Zero-Shot Text Classification via Self-Supervised Tuning

Chaoqun Liu[*12] **Wenxuan Zhang**[†2] **Guizhen Chen**[*12] **Xiaobao Wu**[1]
Anh Tuan Luu[1] **Chip Hong Chang**[1] **Lidong Bing**[2]
[1]Nanyang Technological University, Singapore, [2]DAMO Academy, Alibaba Group
{chaoqun.liu,guizhen.chen,saike.zwx,l.bing}@alibaba-inc.com
{xiaobao002,echchang,anhtuan.luu}@ntu.edu.sg

Abstract

Existing solutions to zero-shot text classification either conduct prompting with pre-trained language models, which is sensitive to the choices of templates, or rely on large-scale annotated data of relevant tasks for meta-tuning. In this work, we propose a new paradigm based on self-supervised learning to solve zero-shot text classification tasks by tuning the language models with unlabeled data, called self-supervised tuning. By exploring the inherent structure of free texts, we propose a new learning objective called first sentence prediction to bridge the gap between unlabeled data and text classification tasks. After tuning the model to learn to predict the first sentence in a paragraph based on the rest, the model is able to conduct zero-shot inference on unseen tasks such as topic classification and sentiment analysis. Experimental results show that our model outperforms the state-of-the-art baselines on 7 out of 10 tasks. Moreover, the analysis reveals that our model is less sensitive to the prompt design. Our code and pre-trained models are publicly available at https://github.com/DAMO-NLP-SG/SSTuning.

1 Introduction

Recent advances in pre-trained language models (PLMs) have brought enormous performance improvements in a large variety of NLP tasks (Radford and Narasimhan, 2018; Devlin et al., 2019). These paradigm shifts towards leveraging generic features learnt by PLMs are driven by the high data cost required for learning each new NLP task afresh. One promising learning method that echoes this paradigm shift is zero-shot text classification, which predicts text labels on unseen tasks. Zero-shot text classification has attracted considerable research attention in recent years (Wei et al., 2022; Sanh et al., 2022; Yang et al., 2022), as labeled data is no longer a necessity for relearning new feature representations for untrained specific tasks.

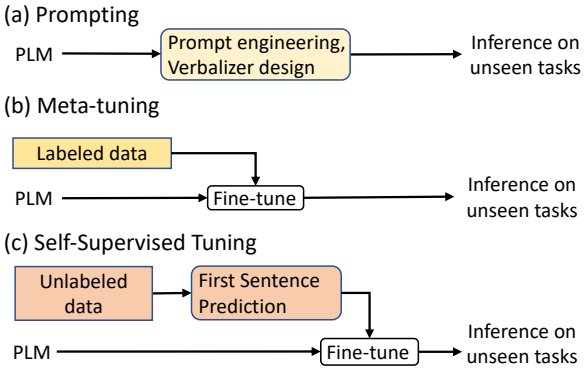


Figure 1: Zero-shot learning approaches: (a) prompting, (b) meta-tuning, and (c) our proposed self-supervised tuning method.

Existing studies on zero-shot text classification can be briefly classified into two types, as shown in Figure 1. The first type is prompting, which uses PLMs to predict labels with designed templates and verbalizers (Figure 1 (a)). This can be achieved by leveraging the generation capability of large language models (Brown et al., 2020; Chowdhery et al., 2022), or reformulating text classification task as a mask-filling task (Schick and Schütze, 2021; Schick and Schütze, 2021). Likewise, generation-based methods (Meng et al., 2022; Ye et al., 2022) and mining-based methods (van de Kar et al., 2022) also rely on prompting to generate or filter noisy labeled samples, which are used for further fine-tuning. The second type is meta-tuning which fine-tunes a PLM on a collection of labeled data of related tasks before conducting inference on unseen tasks (Figure 1 (b)). By reformulating the annotated data into instruction templates (Wei et al., 2022; Sanh et al., 2022), question-answer pairs (Khashabi et al., 2020; Zhong et al., 2021), multiple-choice questions (Yang et al., 2022) or entailment pairs (Yin et al., 2019; Ding et al., 2022;

[*]Chaoqun Liu and Guizhen Chen are under the Joint PhD Program between Alibaba and Nanyang Technological University.

[†]Wenxuan Zhang is the corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 1743–1761
July 9-14, 2023

Du et al., 2023), and fine-tuning on them, PLMs perform well on unseen tasks.

Despite the achieved performance, existing methods have several limitations. Prompting has shown to be sensitive to the choice of patterns and verbalizers (van de Kar et al., 2022). This makes it difficult to design different templates specifically for each task. In addition, generation-based and mining-based methods require fine-tuning PLMs for each downstream task, which is inefficient for deployment. On the other hand, meta-tuning relies on labeled data of relevant tasks or in specific formats to facilitate the learning of desired patterns. The requirement for such large-scale annotated data narrows its application scope.

To address the above issues, we propose to leverage self-supervised learning (SSL) for zero-shot text classification tasks. SSL has been widely used during the pre-training stage of PLMs to alleviate the need for large-scale human annotations (Devlin et al., 2019; Lan et al., 2020) by exploiting the intrinsic structure of free texts. Therefore, with a suitable SSL objective, the model is able to capture certain patterns with the auto-constructed training data and can be applied to a wide range of downstream tasks in a zero-shot manner without specific designs. To our best knowledge, this is the first work to exploit SSL at the tuning stage for zero-shot classification, which we refer to as self-supervised tuning (SSTuning).

The biggest challenge of applying SSTuning to zero-shot text classification tasks is to design a proper learning objective that can effectively construct large-scale training samples without manual annotations. Intuitively, the core of the text classification task can be treated as associating the most suitable label to the text, given all possible options. Motivated by this observation, we propose a new learning objective named first sentence prediction (FSP) for the SSTuning framework to capture such patterns. In general, the first sentence tends to summarize the main idea of a paragraph. Therefore, predicting the first sentence with the rest of the paragraph encourages the model to learn the matching relation between a text and its main idea ("label"). To generate training samples, we use the first sentence in the paragraph as the positive option and the rest as text. The first sentences in other paragraphs are used as negative options. Specifically, if negative options are from the same article as the positive option, they are regarded as hard negatives since the sentences in the same article normally have some similarities, such as describing the same topic. Hard negatives may force the model to learn the semantics of the text instead of simply matching the keywords to complete the task.

In the inference phase, we convert all possible labels of a sample into options, which can be done in two simple ways: 1) use original label names; 2) convert labels using the templates (like "This text is about [label name]"). Then the text and options are combined to create the final input. The tuned model can thus retrieve the most relevant option as the predicted label of the text. Since the tuned model has seen a large number of samples and various first sentences as options, which has a higher chance to consist of similar options to the ones at the inference phase, its performance is less sensitive to verbalizer design. In this way, our SSTuning enables efficient deployment of PLM for classifying texts of unseen classes on-the-fly without requiring further tuning with labeled data or unlabeled in-domain data.

Our main contributions are:

- We propose a new learning paradigm called self-supervised tuning (SSTuning) to solve zero-shot text classification tasks. A simple yet effective learning objective named first sentence prediction is designed to bridge the gap between unlabeled data and text classification tasks.
- We conduct extensive experiments on 10 zero-shot text classification datasets. The results show that SSTuning outperforms all previous methods on overall accuracy in both topic classification tasks and sentiment analysis tasks. Our analysis further demonstrates that our model is less sensitive to prompt design.

2 Proposed Method

In this section, we discuss our proposed framework, SSTuning, and provide details for our dataset preparation process using the idea of first sentence prediction (FSP), the tuning phase, and the zero-shot inference phase.

