# PESCO: Prompt-enhanced Self Contrastive Learning for Zero-shot Text Classification

Yau-Shian Wang Ta-Chung Chi Ruohong Zhang Yiming Yang

Carnegie Mellon University

king6101@gmail.com {tachungc,ruohongz}@andrew.cmu.edu

yiming@cs.cmu.edu

#### Abstract

We present PESCO, a novel contrastive learning framework that substantially improves the performance of zero-shot text classification. We formulate text classification as a neural text matching problem where each document is treated as a query, and the system learns the mapping from each query to the relevant class labels by (1) adding prompts to enhance label matching, and (2) using retrieved labels to enrich the training set in a self-training loop of contrastive learning. PESCO achieves state-of-the-art performance on four benchmark text classification datasets. On DBpedia, we achieve 98.5% accuracy without any labeled data, which is close to the fully-supervised result. Extensive experiments and analyses show all the components of PESCO are necessary for improving the performance of zero-shot text classification.

# 1 Introduction

Text classification is the task of assigning relevant category labels to each input document. It is an important problem in machine learning research with a wide spectrum of applications, including sentiment analysis (Pang et al., 2002; Maas et al., 2011; Socher et al., 2013; Tang et al., 2014), question answering (Rajpurkar et al., 2016, 2018), and intent classification (Tur et al., 2010), etc. Recently, deep neural networks have obtained remarkable improvements in text classification, including CNNs (Kim, 2014; Zhang et al., 2015), RNNs (Tang et al., 2015; Yang et al., 2016), Transformers (Vaswani et al., 2017), and more, thanks to the successful modeling of contextualized representations.

Despite the remarkable progress, training wellperforming neural classifiers still requires a large amount of human-labeled documents, which is costly and time-consuming, especially for new application domains. This stimulates the recent trend of exploring self-supervised pre-training neural models on text classification tasks. In particular, pre-trained language models (PTLMs) (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019) clearly stand out from other methods owing to the pre-training on large-scale unlabeled data. Nevertheless, how to adapt PTLMs to downstream tasks with less supervision remains an open question for the research community, inviting new ideas to explore.

Prompt-based learning (Brown et al., 2020; Shin et al., 2020; Liu et al., 2021; Li and Liang, 2021; Gao et al., 2021a) has been actively studied to better adapt PTLMs to downstream tasks with the goal of reducing human annotation effort. For example, PET (Schick and Schütze, 2020) is a prompt-based method for few-shot text classification. It formulates the task as a *Cloze Test*, where a PTLM is used to predict the output label(s) by completing a prompt concatenated right after an input document. For example, the sentiment of a product review is highly likely to be positive if a PTLM fills the word "good" into the following input:

[Review] | It is a \_ product.

This example shows that prompt-based learning could unleash the potential power of a PTLM by constructing the input format of a downstream task in a way that closely resembles the PTLM pretraining objective, which is masked language modeling (MLM) in this case.

Motivated by the recent success of prompt-based learning, we propose PESCO, a novel self-training framework for zero-shot classification that uses prompts to enhance performance. The self-training consists of two iterative steps, pseudo-label prediction and model update. To make label descriptions more informative, we first put label descriptions into some predefined prompts and call the enhanced descriptions label-prompts. As depicted in Figure 1, to predict the pseudo-label of a document, PESCO formulates text classification as a neural matching task. A pre-trained text encoder maps both documents and label-prompts into a shared embedding space. A label whose embedding is closest to the document is predicted as the pseudo-label.

To effectively update the text encoder with pseudo-labels, we propose the Prompt-enhanced Label-aware Cloze Test (PLCT), a contrastive learning framework for self-training. The text encoder is trained to match a document and the text relevant to its pseudo-label. The relevant texts include pseudo-label prompts and the key sentences from the documents assigned to the same pseudolabel. The key sentence of each document is the sentence most related to its pseudo-label.

In our experiments, we show that the iterative self-training consistently improves the classification performance compared to the same model without self-training and that our proposed approach substantially outperforms other strong zero-shot classification baselines. On some datasets, the zeroshot results are even on par with a fully supervised baseline. On the Dbpedia dataset, in particular, PESCO achieves 98.5% accuracy without any labeled data.

In summary, the contributions of this paper are twofold:

- We explore text classification in a neural matching formulation enhanced by prompts. We demonstrate that even without any finetuning on the text encoder, this straightforward formulation is an effective method for zeroshot text classification.
- 2. The potential of contrastive learning for selftraining has not been explored. We show that this is a promising direction for self-training and can achieve state-of-the-art performance on zero-shot text classification.

### 2 Related Work

#### 2.1 Contrastive Learning

Contrastive learning (CL) (Chopra et al., 2005; Hadsell et al., 2006) is a metric learning method that aims to pull closer similar inputs in the embedding space. Recently, the most popular and efficient methods for CL involve batch contrastive learning (He et al., 2019; Chen et al., 2020), which put similar inputs (positive pairs) and dissimilar inputs (negative pairs) in the same batch, simultaneously minimizing the distance of representations from positive pairs, while maximizing the distance of negative pairs.



