9	14.5	3.0	0.2	53.6	40.3
Sentence-BERT (2019)	110M	94.6	66.8	50.4	62.6	21.6	23.6	25.5	48.4	57.2	50.1
Phrase-BERT (2021a)	110M	96.8	68.7	57.2	68.8	23.7	26.1	35.4	59.5	66.9	54.5
UCTopic (2022)	240M	91.2	64.6	60.2	60.2	5.2	6.9	18.3	33.3	29.5	41.6
E5-small (2022)	34M	96.0	56.8	55.9	63.1	43.3	42.0	27.6	53.7	74.8	57.0
E5-base (2022)	110M	95.4	65.6	59.4	66.3	47.3	**44.0**	32.0	69.3	76.1	61.1
PEARL-small	40M	**97.2**$_{\pm 0.1}$	69.2$_{\pm 0.7}$	56.1$_{\pm 0.1}$	69.7$_{\pm 0.1}$	48.1$_{\pm 0.1}$	43.4$_{\pm 0.2}$	**48.7**$_{\pm \mathbf{0.7}}$	61.0$_{\pm 1.1}$	74.6$_{\pm 0.1}$	63.1$_{\pm 0.2}$
PEARL-base	116M	97.1$_{\pm 0.0}$	**72.7**$_{\pm \mathbf{0.4}}$	**60.9**$_{\pm \mathbf{0.3}}$	**72.3**$_{\pm \mathbf{0.3}}$	**50.2**$_{\pm \mathbf{0.2}}$	43.6$_{\pm 0.4}$	40.9$_{\pm 0.2}$	**69.5**$_{\pm \mathbf{0.6}}$	**76.3**$_{\pm \mathbf{0.0}}$	**64.8**$_{\pm \mathbf{0.2}}$

Table 1: Evaluations of various phrase-level tasks. For the AutoFJ, we report the average accuracy across 50 datasets. The best results are shown in bold and the second best results are underlined. Since the baseline String Distance cannot produce phrase embeddings, we only report its results on the AutoFJ as a reference.

Model	Size	Sentiment Analysis		Intent Classification		Avg
		Twitter-S	Twitter-L	ATIS-S	ATIS-L	
Length		(4.5)	(9.2)	(2.7)	(12.1)	
SimCSE (2021)	110M	70.4$_{\pm 0.3}$	74.5$_{\pm 0.2}$	91.2$_{\pm 0.5}$	96.8$_{\pm 0.1}$	83.2
Phrase-BERT (2021a)	110M	71.9$_{\pm 0.1}$	77.0$_{\pm 0.2}$	50.6$_{\pm 1.4}$	79.5$_{\pm 2.7}$	69.8
UCTopic (2022)	240M	60.3$_{\pm 0.1}$	70.6$_{\pm 0.3}$	26.9$_{\pm 0.0}$	72.2$_{\pm 0.0}$	57.5
E5-small (2022)	34M	70.7$_{\pm 0.4}$	78.1$_{\pm 1.2}$	92.7$_{\pm 0.0}$	94.1$_{\pm 0.1}$	83.9
E5-base (2022)	110M	72.4$_{\pm 0.2}$	**79.5**$_{\pm 0.4}$	93.0$_{\pm 0.6}$	96.2$_{\pm 0.3}$	85.3
PEARL-small	40M	72.8$_{\pm 0.2}$	78.5$_{\pm 0.5}$	**93.7**$_{\pm 0.5}$	96.7$_{\pm 0.1}$	**85.4**
PEARL-base	116M	**73.7**$_{\pm \mathbf{0.3}}$	77.1$_{\pm 0.1}$	93.2$_{\pm 0.7}$	**97.4**$_{\pm \mathbf{0.1}}$	**85.4**

Table 2: Evaluations of text classification tasks. We run each model 10 times and report the average accuracy. "S" and "L" mean short and long, respectively. The best results are shown in bold and the second best results are underlined.

We conclude that our PEARL framework can produce high-quality representations for phrases and short texts across various tasks. If the length of input texts is very short (e.g., less than six tokens), it is beneficial to use PEARL embeddings.

5.2 Ablation Study

We vary components of PEARL to validate architectural choices. We use PEARL-small as the baseline. We fine-tune each variation of PEARL-small in the same experimental setting and test it across five phrase tasks. All results are shown in Table 3.