2.1 First Sentence Prediction

Text classification can be regarded as selecting the most relevant label for the text, given all possible labels. Based on such observation, we propose the

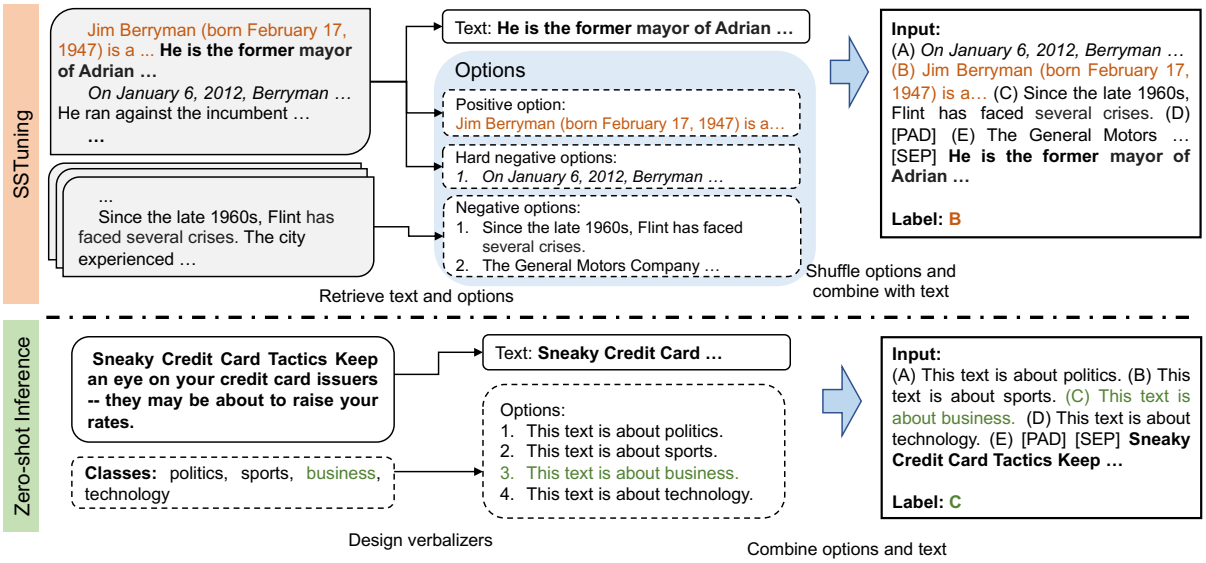


Figure 2: Data construction for SSTuning (top) and zero-shot inference (bottom). The number of labels N_{model} is set as 5 here. The SSTuning example is from Wikipedia and the inference example is from AG News dataset.

FSP task to create datasets for our SSTuning by mimicking the same structure.

We design the FSP task by considering both the nature of the unlabeled corpus and the input/output format of classification tasks. In this subsection, we describe in detail how to construct the tuning and validation sets from the unlabeled corpus. Figure 2 shows the core procedures for our dataset generation.

Data filtering. We first filter data to select appropriate paragraphs for tuning (more details are shown in A.1). Removing meaningless sentences ensures data quality, which helps improve the performance of the model.

First sentence as the positive option. We consider an article A_n that contains M paragraphs, i.e., $A_n = [P_1^n, P_2^n, ...P_M^n]$, and suppose paragraph P_m^n has K sentences $[S_1^{n,m}, S_2^{n,m}, ..., S_K^{n,m}]$, the positive option $O_c^{n,m}$ and the text $x^{n,m}$ are:

$$O_c^{n,m} = S_1^{n,m} \quad (1)$$

$$x^{n,m} = [S_2^{n,m}, ..., S_K^{n,m}] \quad (2)$$

As shown in Figure 2, we can retrieve the first sentence *"Jim Berryman (born February 17, 1947) is a ... "* as the positive option and the rest of the paragraph *"He is the former mayor of Adrian ..."* as the text for the first paragraph in the article.

Negative sampling. After getting the positive option, we randomly sample J "first sentences" from other paragraphs $[S_1^{n_1,m_1}, S_1^{n_2,m_2}, ...S_1^{n_J,m_J}]$ as negative options, where J is a random number that satisfies $1 \leq J \leq N_{\text{maxLabel}} - 1$. We let N_{maxLabel} denote the maximum number of labels that are first sentences, which is pre-defined to ensure the total number of tokens for options is not too long. It is less or equal to N_{model}, where N_{model} is the number of labels for the model output layer. Having a random number of negative options bridges the gap between tuning and zero-shot inference since the number of classes for evaluation datasets may vary from 2 to N_{model}.

Hard negatives. During negative sampling, if the negative options and the positive option are from the same article, we call the options hard negatives. Inspired by the successful application of hard negatives in Gao et al. (2021b), we purposely add more hard negatives to enhance the model performance. Sometimes, when we read articles, we notice that the same words appear in the first sentence and the rest of the paragraph. As shown in Figure 2, we can use the word *"Berryman"* to quickly find the corresponding first sentence for the text. However, if we add the hard negative *"On January 6, 2012, Berryman ..."*, the model has to understand the true semantics to choose the positive option.

Option padding. We pad the options with the special `"[PAD]"` token to make the input format consistent between the tuning phase and the inference phase. Specifically, if the total number of options after negative sampling is $(J + 1) < N_{\text{model}}$, we will add $(N_{\text{model}} - J - 1)$ `[PAD]` options. Thus

the final list of options is:

$$O^{n,m} = [S_1^{n,m}, S_1^{n_1,m_1}, S_1^{n_2,m_2}, ...S_1^{n_J,m_J}, O^1_{\text{PAD}}, O^2_{\text{PAD}}, ...O^{N_{\text{model}}-J-1}_{\text{PAD}}] \quad (3)$$

Generating final text and label. We shuffle the option list because the position of a positive option is random in the evaluation datasets. After shuffling, we assume the option list is:

$$O^{n,m}_{\text{shuffle}} = [O_0, O_1, ...O_{N_{\text{model}}-1}], \quad (4)$$

where the positive option $O^{n,m}_c = O_j$. Then the label for this sample is:

$$L^{n,m} = j. \quad (5)$$

The final input text is the concatenation of the above components:

$$x^{n,m}_{inp} = [\text{CLS}]\{(T_i)\ O_i\}^{N_{\text{model}}-1}_{i=0}[\text{SEP}]x^{n,m}[\text{SEP}] \quad (6)$$

where T_i is the i-th item from the index indicator list T (e.g. $[A, B, C...]$), `[CLS]` is the classification token, and `[SEP]` is the seperator token used by Devlin et al. (2019).

Thus the final text-label pair ($x^{n,m}_{inp}$, $L^{n,m}$) is the generated sample. We can repeat this process to generate a large number of samples as the tuning set. The validation set can also be generated in the same way. Note that if we select a corpus that only contains paragraphs instead of articles, we can treat each paragraph as an article, and no hard negatives are generated.

2.2 Tuning Phase

2.2.1 Network Architecture

We employ BERT-like pre-trained masked language models (PMLM) as the backbone, such as RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2020). Following Devlin et al. (2019), we add an output layer for classification. Such models have both bidirectional encoding capabilities and simplicity. Generative models are not necessary since we only need to predict the index of the correct option. We do not make any changes to the backbone so that the method can be easily adapted to different backbones. In order to cover all test datasets, we config the number of labels for the output layer as the maximum number of classes for all test datasets, denoted by N_{model}.

2.2.2 Learning Objective

Traditional text classification with PMLMs like BERT maps each classification layer output to a class. Such a design requires a dedicated output layer for each dataset as they have different classes. Instead, our learning object for FSP with the same network is to predict the index of the positive option. In this way, we can use the output layer for both tuning and inference and for various kinds of datasets.