Figure 1: In this example, there are three classes, whose label descriptions are "sports", "business", and "world" respectively. We convert the descriptions into label-prompts by placing them into a template. The model predicts a label whose label-prompt embedding is the most similar to the document embedding.

The key to CL is how to construct positive samples. Based on downstream applications, there are various ways to formulate the positive pairs. In self-supervised pre-training, the positive pairs are usually formulated by data augmentation. That is, different versions of a distorted sample are treated as a positive pair. In supervised contrastive learning (Khosla et al., 2020), the examples belonging to the same class are viewed as a positive pair.

In NLP, CL is usually used as an additional selfsupervised pre-training to PTLMs because the sentence embeddings from PTLMs without fine-tuning are not ready to be used in downstream tasks (Li et al., 2020). SimCSE (Gao et al., 2021b) employs dropout as minimal data augmentation and obtains state-of-the-art unsupervised sentence representations. In supervised SimCSE, the sentences with entailment relation are viewed as a positive pair. Other approaches for data augmentation include sentence reformulation (Wu et al., 2020), back translation (Fang et al., 2020), dual encoder (Carlsson et al., 2021), language model corruption (Meng et al., 2021), and translation pairs (Wang et al., 2022).

In addition, CL is a commonly used training algorithm for neural text retrieval (Xiong et al., 2021). Inverse cloze test (ICT) (Lee et al., 2019) is the most commonly used contrastive pre-training task for retrieval that predicts a randomly selected sentence from the rest of the texts. It is also possible to construct positive pairs by leveraging the document structures (Chang et al., 2020).

# 2.2 Self-training and Zero-Shot Text Classification

**Self-training** Self-training (Yarowsky, 1995; Nigam and Ghani, 2000; Lee, 2013; Xie et al.,

2020) is a widely used approach for semisupervised learning and can have additive improvement to pre-training in both computer vision (Zoph et al., 2020) and NLP (Du et al., 2021). The paradigm of self-training is first using a pre-trained base model as "teacher" to generate pseudo-labels on unlabeled data. The pseudo-label is then used to train a "student" model. The teacher-student training is performed iteratively until convergence.

Zero-shot Text Classification Zero-shot classification aims to classify text using only label names without human annotation. Self-training has demonstrated impressive performance on fewshot (Mukherjee and Hassan Awadallah, 2020) and zero-shot text classification. Unlike a few-shot setting which can use supervised information to obtain a base model, in zero-shot text classification, obtaining a base model is non-trivial. LOT-Class (Meng et al., 2020) leverages PTLMs to augment label descriptions with semantically related words and then find category-indicative words among these related words to label documents. They generalize the performance to the documents without category-indicative words via self-training. iPET (Schick and Schütze, 2020) formulates text classification as a cloze test to help PTLMs understand the task. They design several types of prompts for each dataset, and each type of prompt trains an individual teacher model to annotate documents using self-training. A student model aggregates the knowledge from the teachers via knowledge distillation. In this work, we propose a novel self-training method for zero-shot text classification that integrates self-supervised pre-training into self-training in a contrastive learning framework.

## **3** Zero-shot Classification as Matching

In our zero-shot setting, there are N unlabeled documents  $X = \{x_1, x_2, \dots, x_N\}$  and a set of label descriptions  $C = \{c_1, c_2, \dots, c_L\}$ , where L denotes the number of classes. We aim to learn a scoring function g(x, c) so that relevant document and label description pairs can have higher scores. A label whose label description has the highest score is selected as model prediction:

$$\hat{y} = \arg\max_{j} g(x, c_j), \tag{1}$$

Inspired by the recent success of pre-trained sentence encoder (Gao et al., 2021b; Chuang et al., 2022) which has shown impressive performance on matching relevant texts, we explore using pretrained encoders as  $g(x, c_j)$ . Specifically, as illustrated in Figure 1, we formulate zero-shot text classification as a neural text matching problem. Both document and label descriptions are encoded into dense vectors by a shared encoder. The matching score can be obtained by measuring cosine similarity between dense vectors.

However, label descriptions are usually a few words rather than a sentence with full semantics, which makes PTLMs unable to fully understand the meaning of the labels. To tackle this, query reformulation (Nogueira and Cho, 2017; Petroni et al., 2020) is a commonly used technique in retrieval to enhance the semantics of a query. This technique can be further incorporated with promptbased learning (Schick and Schütze, 2020), which has shown that adding prompts to a text helps PTLMs understand classification tasks. We use a prompt function  $p(\cdot)$  to convert a label description c into a prompt by placing label descriptions into pre-defined templates. We design T templates for each dataset, and the scoring function is:

$$g(x,c) = \frac{1}{T} \sum_{i=1}^{T} sim(f_{\theta}(x), f_{\theta}(p^{i}(c))), \quad (2)$$

where  $f_{\theta}(\cdot)$  is a text encoder with parameters  $\theta$  that maps an input text to a dense embedding, and  $sim(\cdot)$  is a similarity function. For the rest of our paper, we use cosine similarity as  $sim(\cdot)$ . For simplicity, in the rest of the article, we use  $p_j$  to refer  $p^i(c_j)$ , which is the "label-prompt" of label j with i randomly sampled from  $\{1, \dots, T\}$ .