Entity Type Classification. If entity type classification is removed, the average performance decreases by 2.2 percentage points and drops dramatically for the entity clustering task. This validates our claim that adding phrase type information enhances representation capabilities.

Character Encoder. PEARL uses LOVE (Chen et al., 2022) to capture morphological variations of phrases. Removing LOVE causes a drop of 0.8 percentage points on average, especially for the entity-clustering task (-3.2).

Data Augmentation. PEARL uses data augmentation at three levels of granularity: character-level, token-level, and phrase-level methods. To validate the effect of each level, we stop using a particular augmentation during fine-tuning. We find that character-level augmentation is beneficial mainly for the tasks of entity clustering (-3.3) and Entity Retrieval (-0.5). Token-level augmentations create lexically diverse positive phrases, and removing these samples degrades performances across all five tasks. Phrase-level augmentation has the strongest impact on the representation capabilities of a model. Removing all augmentations results in an average drop of 2.4 percentage points.

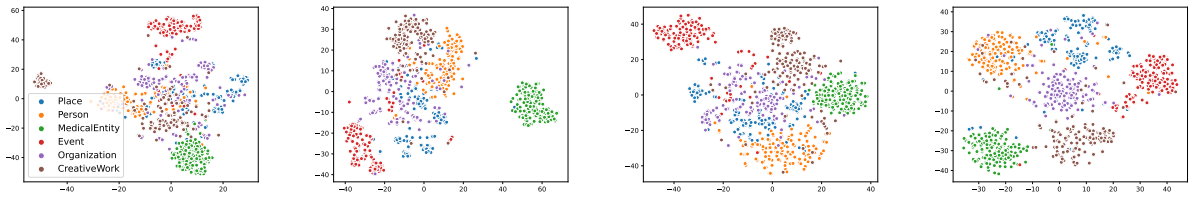
Hard Negatives. As random in-batch negatives contain relatively less information to learn, we insert a number of hard negatives into each batch. These negatives share similar surface forms with the original phrases but differ in their meanings. We find that adding hard negatives brings decent improvements (+0.8 on average), especially considering the nearly zero additional cost of this strategy.

5.3 Visualization

To demonstrate more intuitively the improved quality of phrasal representations, we visualize embeddings generated by different models. Specifically, we use six types of entities from YAGO 4 (Pellissier Tanon et al., 2020) in this experiment: `Place`, `Person`, `MeidicalEntity`, `Event`, `Organization`, `CreativeWork`. For each type, 100 entity names are randomly sampled from the entire set and we feed them into the four models for obtaining phrase embeddings. Then, we apply t-SNE to reduce them to 2 dimensions for visualization. As Figure 4 shows, PEARL can effectively cluster the same types of phrases together.

Model	Paraphrase Classification	Phrase Similarity	Entity Retrieval	Entity Clustering	Fuzzy Join	Avg
PEARL-small	$83.2_{\pm 0.4}$	$62.9_{\pm 0.1}$	$45.8_{\pm 0.2}$	$54.9_{\pm 0.9}$	$74.6_{\pm 0.1}$	$63.1_{\pm 0.2}$
- Phrase DA	**82.6**$_{\pm 0.1}$ ↓	**61.2**$_{\pm 0.4}$ ↓	**41.0**$_{\pm 0.6}$ ↓	$52.1_{\pm 1.7}$ ↓	**72.7**$_{\pm 0.3}$ ↓	**60.7**$_{\pm 0.3}$ ↓
- Entity Type	$82.9_{\pm 1.0}$ ↓	$63.7_{\pm 0.2}$ ↑	$44.3_{\pm 0.2}$ ↓	**45.7**$_{\pm 0.5}$ ↓	$74.9_{\pm 0.1}$ ↑	$60.9_{\pm 0.1}$ ↓
- Token DA	$82.7_{\pm 0.4}$ ↓	$62.8_{\pm 0.4}$ ↓	$44.6_{\pm 0.7}$ ↓	$51.4_{\pm 2.0}$ ↓	$73.9_{\pm 0.3}$ ↓	$61.9_{\pm 0.5}$ ↓
- Hard Negatives	$83.2_{\pm 0.4}$ ↕	$63.2_{\pm 0.4}$ ↑	$45.1_{\pm 0.7}$ ↓	$52.0_{\pm 0.4}$ ↓	$73.9_{\pm 0.2}$ ↓	$62.3_{\pm 0.2}$ ↓
- Character Encoder	$82.8_{\pm 0.4}$ ↓	$63.3_{\pm 0.4}$ ↑	$45.8_{\pm 0.4}$ ↕	$51.7_{\pm 0.7}$ ↓	$73.9_{\pm 0.2}$ ↓	$62.3_{\pm 0.2}$ ↓
- Character DA	$82.9_{\pm 0.3}$ ↓	$63.5_{\pm 0.3}$ ↑	$45.3_{\pm 0.6}$ ↓	$51.6_{\pm 1.4}$ ↓	$74.5_{\pm 0.4}$ ↓	$62.4_{\pm 0.5}$ ↓