As shown in Figure 2, we concatenate the labels and the text as input. The outputs are the indices (0, 1, 2..., which correspond to A, B, C), which are the same as traditional classification datasets. We use a cross-entropy loss for tuning the model.

2.3 Zero-Shot Inference Phase

During the zero-shot inference phase, we can infer directly by converting the input of the sample to the same format as that in the tuning phase.

2.3.1 Input Formulation

As shown in Figure 2, the zero-shot inputs are formulated similarly as the tuning phase, except 1) instead of using first sentences as options, we convert the class names to options. Actually, we can simply use the original labels or some simple templates like "This text is about [label name]." for the conversion, thus little to no effort is needed. 2) No shuffling is needed. Since the converted input and output during SSTuning and zero-shot phases are the same, no further adjustment of the model is required.

2.3.2 Constrained Prediction

Since the dimension of the output logits (N_{model}) may be different from the number of classes in a dataset (N_L), the predictions may be out of range (e.g. the model may output 3 for a dataset with 2 classes). To solve this issue, we simply make predictions based on the first N_L logits:

$$P = \text{argmax}(\text{logits}[0 : N_L]) \quad (7)$$

where P is the index for the positive option.

3 Experiment Setup

3.1 SSTuning Datasets

We choose English Wikipedia and Amazon review dataset (2018) (Ni et al., 2019) for SSTuning. The Wikipedia corpus has more than 6.2M articles[1] by

[1] https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia

the end of 2021, while Amazon Review Data has around 233.1M reviews[2]. Wikipedia articles typically use formal expressions and Amazon reviews contain informal user-written texts, together covering different genres of text.

For English Wikipedia, we collect articles up to March 1st, 2022. To balance the dataset, we select up to 5 paragraphs in each article. The generated dataset has 13.5M samples. For the Amazon review dataset, we only use the review text to create our SSTuning dataset, ignoring other information such as summary and vote. The Amazon review dataset has 29 categories. To keep the model from being dominated by a certain category, we select up to 500k samples from each category. In the end, we collected 11.9M samples.

To have a balanced dataset, we sample 2.56M from the Wikipedia dataset and 2.56M from the Amazon review dataset, forming a total of 5.12M samples as the tuning dataset. In addition, we sampled 32k from each of the two datasets, forming a validation set consisting of 64k samples.

3.2 Evaluation Datasets

We evaluate the models on 4 topic classification (TC) tasks, including Yahoo Topics (yah) (Zhang et al., 2015), AG News (agn) (Zhang et al., 2015), DBPedia (dbp) (Zhang et al., 2015) and 20newsgroup (20n) (Lang, 1995), and 6 sentiment analysis (SA) tasks, including SST-2 (sst2) (Socher et al., 2013), IMDb (imd) (Maas et al., 2011), Yelp (ylp) (Zhang et al., 2015), MR (mr) (Pang and Lee, 2005) and Amazon (amz) (Zhang et al., 2015), which are binary classification tasks, and SST-5 (sst5) (Socher et al., 2013), a fine-grained 5-class SA task. Detailed data statistics for each testing dataset are presented in Table 6 in Appendix A.

Following the baselines (Yang et al., 2022; van de Kar et al., 2022; Gera et al., 2022), we report the accuracy on the test set when available, falling back to the original validation set for SST-2.

3.3 Baselines

We choose the following baselines for comparison after considering their relevancy, impact, checkpoint availability, and model sizes:

- Textual entailment (TE) (Yin et al., 2019): Following Gera et al. (2022), we download the off-the-shelf models trained on MNLI and use the default hypothesis template *"This example is []."* for evaluation.
- TE-Wiki (Ding et al., 2022): This model is also trained with entailment methods but with a dataset constructed from Wikipedia.
- Prompting-based method (Schick and Schütze, 2021): We compare with the results using multiple verbalizers reported in (van de Kar et al., 2022).
- Mining-based (van de Kar et al., 2022): The method has three steps, which are *mine*, *filter* and *fine-tune*. We compare with the results reported.
- UniMC (Yang et al., 2022): We download the released checkpoint and test the model without question prompts since the reported results on text classification tasks are better on average.

We followed the setups and verbalizers of the original works as much as possible. If the original work does not have verbalizers for a dataset, we will use the same or comparable verbalizers as ours, as shown in Table 7.

3.4 Implementation Details

To test the performance of the proposed method on different model sizes and architectures, we tune three versions of models, which are based on $\text{RoBERTa}_{\text{base}}$, $\text{RoBERTa}_{\text{large}}$ (Liu et al., 2019), and $\text{ALBERT}_{\text{xxlarge}}$ (V2) (Lan et al., 2020), denoted as SSTuning-base, SSTuning-large, SSTuning-ALBERT, respectively. We set the maximum token length as 512 and only run one epoch. We repeat all the experiments 5 times with different seeds by default. The experiments on SSTuning-base and SSTuning-large are run on 8 NVIDIA V100 GPUs and the experiments on SSTuning-ALBERT are run on 4 NVIDIA A100 GPUs.

The hyperparameters for fine-tuning and SSTuning are shown in Table 8. We set the batch size based on the constraint of the hardware and do a simple hyperparameter search for the learning rate. We do not add hard negatives for the Amazon review dataset since the reviews are not in the format of articles. We also tried to use the negative options from the same product category as hard negatives but did not find any meaningful improvement. We set N_{model} as 20 and N_{maxLabel} as 10 after simple experiment.

[2] https://nijianmo.github.io/amazon/

4 Results and Analysis

4.1 Main Results

The main results are shown in Table 1. We have the following observations: 1) Our method SSTuning-ALBERT achieves new state-of-the-art results on 7 out of 10 datasets, and significantly reduces the gap between fine-tuning and zero-shot methods compared to UniMC (from 10.6 to 7.2) , showing the superiority of our proposed method. 2) With the same backbone, SSTuning-ALBERT outperforms UniMC by 3.4% on average. Note that different from UniMC, we do not utilize any labeled data to conduct meta-tuning, but purely rely on auto-constructed data for self-supervised tuning, which not only has a much large scale of data but also has more abundant options (first sentences). 3) Comparing methods based on $\text{RoBERTa}_{\text{base}}$, $\text{RoBERTa}_{\text{large}}$ and $\text{BART}_{\text{large}}$, our SSTuning-large and SSTuning-base are the two best-performing models on average. We also observe that SSTuning-large outperforms UniMC, despite the latter possessing a stronger backbone. 4) Our models do not perform very well on SST-5, which is a fine-grained sentiment analysis task. Maybe we can generate more fine-grained options from the unlabeled corpus to improve performance on such tasks. We leave it as a future work.

4.2 Ablation Study

4.2.1 Ablation on Tuning Datasets

We utilize both the Amazon review dataset and English Wikipedia during the tuning stage. To evaluate their effectiveness, we conduct ablation studies to create two model variants that are only trained on one dataset. We set the number of samples for each case to 5.12M for a fair comparison. As shown in Table 2, both datasets contribute to the final performance, thus discarding any one leads to a performance drop. It is interesting that tuning with Amazon review data performs the same as tuning with Wikipedia on topic classification tasks. This is unexpected since Wikipedia is more related to topic classification tasks intuitively. We anticipate the reason is that the backbone models have already been pre-trained with Wikipedia, thus further tuning with it does not bring significant advantages.