## 4 PESCO

PESCO is a simple but effective self-training framework for zero-shot text classification. Algorithm 1 gives an overview of PESCO. In our iterative selftraining loop, we first use a pre-trained sentence encoder  $f_{\theta}$  to generate pseudo-labels (i.e. predicted labels) by the matching process described in Section 3. We then use the pseudo-labels to update  $f_{\theta}$  by Prompt-enhanced Label-aware Cloze Test (PLCT), which leverages pseudo-labels to construct positive training pairs. We continue the selftraining process by iteratively generating pseudolabels and updating the model using the PLCT objective function.



Figure 2: The framework of the PLCT. (A) Suppose the pseudo-label  $\hat{y}_1$  for  $x_1$  is 1. We select  $s_1^2$  as the key sentence  $k_1$  for the document  $x_1$  because the embedding of  $s_1^2$  is the most similar to the embedding of label-prompt  $p_1$ .  $\hat{x}_1$  is the augmented version of  $x_1$ , which removes  $s_1^2$  from  $x_1$ . (B) We use k and  $\hat{x}$  from part (A) to construct an example batch of PLCT with batch size B = 3. Similar to self-supervised training, we use  $\hat{x}_1$  to retrieve  $k_1$  because they are from the same document. We use  $\hat{x}_1$  to retrieve  $k_2$  because  $x_1$  and  $x_2$  have the same pseudo-label. We also use  $x_1$  to retrieve the its pseudo-label-prompt  $p_1$ . (C) We separate PLCT into LCT and PCL losses.

#### 4.1 Prompt-enhanced Label-aware Cloze Test

We propose Prompt-enhanced Label-aware Cloze Test (PLCT) to update our model using pseudolabels. As shown in Figure 2, PLCT consists of two losses, Label-aware Cloze Test (LCT) loss and Prompt Contrastive Loss (PCL). To compute LCT, for each document, we first select a key sentence from the document that is most relevant to its pseudo label. In LCT, given a document, the positive texts are the key sentences from the documents belonging to the same pseudo-label. For PCL, the positive texts for a document are its pseudo-label prompt (i.e. the label-prompt of a pseudo-label). We combine these two losses by putting the positive texts of LCT and PCL into the same batch of a contrastive loss.

#### 4.1.1 Label-aware Cloze Test

LCT is inspired by Inverse Cloze Test (Lee et al., 2019) which is a widely used self-supervised pretraining task for neural text retrieval. It uses a randomly selected sentence from a document to match the remaining texts. In a document, as some sentences don't contain useful information, using a randomly selected sentence for training is not an optimal choice. Instead, we use pseudo-label to select the key sentences. Note that we use "Cloze Test" without "Inverse" because we use the remaining long texts to match its relevant short sentences, which can be viewed as label descriptions.

As illustrated in Figure 2-(A), given an input document  $x_i = \{s_i^1, s_i^2, \dots, s_i^n\}$  consists of *n* sentences and its predicted pseudo label  $\hat{y}_i$ , its key

sentence  $k_i$  is  $s_j$ , where:

$$j = \arg\max_{n} g(s_i^n, p_{\hat{y}_i}). \tag{3}$$

Here,  $g(\cdot)$  is the scoring function in Eq.(1). As key sentence  $k_i$  is more relevant to the pseudolabel than any other sentences in  $x_i$ , optimizing this objective is similar to minimize the distance between a document and its pseudo-label in embedding space, so  $k_i$  can be viewed as an augmented version of the pseudo-label prompt. Predicting the augmented version can have additional training signal than simply predicting pseudo-label prompt. We provide a real example of  $\hat{x}$  and k in Table. 1 and more examples can be found in the Appendix Table 8.

Since key sentences are highly correlated to corresponding pseudo-label prompts, given a document, it should not only match its key sentence but also key sentences in documents assigned to the same pseudo-label as shown in Figure 2 (C)-1. We use the supervised contrastive loss (Khosla et al., 2020) to optimize LCT, which extends the Sim-CLR (Chen et al., 2020) to allow multiple positive keys for a query in a supervised setting. Specifically, let  $I = \{1, \dots, B\}$  be the set of the indices of the texts in a batch, where *B* denotes the batch size. The LCT loss  $\mathcal{L}_{LCT}$  is written as:

$$\sum_{i \in I} \frac{-1}{|K(i)|} \sum_{\hat{k} \in K(i)} \log \frac{e^{sim(f_{\theta}(\hat{x}_i), f_{\theta}(k))/\gamma}}{\sum_{j \in I} e^{sim(f_{\theta}(\hat{x}_i), f_{\theta}(k_j))/\gamma}}.$$
(4)

Here,  $K(i) \equiv \{k_j, \forall j \in I : \hat{y}_j = \hat{y}_i\}$  denotes the keys belonging to the same pseudo class  $\hat{y}_i$ , and  $\gamma$ 

denotes a temperature commonly-used in CL. To prevent trivial training signal, the input document is  $\hat{x}_i = x_i \setminus \{k_i\}$  rather than  $x_i$ , where the key sentence  $k_i$  is removed.