Table 3: Ablation study. DA means Data Augmentation. The biggest drop is in bold.

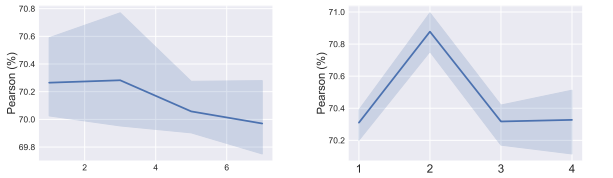


(a) Phrase-BERT (110M) (b) UCTopic (240M) (c) E5-base (110M) (d) PEARL-small (40M)

Figure 4: t-SNE visualizations of phrase embeddings generated by different models. We randomly selected 100 samples for each entity type from YAGO 4 (`Place`, `Person`, `MeidicalEntity`, `Event`, `Organization`, `CreativeWork`). Markers with the same color are supposed to be grouped together.

Model	BERT	RoBERTa	ALBERT	SpanBERT	LUKE
Original	39.4	33.2	33.6	29.6	31.9
+ PEARL	57.1	53.4	52.5	50.6	52.7
Δ	17.7 ↑	20.2 ↑	18.9 ↑	21.0 ↑	20.8 ↑

Table 4: The performances of language models after using our framework. The results are the average score across five phrase tasks.



(a) Learning Rate (e-5) (b) Number of Hard Negatives

Figure 5: Hyperparameter selection on BIRD dataset.

5.4 Generalizability of Our Framework

We now demonstrate that PEARL can enhance the phrase representations of various language models. Beyond E5, we test five other language models: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020), SpanBERT (Joshi et al., 2020), and LUKE (Yamada et al., 2020). We first check the original performance of each language model across five phrase tasks and then use PEARL to fine-tune them by following the same experimental setting as before (but using 30% of training samples to save time). Table 4 shows that PEARL consistently obtains significant enhancements, showing that our method can be generalized to various models.

5.5 Hyperparameter Selection

Figure 5 shows the performances on the BIRD datasets by varying learning rates and numbers of hard negatives. We observe that a learning rate of 3e-5 and using 2 hard negatives in each batch can yield better phrase embeddings.

6 Conclusion

In this study, we have presented PEARL, a novel contrastive learning framework for more powerful phrase representations. PEARL incorporates phrase type information and morphological features, and thereby captures better the nuances of phrases. Furthermore, PEARL enriches training samples with distinct granularities of data augmentations. Our empirical results show that it improves phrase embeddings for a wide range of tasks, from paraphrase classification to entity retrieval, useful in applications across NLP and data engineering. Adding character-level support to language models appears crucial to success on short texts. Indeed, these provide much less context than full paragraphs and thus it is important to go beyond the tokens of the original language model that mainly capture word stems.

Limitations

One potential limitation is that our PEARL may not provide significant advantages when dealing with long sentences. Since PEARL is specifically dedicated to modeling morphological variations of short texts by using context-free input, current PEARL models do not capture long-distance contextual semantics very well, which can limit their performances and benefits on long texts.

Acknowledgements

This work was partially funded by projects NoRDF (ANR-20-CHIA-0012-01) and LearnI (ANR-20-CHIA-0026).

References

Shima Asaadi, Saif Mohammad, and Svetlana Kiritchenko. 2019. Big BiRD: A large, fine-grained, bigram relatedness dataset for examining semantic composition. In *Proc. of NAACL-HLT*.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, (suppl_1).

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proc. of EMNLP*.