4.2.2 Alternative Tuning Objectives

We have proposed first sentence prediction (FSP) as the tuning objective to equip the model learning to associate the label and text in the inference stage. We consider some alternative objectives here for comparison: 1) last sentence prediction (LSP), which treats the last sentence as the positive option for the rest of the paragraph; 2) next sentence selection (NSS)[3], which treats the first sentence in a consecutive sentence pair as text and the next as the positive option; 3) random sentence prediction (RSP), which randomly pick a sentence in a paragraph as the positive option and treat the rest as text. The comparison between the four settings is shown in Table 3. We find that FSP performs the best, especially for topic classification tasks. Among the alternatives, utilizing LSP as the tuning objective leads to the best performance, which is expected since the last sentence in a paragraph usually also contains the central idea, sharing a similar function as the first sentence. Unlike topic classification tasks, the four settings perform similarly on sentiment analysis tasks. The possible reason is that each sentence in a paragraph shares the same sentiment.

4.3 Analysis

4.3.1 Impact of Verbalizer designs

During self-supervised tuning, the model saw a large number of first sentences as options, which may contain similar options to the unseen tasks, thus it may have better generalization capabilities. To test how robust the model is to the verbalizer changes compared with UniMC, we design 10 sets of verbalizers for SST-2 and IMDb, covering various scenarios: 1) verbalizers with a single word; 2) verbalizers with different punctuation marks; 3) combinations of single verbalizers; 4) different format for different classes. For a fair comparison, we only use one of our checkpoints and compare it with the UniMC checkpoint released. The results are shown in Table 4. We find that SSTuning-ALBERT performs better on average and is more stable. For the most challenging case, which is *"Terrible!"* and *"I like the movie! It is wonderful!"*, SSTuning-ALBERT outperforms UniMC by 20.4 points for SST-2 and 17 points for IMDb.

4.3.2 Classification Mechanism

To investigate how our models make correct decisions, we did a case study on a movie review example. As shown in Figure 3, we used SSTuning-base (number of labels configured as 2) to classify

[3]Note that we use NSS here to distinguish from NSP (next sentence prediction) used by Devlin et al. (2019).

	Backbone	Labeled	Topic Classification				Sentiment Analysis						Avg
			yah	agn	dbp	20n	sst2	imd	ylp	mr	amz	sst5	
Fine-tuning❖	$\text{RoBERTa}_{\text{large}}$	-	77.1	95.5	99.2	75.3	95.9	96.4	98.3	91.3	97.2	59.9	88.6
TE-Wiki	$\text{BERT}_{\text{base}}$	✓	56.5	79.4	90.4	53.9	57.3	62.0	58.5	56.2	55.8	24.5	59.5
TE-MNLI	$\text{RoBERTa}_{\text{large}}$	✓	28.6	77.6	60.4	40.2	89.6	90.2	92.8	82.8	92.0	**48.8**	70.3
TE-MNLI	$\text{BART}_{\text{large}}$	✓	48.2	74.8	57.1	35.4	89.0	91.1	93.1	81.4	91.9	47.7	71.0
Prompting*	$\text{RoBERTa}_{\text{base}}$	-	34.1	54.6	51.1	-	81.9	81.8	83.1	78.3	83.5	-	-
Mining-based*	$\text{RoBERTa}_{\text{base}}$	✗	56.1	79.2	80.4	-	85.6	86.7	92.0	80.5	92.0	-	-
UniMC*	$\text{ALBERT}_{\text{xxlarge}}$	✓	-	81.3	88.9	-	**91.6**	**94.8**	-	-	-	-	-
UniMC (Rerun)	$\text{ALBERT}_{\text{xxlarge}}$	✓	59.0	84.3	89.2	43.7	90.1	93.6	94.3	87.3	93	45.6	78.0
SSTuning-base	$\text{RoBERTa}_{\text{base}}$	✗	59.1	79.9	82.7	47.2	86.4	88.2	92.9	83.8	94.0	45.0	75.9
SSTuning-large	$\text{RoBERTa}_{\text{large}}$	✗	62.4	83.7	85.6	56.7	90.1	93.0	95.2	87.4	95.2	46.9	79.6
SSTuning-ALBERT	$\text{ALBERT}_{\text{xxlarge}}$	✗	**63.5**	**85.5**	**92.4**	**62.0**	90.8	93.4	**95.8**	**89.5**	**95.6**	45.2	**81.4**

Table 1: Main results for 4 topic classification tasks and 6 sentiment analysis tasks. ❖: the original training sets (see dataset sizes in Table 6) are used to provide results under supervised settings, served as upper bound, otherwise zero-shot results are reported. *: results are taken from corresponding papers. "Labeled" indicates whether the model uses labeled (✓) or unlabeled (✗) data. "Avg" is the arithmetic mean accuracy of all the datasets. For SSTuning models, we report the mean accuracy of 5 runs using different seeds. The best results for each dataset are in bold.

	TC	SA	All
Amazon	63.4	81.4	74.2
Wikipedia	63.4	77.9	72.1
Amazon + Wikipedia	**67.2**	**81.7**	**75.9**

Table 2: Zero-shot results with different tuning datasets. The best result is in **Bold**.

	TC	SA	All
First sentence prediction	**67.2**	81.7	**75.9**
Last sentence prediction	59.8	**82.2**	73.3
Next sentence selection	54.8	81.9	71.1
Random sentence prediction	56.8	80.8	71.2

Table 3: Zero-shot results with different tuning objectives. The best results are in **Bold**.

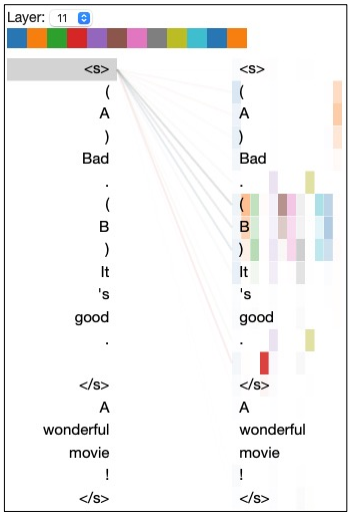


Figure 3: Attention map of `[CLS]` token (which is <s> here for RoBERTa backbone) in the last layer for a movie review. This figure is generated with BertViz (Vig, 2019).

whether the movie review "A wonderful movie!" is negative or positive. We set the verbalizers as "Bad." and "It's good." to see how the length of options impacts the decision. The prediction of the model is 1, which is correct. We find that `[CLS]` token attends more to the second opinion, especially to the tokens around the index indicator "B" in the last layer. This is consistent with our intuitions. For humans, when we do classification tasks, we normally compare the options and select the option that best matches the text. We show additional attention maps and analysis in Appendix B.2.3.

4.3.3 Importance of Index Indicators

To further understand how the index indicator guides the model to make the prediction, we employ different indicator designs during the tuning and inference stage. Specifically, we consider different formats of the index indicator, which are: 1) alphabet characters (A, B, C...), which is the default format; 2) numerical index (0, 1, 2...); 3) same index indicator for all options (0, 0, 0...). During the inference, we also consider two special indicators: 4) same alphabet characters (A, A, A...), and 5) rearranged alphabet characters (B, A, D, C...). The results are shown in Table 5. There is not much difference between using alphabet characters and numerical indexes, as shown in cases 1 and 2. As shown in case 3, using the same characters will de-

Verbalizer for "negative"	Verbalizer for "positive"	UniMC(w/o Qn)		SSTuning-ALBERT	
		SST-2	IMDb	SST-2	IMDb
Bad.	Good.	87.0	91.9	90.7	93.9
Terrible.	Great.	88.5	91.7	91.4	94.3
Negative.	Positive.	86.0	90.3	92.2	92.6
Negative!	Positive!	88.9	90.2	92.1	92.4
Terrible!	Awesome!	88.4	91.1	90.9	94.0
Bad, terrible and negative.	Good, great, and positive.	80.7	87.5	87.3	90.8
I don't like the movie!	I like the movie!	91.5	92.9	89.8	90.3
Terrible!	I like the movie! It is wonderful!	66.4	75.1	86.8	92.1
It's terrible.	It's great.	91.6	93.0	90.6	94.1
It's negative.	It's positive.	85.6	89.9	89.2	91.3
	Average	85.5	89.4	**90.1**	**92.6**
	Standard Deviation	7.4	5.3	**1.9**	**1.5**

Table 4: Comparison of zero-shot results for 2 sentiment analysis tasks with different verbalizers. The best average results are in **bold**.