#### 4.1.2 Prompt Contrastive Loss

As the update target of self-training is to maximize the similarity between  $x_i$  and its pseudo-labelprompt  $p_{\hat{y}_i}$  in embedding space, we use the prompt contrastive loss (PCL)  $\mathcal{L}_{PCL}$  to directly maximize the similarity:

$$\mathcal{L}_{PCL} = -\sum_{i \in I} \log \frac{e^{sim(f_{\theta}(\hat{x}_i), f_{\theta}(p_{\hat{y}_i}))/\gamma}}{\sum_{c \in C} e^{sim(f_{\theta}(\hat{x}_i), f_{\theta}(p(c)))/\gamma}}.$$
(5)

Depicted in Figure 2 (C)-2, this loss predicts  $\hat{y}_i$  from  $\hat{x}_i$ .

#### 4.2 Combining LCT and PCL

Naturally, to combine LCT and PCL, the simplest way is to use  $\mathcal{L}_{PCL} + \mathcal{L}_{LCT}$  as the final training loss. However, we found that minimizing this loss has limited improvement over minimizing  $\mathcal{L}_{LCT}$  or  $\mathcal{L}_{PCL}$  alone. As depicted in Figure 2 (B), we come up with a more effective approach that puts the positive texts from these two losses into the same batch. By doing so, pseudo keys k and pseudo prompt p can serve as mutually challenging negative samples, thus enhancing the representative power through more difficult contrastive tasks. In our experiment, this simple solution significantly improves the performance.

Specifically, we use  $\hat{x}_i$  as a query to retrieve (1) the key  $k_i$  from the same text  $x_i$ , (2) K(i), the keys belonging to the same pseudo class  $\hat{y}_i$ , and (3) the positive pseudo-label-prompt  $p_{\hat{y}_i}$ . The PLCT loss  $\mathcal{L}_{PLCT}$  is written as:

$$\sum_{i \in I} \frac{-1}{|A(i)|} \sum_{a \in A(i)} \log \frac{e^{sim(f_{\theta}(\hat{x}_i), f_{\theta}(a))/\gamma}}{\sum_{m \in M} e^{sim(f_{\theta}(\hat{x}_i), f_{\theta}(m))/\gamma}}$$
(6)

Here,  $A(i) \equiv K(i) \cup \{p_{\hat{y}_i}\}$  is the set of positive texts in the mini-batch for  $x_i, M \equiv \{k_j, \forall j \in I\} \cup \{p_c, \forall c \in C\}$  denotes the set of all the candidate keys.

Interestingly,  $\hat{x}_i$  can be viewed as a challenging data augmentation of  $x_i$  for predicting pseudolabel prompt because it removes the most salient sentence from  $x_i$ . A model can make a prediction simply based on one salient sentence, neglecting the information of remainder. This data augmentation method forces the model to capture additional information.

## Algorithm 1 PESCO

**Require:** Unlabeled texts X, label descriptions C. **Initialization:** A pre-trained sentence encoder  $f_{\theta}(\cdot)$ .

### **Repeat until convergence:**

- 1. Use  $f_{\theta}(\cdot)$  to generate hard pseudo-labels  $\hat{y}$  with Eq.(1) for all unlabeled texts without data augmentation.
- Sample T<sub>t</sub> training pairs (x, ŷ) from step 1 based on the pseudo-label predicted probability. Use these pairs to update the θ of f<sub>θ</sub>(·) that minimizes the L<sub>PLCT</sub> in eq 6.
- 3. With a more powerful  $f_{\theta}(\cdot)$ , go back to step 1.

**Output:**  $f_{\theta}(\cdot)$ 

#### 4.3 Self-training

Algorithm 1 describes PECOS self-training loop. Our self-training algorithm is a simplified version of noisy student training (Xie et al., 2020) that a single model alternately serves as a student and a teacher. The key idea of noisy student training is that the teacher uses clean data without data augmentation to generate pseudo-labels, while the student learns to predict the pseudo-label on augmented data.

We first use pre-trained sentence encoder to initialize  $f_{\theta}(\cdot)$ . Then, in step 1,  $f_{\theta}(\cdot)$  serves as a teacher to generate pseudo-labels from clean data x as described in Section 3. In step 2,  $f_{\theta}(\cdot)$  serves as a student that learns to increase the probability of predicting pseudo-labels by minimizing  $\mathcal{L}_{PLCT}$ . Step 2 is a noisy student training because the model takes  $\hat{x}$  as input rather than clean x. The selftraining repeats step 1 and step 2 until convergence. We use  $f_{\theta}(\cdot)$  from the last iteration as our final model.