Lihu Chen, Gael Varoquaux, and Fabian Suchanek. 2022. Imputing out-of-vocabulary embeddings with LOVE makes LanguageModels robust with little cost. In *Proc. of ACL*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proc. of ICML*, Proceedings of Machine Learning Research.

Peter Christen. 2011. A survey of indexing techniques for scalable record linkage and deduplication. *IEEE transactions on knowledge and data engineering*, (9).

Amir Cohen, Hila Gonen, Ori Shapira, Ran Levy, and Yoav Goldberg. 2022. McPhraSy: Multi-context phrase similarity and clustering. In *Findings of the Association for Computational Linguistics: EMNLP 2022*.

Prithiviraj Damodaran. 2021. Parrot: Paraphrase generation for nlu.

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

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proc. of EMNLP*.

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

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Karl Moritz Hermann and Phil Blunsom. 2013. The role of syntax in vector space models of compositional semantics. In *Proc. of ACL*.

R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *Proc. of ICLR*.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, (3).

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proc. of ACL*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proc. of ICLR*.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning dense representations of phrases at scale. In *Proc. of ACL*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, (2).

Jiacheng Li, Jingbo Shang, and Julian McAuley. 2022. UCTopic: Unsupervised contrastive learning for phrase representations and topic mining. In *Proc. of ACL*.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*.

Peng Li, Xiang Cheng, Xu Chu, Yeye He, and Surajit Chaudhuri. 2021. Auto-fuzzyjoin: auto-program fuzzy similarity joins without labeled examples. In *Proceedings of the 2021 International Conference on Management of Data*.

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*.

James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 14. Oakland, CA, USA.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*.

George A. Miller. 1992. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*.

Lorena Otero-Cerdeira, Francisco J Rodríguez-Martínez, and Alma Gómez-Rodríguez. 2015. Ontology matching: A literature review. *Expert Systems with Applications*, (2).

George Papadakis, Dimitrios Skoutas, Emmanouil Thanos, and Themis Palpanas. 2020. Blocking and filtering techniques for entity resolution: A survey. *ACM Computing Surveys (CSUR)*, (2).

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proc. of ACL*.

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. 2020. Yago 4: A reason-able knowledge base. In *The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17*. Springer.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proc. of EMNLP*.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proc. of ACL*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proc. of EMNLP*.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proc. of EMNLP*.

Richard Socher, Christopher D Manning, and Andrew Y Ng. 2010. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 deep learning and unsupervised feature learning workshop*. Vancouver.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Peter D Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of artificial intelligence research*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, (10).

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *ArXiv preprint*.

Shufan Wang, Laure Thompson, and Mohit Iyyer. 2021a. Phrase-BERT: Improved phrase embeddings from BERT with an application to corpus exploration. In *Proc. of EMNLP*.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021b. MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proc. of EMNLP*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proc. of EMNLP*.

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers. In *Proc. of EMNLP*.

Minghe Yu, Guoliang Li, Dong Deng, and Jianhua Feng. 2016. String similarity search and join: a survey. *Frontiers of Computer Science*.

Mo Yu and Mark Dredze. 2015. Learning composition models for phrase embeddings. *Transactions of the Association for Computational Linguistics*.

Xiang Zhao, Weixin Zeng, Jiuyang Tang, Wei Wang, and Fabian M Suchanek. 2020. An experimental study of state-of-the-art entity alignment approaches. *TKDE*, (6).

Zhihao Zhou, Lifu Huang, and Heng Ji. 2017. Learning phrase embeddings from paraphrases with GRUs. In *Proceedings of the First Workshop on Curation and Applications of Parallel and Comparable Corpora*.

A Appendix

A.1 Details of Datasets

A.1.1 Phrase Datasets

Paraphrase Classification (PC) judges whether two phrases convey the same meaning. We use two paraphrase classification datasets used by Phrase-BERT (Wang et al., 2021a): **PPDB** and **PPDB-filtered**. PPDB is constructed from PPDB 2.0 (Pavlick et al., 2015), which includes 23,364 phrase pairs by sampling examples from PPDB-small with a high score, and negative examples are randomly selected from the dataset. PPDB-filtered contains more challenging samples, which are obtained by removing phrase pairs with lexical overlap cues. In total, there are 19,416 phrase pairs. We follow the setting of previous work for experiments (Wang et al., 2021a), where a simple classifier layer (multilayer perceptron with a ReLu activation) is added on top of the concatenated embeddings of a phrase pair. We measure accuracy.