	Tuning	Inference	Avg	Std
1	(A, B, C...)	(A, B, C...)	75.9	0.3
2	(0, 1, 2...)	(0, 1, 2...)	75.6	0.4
3	(0, 0, 0...)	(0, 0, 0...)	74.1	0.6
4	(A, B, C...)	(A, A, A...)	32.0	1.1
5	(A, B, C...)	(B, A, D, C...)	23.4	12.1

Table 5: Performance with same and different index indicators during tuning and inference. "Std" indicates Standard Deviation.

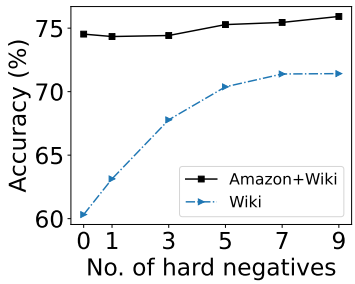


Figure 4: Zero-shot accuracy with different numbers of hard negatives.

grade the performance but not much, which means the model can rely on position embedding of the index indicator to make the correct predictions. As shown in cases 4 and 5, using inconsistent index indicators will greatly degrade the performance, which further verifies the importance of using consistent index indicators to make correct predictions.

4.3.4 Impact of Hard Negative Samples

Intuitively, adding more hard negatives will make the task more difficult, thus forcing the mode to better understand the semantics of the sentences. We tested the impact of hard negatives based on two settings: 1) train with both the Amazon reviews and Wikipedia, each with 2.56M samples; 2) train with only 2.56M Wikipedia samples. We don't train with only Amazon reviews since they don't have hard negatives. The results with $0, 1, 3, 5, 7, 9$ hard negatives are shown in Figure 4.

In general, adding more hard negatives will improve the performance. For the case with both datasets, the impact of hard negatives is small. This is because the Amazon review dataset alone can achieve good performance, as shown in Table 2. However, hard negatives have a significant impact on the setting with only Wikipedia for tuning. The possible reason is that without hard negatives the model may only learn keyword matching instead of semantics since the keywords may appear many times in the same Wikipedia article.

4.3.5 Additional Analysis

We report additional analysis in Appendix B.2. As shown in Figure 5, we can further improve the performance by increasing the tuning sample size. We also compared SSTuning-base with different numbers of output labels N_{model}. As shown in Appendix B.2.2, we can increase N_{model} to inference on datasets with more classes.

5 Related Work

Zero-shot text classification. Zero-shot learning has the advantage that no annotated data is required for downstream tasks. Prompting-based methods (Brown et al., 2020; Chowdhery et al., 2022; Schick and Schütze, 2021; Gao et al., 2021a) that reformulate the inputs as prompts can perform much

worse in the zero-shot setting than few-shot settings as it may be hard for the PLMs to interpret the templates. A better option may be mining-based method (van de Kar et al., 2022), which mines the labeled data from the unlabeled corpus for fine-tuning each downstream task. Similarly, generation-based approaches (Meng et al., 2022; Ye et al., 2022) generate labeled data with a generative PLM.

More works on zero-shot text classifications are based on transfer learning. Instruction-tuning-based models like FLAN (Wei et al., 2022) and T0 (Sanh et al., 2022), fine-tine PLMs on a collection of datasets described by instructions or prompts to improve performances on unseen tasks. PLMs can also be meta-tuned (Zhong et al., 2021) on text classification datasets and do zero-shot on other classification datasets. UniMC (Yang et al., 2022) converts several tasks to multiple-choice tasks and does zero-shot inference on tasks that can be formulated in the same format. Another line of work is to convert text classification problems to textual entailment problems. By fine-tuning on natural language inference datasets (Yin et al., 2019) or a dataset from Wikipedia (Ding et al., 2022), the models can do inference directly on text classification datasets. Instead of using annotated datasets, we only need unlabeled data to generate a large number of labeled samples as tuning and validation sets by exploring the inherent text structure.

Self-supervised learning. Self-supervised learning has been widely applied during language model pre-training by leveraging the input data itself as supervision signals (Liu et al., 2021). Left-to-right language modeling (Radford and Narasimhan, 2018) and masked language modeling (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020) help learn good sentence representations. In order to capture the sentence-level relations of downstream tasks, Devlin et al. (2019) pre-train a next sentence prediction task and Lan et al. (2020) use sentence order prediction task to model the inter-sentence coherence. Wang et al. (2020) combine the two objectives to form a three-way classification task. Instead of modeling the inter-sentence relations, Meng et al. (2021) employs sequence contrastive learning to align the corrupted text sequences that originate from the same input source and guarantee the uniformity of the representation space. Our work uses a harder learning objective called first sentence prediction: given several options and text, find the corresponding first sentence preceding the text.

6 Conclusions

In this work, we propose a new learning paradigm called SSTuning for zero-shot text classification tasks. By forcing the model to predict the first sentence of a paragraph given the rest, the model learns to associate the text with its label for text classification tasks. Experimental results show that our proposed method outperforms state-of-the-art baselines on 7 out of 10 tasks and the performance is more stable with different verbalizer designs. Our work proves that applying self-supervised learning is a promising direction for zero-shot learning. In the future, we plan to apply SSTuing to other tasks by designing proper learning objectives.

Limitations

In this work, we proposed SSTuning for zero-shot text classification tasks. During inference, we may need to design verbalizers even though we can use templates like "This text is about [label name]". For simplicity and fair comparison, we only refer to previous works for such designs, which may be sub-optimal. As shown in Table 4, using the verbalizers "Terrible." and "Great." work better than "It's terrible." and "It's great." for the SST-2 and IMDA tasks that we reported in the main results. If the labeled validation set is provided, the model may perform better by choosing verbalizers based on the validation set.

Due to limited computation resources, we only tuned the model with 5.12 million samples, which is only a small portion of the available samples. We believe that tuning the model on a larger dataset help improve the performance. Even though the computational cost will also increase, it is worth it since no more training is needed at the inference phase. In addition, we did not do extensive hyperparameter searches except for the learning rate, which may further improve the performance.

In our experiment, we only tested the method with discriminative models like RoBERTa and ALBERT. Its performance with generative models is not known. It is non-trivial to test on such models since generative models can do both natural language understanding tasks and natural language generation tasks. We leave this as future work.

Acknowledgements

This research is supported, in part, by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), Nanyang Technological University, Singapore. Chaoqun Liu and Guizhen Chen extend their gratitude to Interdisciplinary Graduate Programme and School of Computer Science and Engineering, Nanyang Technological University, Singapore, for their support. This research is also supported by the Ministry of Education Tier 1 grant (MOE Tier 1 RS21/20).

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Hantian Ding, Jinrui Yang, Yuqian Deng, Hongming Zhang, and Dan Roth. 2022. Towards open-domain topic classification. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*.