In the algorithm, we set  $T_t = d \cdot T_{t-1}$  that gradually increases T until a threshold T'. The probability of sampling a pseudo training pair is proportional to the normalized scores outputed by the score function, so a more confident pseudo training pair is more likely to be sampled. When sampling pseudo training pairs, we found that it is important

Label Description	x	k
Family and Relationship	how do you know if you're in love? is it possible to know for sure? in my experience you just know. it's a long term feeling of always wanting to share each new experience with the other person in order to make them happy, to laugh or to know what they think about it. it's jonesing to call even though you just got off an hour long phone call with them. it's knowing that being with them makes you a better person. it's all of the above and much more.	how do you know if you're in love?

Table 1: An example of the document  $\hat{x}$  and the selected pseudo positive keys k in Yahoo Answers. In this example, k is very related to label description.

Dataset	Class Number	Test Examples
AG News	4	7,600
DBPedia	14	70,000
Yahoo Answers	10	60,000
Amazon	2	400,000

Table 2: Dataset statistics.

to keep the ratio of all the labels balanced. If a class doesn't have enough instances to be sampled, then we upsample the class to keep it balanced.

### 5 Experiments

## 5.1 Experimental Setting

**Implementation Details** Inspired by Yin et al. (2019) who formulate zero-shot text classification as entailment prediction, we choose the version of SimCSE (Gao et al., 2021b) pre-trained on natural language inference (NLI) task <sup>1</sup> as our text encoder for all datasets. Our experiments have shown that sentence encoder fine-tuned on NLI performs better on zero-shot classification tasks. We use the representation outputted by the last layer as our sentence representation.

Following supervised contrastive learning (Khosla et al., 2020), the value of  $\gamma$  in all equations is set to be 0.07. For the value of d in the self-training section, we set it to be 2 because we want the model to annotate unlabeled data slowly. The details of other hyperparameters in the Appendix B.

**Datasets** We conduct experiments on various text classification datasets: (1)**AG News**: topic classification on news article. (2)**DBpedia**: Ontology classification on selected classes from DBpedia. (3)**Yahoo Answers**: question type classification. (4)**Amazon**: binary sentiment classification on Amazon product review. The statistics of these

dataset are listed in Table 2.

We provide the label descriptions in Table 3. The label descriptions of Yahoo Answers and AG news are mainly from the original dataset, and the label description of DBpedia is mainly from LOT-Class (Meng et al., 2020).

## 5.2 Effect of Using Prompts

We investigate whether supplementing the label description with the prompt can help the model better understand the meaning of the label, and thus improve the performance. In Table 3, we provide the label descriptions and the prompts we use. For each dataset, we manually design two prompts, where the '[desc]' in the templates is the label description. For example, given a label description "Health", the prompting function converts it into either "It is about Health" or "Category: Health".

Our experiments showed that the choice of prompts doesn't affect performance much as long as reasonable prompts are given. For example, in AG news, without self-training, the accuracy of using "Category: <label> news", "This is about <label> news", and "<label> news" are 76.4, 76.0, and 78.0 respectively. Furthermore, our scoring function, as described in Eq.(2), combines the scores of different prompts, which further reduces the gap. The performance gap among different prompts is less than 2% without self-training and less than 1% after self-training.

In Table 4, we analyze the effect of using prompts on SimCSE without self-training. By comparing [1] with [2], we find that using prompts for retrieval improves the performance on most of the datasets, especially on AG News. We find that without the word "news", the model can not understand the meaning of the class only with the description "world". Using the prompt-enhanced SimCSE [2] as the initial base model provides a better start for self-training. However, comparing with the performance gap of [1] and [2] in Table 4, we observed that the gap between [6] and [7] becomes smaller,

<sup>&</sup>lt;sup>1</sup>We choose the model named "sup-simcse-bert-baseuncased" at https://github.com/princeton-nlp/ SimCSE.

Datasets	Label Descriptions	Prompts
AG news	(1)World (2)Sports (3)Business (4)Technology and Science	(1)Category: [desc] news. (2)[desc] news.
DBpedia	(1)company (2)school and university (3) artist (4)athlete (5)politics (6)means of transportation (7)building (8)river and mountain and lake (9)vil- lage (10)animal species (11)plant and tree (12)al- bum (13)film (14)novel and publication and book	(1)Category: [desc]. (2)It is about [desc].
Yahoo Answers	(1)Society and Culture (2)Science and Mathemat- ics (3)Health (4)Education and Reference (5)Com- puters and Internet (6)Sports (7) Business and Fi- nance (8)Entertainment and Music (9)Family and Relationships (10)Politics and Government	(1)Category: [desc]. (2)It is about [desc].
Amazon-review-P	bad, good	(1)It is a [desc] product. (2)In summary, the product is [desc]

Table 3: The label descriptions and their prompts. [desc] in the templates denotes the label descriptions.