Phrase Similarity (PS) aims to calculate the semantic similarity for phrase pairs. We use two datasets, **Turney** (Turney, 2012) and **BIRD** (Asaadi et al., 2019). Turney evaluates bigram compositionality. A model is supposed to select the most similar unigram from five candidates given a bigram input. The dataset has 2180 samples and the metric is accuracy. BIRD is a fine-grained and human-annotated bigram relatedness dataset, which contains 3345 English term pairs. Each pair of phrases has a relatedness score between 0 and 1, and the metric for this dataset is the Pearson correlation coefficient.

Entity Retrieval (ER) aims to retrieve a standard entity from a reference knowledge base given a textual mention of that entity. We consider a particularly challenging form of the task, where the mention is given without any context, and the reference knowledge base provides only the canonical name of the entity. For example, given the mention *"NYTimes"*, the goal is to determine the canonical entity *"The New York Times"* in Wikidata. We construct two entity retrieval datasets by using a general knowledge base **Yago** (Pellissier Tanon et al., 2020) and a biomedical terminology **UMLS** (Bodenreider, 2004), respectively. Both Yago and UMLS offer alternate names for an entity, and we randomly selected 10K of these alternate names as mentioned. The canonical names of the entities serve as the reference dictionary and there are no duplicate names in the dictionary. The dictionary size of Yago and UMLS is 572K and 750K, respectively. To accelerate the inference, we use Faiss (Johnson et al., 2019) with all competing systems to do an approximate search. The metric here is top-1 accuracy.

Entity Clustering (EC) tests whether the phrase embeddings can be grouped together according to their semantic categories. We use the general-purpose **CoNLL 03** (Tjong Kim Sang, 2002) benchmark and the biomedical **BC5CDR** (Li et al., 2016) benchmark. CoNLL 03 consists of 3,453 sentences with entities, and the three entity types are used in the experiment: `Person`, `Location`, and `Organization`. BC5CDR has 7,095 sentences with two types of entities: `Disease` and `Chemical`. We apply KMeans (MacQueen et al., 1967) to the embeddings generated by a phrase representation model and use the NMI (normalized mutual information) metric.

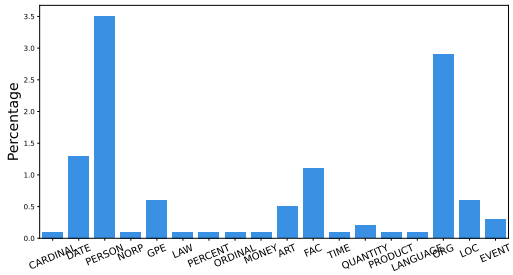


Figure A1: Distributions of each entity type (without the OTHER tag, with 88.5%).

Fuzzy Join (FJ) is an important database operator widely used in practice (also known as fuzzy-match), which matches record pairs from two tables. We use the **AutoFJ** benchmark (Li et al., 2021), which contains 50 diverse fuzzy-join datasets derived from DBpedia (Lehmann et al., 2015). It aims to match entity names that have changed over time (e.g., *"2012 Wisconsin Badgers football team"* and *"2012 Wisconsin Badgers football season"*). In this experiment, we use the left table names as reference tables and the right table names as input tables. We report the average accuracy across all datasets.

All experiments except paraphrase classification are conducted without fine-tuning.

A.1.2 Short Text Datasets

Sentiment Analysis (SA) analyzes texts to determine whether the emotion is positive or negative. We use a Twitter corpus for this goal due to its short length. Two datasets are constructed based on this corpus: Twitter-S and Twitter-L, which contain 10,000 short Twitter sentences and 20,000 long Twitter sentences, respectively.

Intent Classification (IC) identifies customer's intents from text queries. We use ATIS (Airline Travel Information Systems) dataset (Hemphill et al., 1990), which consists of 5400 queries with 8 intent categories. We constructed two subsets, ATIS-S and ATIS-L, based on the length of query sentences.

For the two short text classification tasks, we add a classifier layer (multilayer perceptron with a ReLu activation) on top of the text embeddings and report the average accuracy across 10 times run.