Jiangshu Du, Wenpeng Yin, Congying Xia, and Philip S. Yu. 2023. Learning to select from multiple options. In *Proceedings of the 2023 AAAI*.

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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

Ariel Gera, Alon Halfon, Eyal Shnarch, Yotam Perlitz, Liat Ein-Dor, and Noam Slonim. 2022. Zero-shot text classification with self-training. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1896–1907. Association for Computational Linguistics.

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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, pages 331–339. Morgan Kaufmann.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Sasko, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander M. Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2021, Online and Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 175–184. Association for Computational Linguistics.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*.

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. *CoRR*, abs/1907.11692.

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 *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150. Association for Computational Linguistics.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. In *Advances in Neural Information Processing Systems*.

Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. 2021. COCO-LM: correcting and contrasting text sequences for language model pretraining. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 23102–23114.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 115–124. The Association for Computer Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 255–269. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. 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, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Mozes van de Kar, Mengzhou Xia, Danqi Chen, and Mikel Artetxe. 2022. Don't prompt, search! mining-based zero-shot learning with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42,

Florence, Italy. Association for Computational Linguistics.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. 2020. Structbert: Incorporating language structures into pre-training for deep language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022*.

Ping Yang, Junjie Wang, Ruyi Gan, Xinyu Zhu, Lin Zhang, Ziwei Wu, Xinyu Gao, Jiaxing Zhang, and Tetsuya Sakai. 2022. Zero-shot learners for natural language understanding via a unified multiple choice perspective. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. *CoRR*, abs/2202.07922.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 3912–3921.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 2856–2878. Association for Computational Linguistics.

A Additional Dataset Details

A.1 Tuning Datasets

The original unlabeled datasets can be noisy and some paragraphs are not suitable for generating tuning datasets. We filter the paragraphs with the following features: 1) the paragraph only contains 1 sentence; 2) the first sentence contains less than or equal to 3 characters; 3) the first sentence only contains non-alphabetic symbols; 4) repeated paragraphs. Some of the final generated samples from English Wikipedia and Amazon product reviews are shown in Table 9.

Dataset	# Class	# Train	# Val	# Test
Yahoo.	10	1.4M	0	60k
AG News	4	120k	0	7.6k
DBPedia	14	560k	0	70k
20 News.	20	11,314	0	7532
SST-2	2	67,349	872	0
IMDB	2	25k	0	25k
Yelp	2	560k	0	38k
MR	2	8,530	1,066	1,066
Amazon	2	3.6M	0	400k
SST-5	5	8,544	1,101	2,210

Table 6: Dataset statistics for evaluation datasets

A.2 Evaluation Datasets

We summarize the dataset statistics for the evaluation datasets in Table 6. We download all the datasets from Huggingface (Lhoest et al., 2021), except 20newsgroup. For Yahoo Topics, we concatenate the question and answer as inputs. For DBPedia and Amazon, we concatenate the title and content. For 20newsgroup, we follow the recommendations to remove headers, footers, and quotas[4]. However, if the text becomes empty after removing the components, we will use the original text instead.

The verbalizers for each dataset are shown in Table 7. We try to unify the verbalizer design for similar tasks. For topic classification tasks, we use the template *"This text is about [ ]."* after converting the class names to meaningful words. For binary classifications, we use *"It's terrible."* for negative class and *"It's great."* for positive class. For SST-5, we refer to (Gao et al., 2021a) to design the verbalizers. Some of the reformulated text for the evaluation datasets are shown in Table 10.

B Additional Experiment Details

B.1 Experiment setup

The hyperparameters for the main results (Section 4.1) are shown in Table 8. We try to use the same settings as much as possible. The training time for the three SSTuning models is with 5.12M tuning

[4] https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html

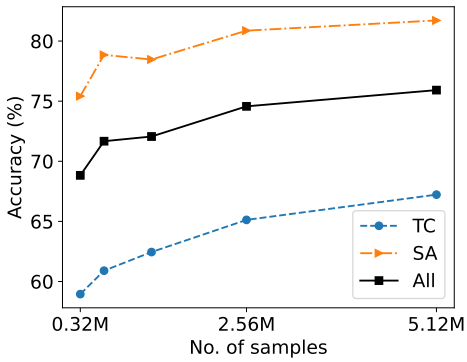


Figure 5: Zero-shot accuracy with different training sample sizes. Mean accuracy over 4 topic classification tasks, 6 sentiment analysis tasks, and all the tasks are reported.

samples and 64k validation samples (also generated via FSP).

B.2 Additional Results

B.2.1 Impact of Tuning Sample Size

To test how the tuning sample size impacts the performance, we trained SSTuning-base with 320k, 640k, 1.28M, 2.56M, and 5.12M samples, with half generated from Wikipedia and half from Amazon reviews. The results are shown in Figure 5. With more samples, the performances are increasing in general, especially for topic classification tasks. With such observation, it is likely to further improve the performance by increasing the tuning sample size. Even though tuning on larger datasets is more computationally expensive, it is worth doing since no further training is required for downstream tasks.

B.2.2 Impact of the Number of Output Labels

In our main results, we set the number of output labels N_{model} as 20. However, a classification dataset may have more than 20 classes. To test the scalability of the label number, we tune another variant for SSTuning-base. We use numerical numbers (0, 1, 2...) as the index indicator and set N_{model} as 40. The comparison between the two versions is shown in Table 11. Increasing N_{model} from 20 to 40 only degrade the performance by 1.4 points (75.9% to 74.5%), showing the good scalability of our approach. As an alternative for the datasets with more classes, we can split the labels and do a multi-stage inference.

B.2.3 Classification Mechanism

We plot more attention maps for the example discussed in Section 4.3.2 in Figure 6. We focus on a few important tokens, including the classification token <s>, the option indicators A and B, and the separator token </s>. In Layer 0, <s> attends to all the options and the text. A and B attend more to its own options. </s> attend more to the text tokens. In higher layers, A and B attend even more to their own option tokens (Layer 1) but also have some interactions (Layer 4). In layer 9, A and B attend more its own option tokens again and also the period mark, while </s> attend to both the text tokens and the options tokens for B (the positive option). In the end, <s> attends to B, which is the positive option. Based on the observations, we hypothesize that the model has the capability to encode the options and text separately, compare the options and text, and choose the positive option in the end.

Dataset	Verbalizers
Yahoo Topics	"This text is about society & culture.", "This text is about science & mathematics.", "This text is about health.", "This text is about education & reference.", "This text is about computers & internet.", "This text is about sports.", "This text is about business & finance.", "This text is about entertainment & music.", "This text is about family & relationships.", "This text is about politics & government."
AG News	"This text is about politics.", "This text is about sports.", "This text is about business.", "This text is about technology."
DBPedia	"This text is about company.", "This text is about educational institution.", "This text is about artist.", "This text is about athlete.", "This text is about office holder.", "This text is about mean of transportation.", "This text is about building.", "This text is about natural place.", "This text is about village.", "This text is about animal.", "This text is about plant.", "This text is about album.", "This text is about film.", "This text is about written work."
20 Newsgroup	"This text is about atheism.", "This text is about computer graphics.", "This text is about microsoft windows.", "This text is about pc hardware.", "This text is about mac hardware.", "This text is about windows x.", "This text is about for sale.", "This text is about cars.", "This text is about motorcycles.", "This text is about baseball.", "This text is about hockey.", "This text is about cryptography.", "This text is about electronics.", "This text is about medicine.", "This text is about space.", "This text is about christianity.", "This text is about guns.", "This text is about middle east.", "This text is about politics.", "This text is about religion."
SST-2, IMDB, Yelp, MR, Amazon	"It's terrible.", "It's great."
SST-5	"It's terrible.", "It's bad.", "It's okay.", "It's good.", "It's great."