Id	Self-train	Methods	AG News	DBpedia	Yahoo Answers	Amazon
[1]	No	SimCSE w/o prompt	69.7	73.8	55.2	88.3
[2]	No	SimCSE w/ prompt	76.3	76.0	56.5	88.3
[3]	No	PET	79.4	75.2	56.4	87.1
[4]	Yes	iPET	86.0	85.2	68.2	95.2
[5]	Yes	LOTClass	86.4	91.1	-	91.6
[6]	Yes	PESCO w/o prompt	87.1	96.0	69.9	95.1
[7]	Yes	PESCO	89.6	98.5	71.1	95.2
[8]	_	Supervised	94.2	99.3	77.3	97.1

Table 4: Test-set accuracy of zero-shot text classification methods. The Self-train column indicates whether a method performs self-training on unlabeled data.

which indicates that the effect of using prompts decreases after self-training.

## 5.3 Zero-shot Text Classification

In Table 4, we compare our results against two stateof-the-art zero-shot text classification baselines, LOTClass (Meng et al., 2020) and iPET (Schick and Schütze, 2020). We select these two methods as our baselines because they both employ selftraining for zero-shot classification. In [1], [2], and [3], they do not employ self-training on unlabeled data, so the Self-train column is "No". In [7], we report the best results over 5 runs on PESCO single model performance without an ensemble. We also report the average, maximum, and minimum accuracy over 5 runs in Appendix Table 6. In [8], to see the gap between zero-shot and fully-supervised settings, we train a typical BERT (Devlin et al., 2019) classifier on a labeled training set. We jointly finetune BERT and a linear classifier on top of BERT [CLS] output layer.

**Effect of Self-training** First, by comparing [7] against [2] in Table 4, we find that the proposed

self-training framework significantly improves the performance by more than 10% on average. On DB-pedia, self-training improves performance substantially by 20%, and it even achieves 98.5% accuracy. This demonstrates that self-training is an effective method to enhance performance after general pre-training, closing the gap between fully supervised training.

**Comparison against LOTClass** Comparing [7] PESCO against [5] LOTClass, PESCO significantly improves the zero-shot text classification performance on all datasets. LOTClass leverages PTLMs to find the category-indicative words which are semantically related to label descriptions. The documents containing category-indicative words are classified as the corresponding category. Our method uses a pre-trained sentence encoder to define the relevance between document and category, which is more effective and requires less human heuristics.

**Comparison against iPET** Our main baseline is [4] iPET, which uses [3] PET as a base model to generate initial pseudo-labels followed by a se-

Id	Methods	AG News	DBpedia	Yahoo Answers	Amazon
[1]	PESCO	89.6	98.5	71.1	95.2
[2]	PESCO - R	87.0	97.1	69.1	95.0
[3]	LCT	88.0	89.1	69.6	94.3
[4]	LCT - R	80.7	86.9	68.6	93.3
[5]	PCL	87.8	89.4	68.7	95.1
[6]	LCT+PCL	88.2	97.0	69.8	95.2
[7]	PESCO w/o aug	87.8	96.7	68.6	93.5

Table 5: Contrastive losses of different methods. The methods end with "-R" means their pseudo positive key sentences are randomly selected instead of picking the most salient sentence.

ries of self-training steps. We find that our base model [2] achieves similar performance with [3] on all datasets except Ag News, on which ours lags behind by 3%. The lesson here is that using text retrieval as a means of text classification gives a similar performance to that using cloze tests. Next, our full model [7] is also better than [4] iPET on three datasets while achieving similar performance on the Amazon dataset, demonstrating the effectiveness of our method. Also, we notice that PET requires a massive model ensemble (e.g. 15 models) to achieve the reported accuracy. We run their code with a PvP ensemble without using various random seeds for ensembling. Even with this simplified setting, iPET still needs far more disk space (1.4 GB vs 26 GB) and more training time than us in that we do not need to train various models for model ensembling in each self-training step.

Note that It is not feasible to test our method using Roberta-base/large because language models without SimCSE finetuning poorly capture the semantic meaning of texts in cosine similarity space and cannot be used for retrieval. On the other hand, simCSE is finetuned for sentence embeddings, making language models lose text generation ability. Because iPET and LOTClass require language models to generate tokens, using SimCSE-Roberta for iPET or LOTClass is also not feasible.

#### 5.4 Ablation Study and Analysis

**Comparison of different contrastive losses** The results of different contrastive learning losses are shown in Table 5. In the table, LCT means we only use  $\mathcal{L}_{LCT}$  in Eq.(4) to train our model, PCL means we use  $\mathcal{L}_{PCL}$ , and LCT+PCL means we sum the  $\mathcal{L}_{LCT}$  and  $\mathcal{L}_{PCL}$  as our loss function rather than using PLCT loss which puts keys and label-prompts in the same batch. The methods end with "-R" means the pseudo positive sentences k are randomly selected from the documents instead of picking the most salient sentences.

In LCT, although it doesn't explicitly minimize the distance between an input document and its predicted pseudo-label-prompt, optimizing this loss still obtains performance similar to PLC. This implies the selected key sentences can serve as augmented version of label-prompts.

Furthermore, we analyze the difference in the performance between using randomly selected sentences and the most salient sentences. By comparing [1] and [2], and [3] and [4], we can see that the model has a significant performance drop in predicting randomly selected sentences. This demonstrates the importance of choosing a salient sentence as the training target.

Finally, to demonstrate the effectiveness of putting pseudo-label-prompts and key sentences in the same batch, we compare [1] against [6]. [1] yields better performance than [6], which implies using this more challenging contrastive task allows the model to learn more general representations.

Effect of Data Augmentation In Table 5, [7] PESCO w/o aug means we use  $x_i$  as a query to retrieve its positive examples A(i) instead of using  $\hat{x}_i$  as a query. Comparing [1] and [7], removing the most salient sentence from a document is an effective data augmentation method that can greatly improve performance. This is consistent with previous literature (Xie et al., 2020) that updating student models with noisy data is important in selftraining.

## 6 Conclusion

This paper presents a novel approach to zero-shot text classification, which significantly improves the SOTA results on four benchmark datasets by formulating the classification task as a prompt-enhanced retrieval problem and by combining the strengths of pre-trained language models and contrastive learning over pseudo-labeled data in a self-training loop. Our experiments in comparison with representative baselines and ablation analysis show evidence for the effectiveness of the proposed approach.

# 7 Limitations

The main limitation of our method is that it heavily depends on the quality of the label description. If a label description does not precisely describe the meaning of the label, our method cannot work. For some classification tasks such as microaggression detection, their labels have abstract meaning that is difficult to be understood by pre-trained language models. Similarly, our method cannot work on the domain that is not covered by the pre-training corpora of language models, such as the medical domain.

Another limitation of our method is that PLCT loss cannot handle short texts. If a text consists of only one sentence, PLCT loss will no longer work because LCT requires a document to be more than one sentence. In this case, PCL loss can still be used for self-training.

## References

- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In Advances in Neural Information Processing Systems, volume 33, pages 1877–1901. Curran Associates, Inc.
- Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2021. Semantic re-tuning with contrastive tension. In *International Conference on Learning Representations*.
- Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. In *International Conference on Learning Representations*.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.
- S. Chopra, R. Hadsell, and Y. LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In 2005 IEEE Computer Society

Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 1, pages 539–546 vol. 1.

- Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based contrastive learning for sentence embeddings. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4207–4218, Seattle, United States. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Veselin Stoyanov, and Alexis Conneau. 2021. Self-training improves pre-training for natural language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5408–5418, Online. Association for Computational Linguistics.
- Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2021a. Making pre-trained language models better few-shot learners. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3816–3830, Online. Association for Computational Linguistics.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, page 1735–1742, USA. IEEE Computer Society.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2019. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*.

- Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.
- Dong-Hyun Lee. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshops*.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6086–6096. Association for Computational Linguistics.
- Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 9119–9130, Online. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4582– 4597, Online. Association for Computational Linguistics.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing.
- 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.
- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.
- Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. 2021. Cocolm: Correcting and contrasting text sequences for language model pretraining. In *Advances in Neural Information Processing Systems*, volume 34.

- Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. In *Proceedings of the* 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 9006–9017, Online. Association for Computational Linguistics.
- Subhabrata Mukherjee and Ahmed Hassan Awadallah. 2020. Uncertainty-aware self-training for few-shot text classification. In Advances in Neural Information Processing Systems (NeurIPS 2020), Online.
- Kamal Nigam and Rayid Ghani. 2000. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, CIKM '00, page 86–93, New York, NY, USA. Association for Computing Machinery.
- Rodrigo Nogueira and Kyunghyun Cho. 2017. Taskoriented query reformulation with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 574–583, Copenhagen, Denmark. Association for Computational Linguistics.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the* 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), pages 79–86. Association for Computational Linguistics.
- Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few-shot text classification and natural language inference. *Computing Research Repository*, arXiv:2001.07676.
- Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4222–4235, Online. Association for Computational Linguistics.

- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.
- Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, Lisbon, Portugal. Association for Computational Linguistics.
- Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for Twitter sentiment classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1555–1565, Baltimore, Maryland. Association for Computational Linguistics.
- Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in atis? In 2010 IEEE Spoken Language Technology Workshop, pages 19– 24.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Yau-Shian Wang, Ashley Wu, and Graham Neubig. 2022. English contrastive learning can learn universal cross-lingualsentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *ArXiv*, abs/2012.15466.
- Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. 2020. Self-training with noisy student improves imagenet classification. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 10684–10695.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc.

- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1480–1489, San Diego, California. Association for Computational Linguistics.
- David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In 33rd Annual Meeting of the Association for Computational Linguistics, pages 189–196, Cambridge, Massachusetts, USA. Association for Computational Linguistics.
- Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. 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 3914–3923. Association for Computational Linguistics.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.
- Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. In Advances in Neural Information Processing Systems, pages 3833–3845.

	AG News	DBpedia	Yahoo	Amazon
avg	88.7	96.9	70.5	94.3
max	89.6	98.5	71.1	95.2
min	87.7	96.1	70.0	93.9

Table 6: Average/minimum/maximum accuracy over 5 runs.