Table 7: Verbalizers for the evaluation datasets.

Parameter	Fine-tuning	SSTuning-base/SSTuning-large	SSTuning-ALBERT
Model	$\text{RoBERTa}_{\text{large}}$ (355M)	$\text{RoBERTa}_{\text{base}}$/$\text{RoBERTa}_{\text{large}}$ (355M)	$\text{ALBERT}_{\text{xxlarge}}$(V2)(235M)
Model Selection	Best	Best	Best
Batch Size	16	128	64
Precision	FP16	FP16	FP16
Optimiser	AdamW	AdamW	AdamW
Learning Rate	1e-5	2e-5	1e-5
LR Scheduler	linear decay	linear decay	linear decay
AdamW Epsilon	1e-8	1e-8	1e-8
AdamW β_1	0.9	0.9	0.9
AdamW β_1	0.999	0.999	0.999
Weight Decay	0.01	0.01	0.01
Classifier Dropout	0.1	0.1	0.1
Attention Dropout	0.1	0.1	0
Hidden Dropout	0.1	0.1	0
Max Steps	-	40000	80000
Max Epochs	3	1	1
Hardware	1 NVIDIA V100	8 NVIDIA V100	4 NVIDIA A100
Training time	-	3h/8h	31h

Table 8: Hyperparameters and training information for full-shot fine-tuning, SSTuing-base, SSTuning-large and SSTuing-ALBERT.

Dataset	Label	Positive Option	Generated Text
Wikipedia	12 (M)	In parliament, Satouri serves on the Committee on Employment and Social Affairs and the Subcommittee on Security and Defence.	(A) [PAD] (B) The work of lojas, are found in both the town and the countryside. (C) [PAD] (D) [PAD] (E) [PAD] (F) [PAD] (G) In 1848 riots and looting took place, and in 1849 an epidemic broke out. (H) [PAD] (I) Leptostylus retrorsus is a species of beetle in the family Cerambycidae. (J) The 2020 – 21 Russian Football National League was the 29th season of Russia's second - tier football league since the dissolution of the Soviet Union. (K) [PAD] (L) He opposed several times to the decisions of his party, as when Congress was dissolved in 2019, he supported Martín Vizcarra's measure and did not attend to the inauguration of Vice President Mercedes Araoz. (M) In parliament, Satouri serves on the Committee on Employment and Social Affairs and the Subcommittee on Security and Defence. (N) [PAD] (O) [PAD] (P) [PAD] (Q) [PAD] (R) [PAD] (S) The church has a rectangular nave with stone walls that are around 2 meters thick. (T) On February 2,, the Blue Jays and Downs agreed to a one - year, $ 1. 025 million contract, avoiding the arbitration process. [SEP] In addition to his committee assignments, he is part of the parliament's delegations to the Parliamentary Assembly of the Union for the Mediterranean and for relations with the NATO Parliamentary Assembly.
Wikipedia	0 (A)	Rawat emigrated to Canada from India in 1968.	(A) Rawat emigrated to Canada from India in 1968. (B) Meskowski was a racing car constructor. (C) [PAD] (D) , there were 42 people who were single and never married in the municipality. (E) [PAD] (F) [PAD] (G) [PAD] (H) [PAD] (I) [PAD] (J) [PAD] (K) It is a Church of England school within the Diocese of Salisbury. (L) Falkoner Allé was opened to the public after Hømarken (literally " Hayfield "), an area to the north belonging to Ladegården, originally a farm under Copenhagen Castle, was auctioned off. (M) [PAD] (N) [PAD] (O) In the fall of her senior year at McDonogh, Cummings committed to play for the University of Marylandś womenś lacrosse team as the nationś top recruit. (P) Ranville is a native of Flint, Michigan and attended St. Agnes High School. (Q) The Dodgeś Institute of Telegraphy was housed in the Institutes building at 89 East Monroe. (R) During 2004 - 2011, Rawat was President of the Communications Research Centre, Canadaś centre of excellence for telecommunications R & D, with 400 staff and an annual budget of over $ 50 million. (S) [PAD] (T) [PAD] [SEP] She speaks English, French, Hindi and Spanish.
Amazon Product Review	1 (B)	This popcorn is really best suited for kettle corn.	(A) [PAD] (B) This popcorn is really best suited for kettle corn. (C) Professional Quality with Amazing results. (D) [PAD] (E) [PAD] (F) [PAD] (G) I found my new S6 to be a little TOO thin, and so slick it was sliding off of everything, so I wanted a clear bumper. (H) Excellent price. (I) [PAD] (J) [PAD] (K) Iv́e always loved Bounce dryer sheets, but was not too fond of the synthetic " Outdoor Fresh " scents. (L) [PAD] (M) [PAD] (N) [PAD] (O) I cut the cord and bought this mohu leaf antenna to get the local channels. (P) [PAD] (Q) The product came pretty quickly with very easy instructions. (R) [PAD] (S) [PAD] (T) Watch Land Before Time and had to have one for Xmas. [SEP] The kernels pop up to a nice large size. Donf́ think I would compare them to mushrooms - button mushrooms maybe (LOL). They are a bit on the chewy side if you go the butter route. They are really best as crisp, salty-sweet kettle corn. Yum! We use a Whirley Pop for popcorn–our favorite kitchen "appliance"! Donf́ know if some other method would make the popcorn crisper. No matter–would buy this again just for the way it tastes as kettle corn!
Amazon Product Review	18 (S)	Works pretty good.	(A) [PAD] (B) [PAD] (C) [PAD] (D) [PAD] (E) [PAD] (F) [PAD] (G) [PAD] (H) [PAD] (I) [PAD] (J) [PAD] (K) [PAD] (L) [PAD] (M) [PAD] (N) [PAD] (O) [PAD] (P) Great value for a creeper. (Q) [PAD] (R) [PAD] (S) Works pretty good. (T) [PAD] [SEP] Just wish the fm stations on the device would go lower. The best one in my area is 85.1 but the device only goes to 88.1. Still a great product.

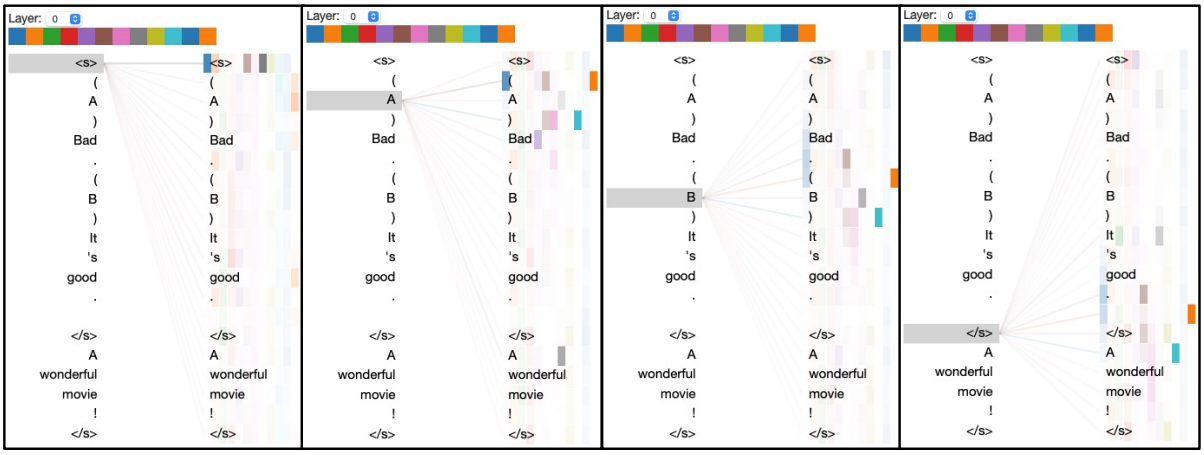
Table 9: Examples generated for SSTuning with English Wikipedia and Amazon product review dataset.