# A Discussion

Text Classification as neural text retrieval Formulating text classification as neural retrieval is straightforward but not widely explored by previous work. In this work, we show that this formulation can also obtain good performance with a well-pre-trained sentence encoder. The benefit of this formulation over cloze test is that we don't need to restrict the label description to only one word. PET requires a carefully selected word (verbalizer) to represent each class. If a classification task has hundreds or even more than thousands of categories, it is not feasible to manually select a word to represent each class. Furthermore, if the meaning of a category in a classification task is too abstract or complex, we cannot simply represent it with a single word. Our formulation allows the model to describe categories using sentences or even short texts and maybe a better choice for more challenging classification tasks.

**Contrastive Learning for Self-training** The effect of contrastive learning for self-training is not well-studied by previous work. Contrastive learning obtains impressive results on unsupervised representation learning. In a supervised setting, it is also robust to noisy labels and noisy data, and it also shows impressive performance on a few-shot classification. Considering these good properties of contrastive learning, we believe contrastive learning is a promising direction for self-training and propose PESCO to explore its potential on zero-shot text classification.

# **B** Hyperparameters

As indicated by previous work (Chen et al., 2020), using a larger batch size generally yields better performance because it includes more negative samples. We analyze how different batch size influences the performance of PESCO in Figure 3. We found that PESCO is not very sensitive to batch size. Using a smaller batch size only reduces the accuracy by less than 2%. Also, we observe that our



Figure 3: The effect of different batch sizes.



Figure 4: Training epoch versus validation set accuracy on AG News dataset.

algorithm converges after 1000 steps (1 epoch) of training, and additional training steps only slightly increase the performance. In other datasets, our algorithm also converges after 1 training epoch.

We list the hyperparameters of our model in Table 7. We use AdamW as our optimizer. The T'is the threshold mentioned in Section 4.3, we set it proportional to N, where N is the total number of unlabeled data in the corresponding dataset. We find that the number of training epoch only slightly influence the final performance that usually influences the accuracy by less than 1%. In Figure 4, we plot the training epoch versus validation set accuracy. Although we train 5 epochs on the AG news to obtain the best result, the model actually converges in the early training stage. Similar training curves can be observed in all the datasets.

	AG News	DBpedia	Yahoo Answers	Amazon
Learning rate	1e-5	1e-5	5e-6	5e-6
Document length	156	128	192	128
Batch size	32	32	32	32
Epsilon	1e-6	1e-8	1e-8	1e-8
T'	0.2N	0.5N	0.1N	0.1N
Epoch	5	5	2	1

Table 7: Hyperparameters.

Label Description	x	k
Family and Relationship	where is the best place to look for love? it might be easy to use the internet- there are many good matching web sites that can help	where is the best place to look for love?
Entertainment and Music	what is the best place to get guitar lessons in the south bay area? look- ing for a great instructor and relatively affordable price. i have no experience but have a desire to learn. it's really according to what you are looking for. certain teachers specialize in acoustic vs. electric (for example). your best bet is to place a request on a service such as click for lessons that will show you several teacher bios and let you decide for yourself.	what is the best place to get guitar lessons in the south bay area?
Business and Finance	does anyone know a good apartment rental agency around washington dc? i've had personal experience with arch- stone apartments and summit (just bought by camden) apartments in the past two years. while neither one is stel- lar, both were acceptable. both of these were in the northern virginia area - bed- room communities for d.c. best of luck apartment hunting! the housing market around here is absolutely insane.	does anyone know a good apartment rental agency around washington dc?
Sports	why are there 5 rings in the olympics symbol? what does it represent? i heard few theories about it but not sure what is the correct one the 5 rings were intro- duced at the the 1920 games in antwerp games. the rings included at least one color from the flag of every participat- ing country.	why are there 5 rings in the olympics symbol?

Table 8: More examples of the distorted document  $\hat{x}$  and the selected pseudo positive keys k in Yahoo Answers. It happens that k seems to be the most important sentence of the texts, so their semantics are closest to label descriptions.

## ACL 2023 Responsible NLP Checklist

## A For every submission:

- A1. Did you describe the limitations of your work? 7
- □ A2. Did you discuss any potential risks of your work? *Not applicable. Left blank.*
- $\checkmark$  A3. Do the abstract and introduction summarize the paper's main claims? 0,1
- A4. Have you used AI writing assistants when working on this paper? *Left blank.*

## **B ☑** Did you use or create scientific artifacts?

3,4,5

- B1. Did you cite the creators of artifacts you used?
  2
- □ B2. Did you discuss the license or terms for use and / or distribution of any artifacts? *Not applicable. Left blank.*
- □ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)? *Not applicable. Left blank.*
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it? Left blank.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.? *Not applicable. Left blank.*
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.

# C ☑ Did you run computational experiments?

5

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

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

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
   5
- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run? 5
- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
   5
- **D** Z Did you use human annotators (e.g., crowdworkers) or research with human participants? *Left blank.* 
  - □ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.? *No response.*
  - □ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
     *No response*.
  - □ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used? No response.
  - □ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board? *No response.*
  - D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
     *No response.*