Dataset	Label	Positive Option	Reformulated Text
AG News	3 (D)	This text is about technology.	(A) This text is about politics. (B) This text is about sports. (C) This text is about business. (D) This text is about technology. (E) [PAD] (F) [PAD] (G) [PAD] (H) [PAD] (I) [PAD] (J) [PAD] (K) [PAD] (L) [PAD] (M) [PAD] (N) [PAD] (O) [PAD] (P) [PAD] (Q) [PAD] (R) [PAD] (S) [PAD] (T) [PAD] [SEP] REVIEW: 'Half-Life 2' a Tech Masterpiece (AP) AP - It's been six years since Valve Corp. perfected the first-person shooter with "Half-Life." Video games have come a long way since, with better graphics and more options than ever. Still, relatively few games have mustered this one's memorable characters and original science fiction story.
DBPedia	9 (J)	This text is about animal.	(A) This text is about company. (B) This text is about educational institution. (C) This text is about artist. (D) This text is about athlete. (E) This text is about office holder. (F) This text is about mean of transportation. (G) This text is about building. (H) This text is about natural place. (I) This text is about village. (J) This text is about animal. (K) This text is about plant. (L) This text is about album. (M) This text is about film. (N) This text is about written work. (O) [PAD] (P) [PAD] (Q) [PAD] (R) [PAD] (S) [PAD] (T) [PAD] [SEP] Periscepsia handlirschi. Periscepsia handlirschi is a species of fly in the family Tachinidae.
SST-2	1 (B)	It's great.	(A) It's terrible. (B) It's great. (C) [PAD] (D) [PAD] (E) [PAD] (F) [PAD] (G) [PAD] (H) [PAD] (I) [PAD] (J) [PAD] (K) [PAD] (L) [PAD] (M) [PAD] (N) [PAD] (O) [PAD] (P) [PAD] (Q) [PAD] (R) [PAD] (S) [PAD] (T) [PAD] [SEP] charles ' entertaining film chronicles seinfeld 's return to stand-up comedy after the wrap of his legendary sitcom , alongside wannabe comic adams ' attempts to get his shot at the big time .
SST-5	3 (D)	It's good.	(A) It's terrible. (B) It's bad. (C) It's okay. (D) It's good. (E) It's great. (F) [PAD] (G) [PAD] (H) [PAD] (I) [PAD] (J) [PAD] (K) [PAD] (L) [PAD] (M) [PAD] (N) [PAD] (O) [PAD] (P) [PAD] (Q) [PAD] (R) [PAD] (S) [PAD] (T) [PAD] [SEP] u.s. audiences may find -lrb- attal and gainsbourg 's -rrb- unfamiliar personas give the film an intimate and quaint reality that is a little closer to human nature than what hollywood typically concocts .

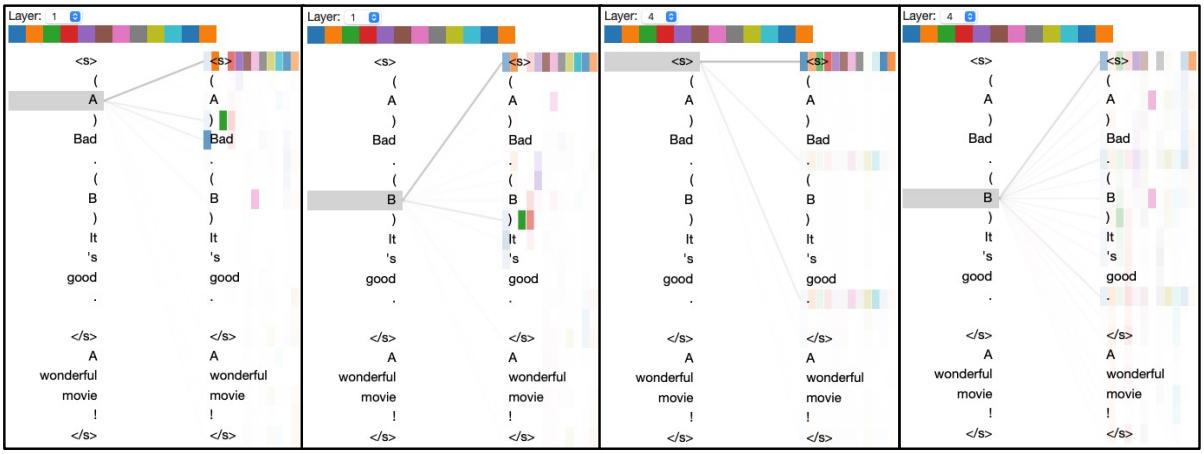
Table 10: Examples after reformulation for 4 evaluation datasets.

	N_{model}	Topic Classification				Sentiment Analysis						Avg
		yah	agn	dbp	20n	sst2	imd	ylp	mr	amz	sst5	
SSTuning-base	20	59.1	79.9	82.7	47.2	86.4	88.2	92.9	83.8	94.0	45.0	75.9
SSTuning-base	40	58.0	79.3	79.8	49.1	84.4	88.2	91.7	82.2	93.3	39.4	74.5

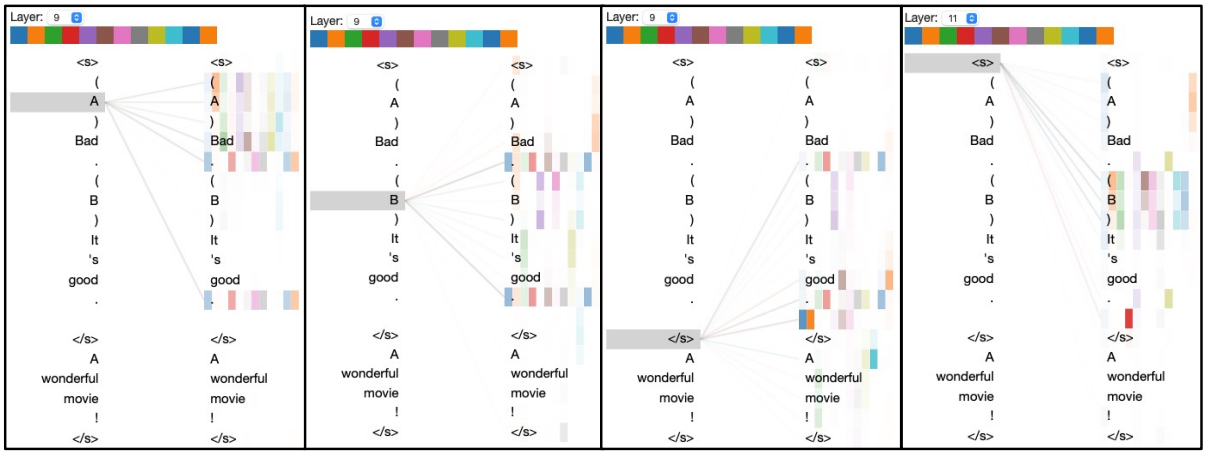
Table 11: Accuracy over different number of labels N_{model}.



Layer 0: token “<s>” Layer 0: token “A” Layer 0: token “B” Layer 0: token “</s>”



Layer 1: token “A” Layer 1: token “B” Layer 4: token “<s>” Layer 4: token “B”



Layer 9: token “A” Layer 9: token “B” Layer 9: token “</s>” Layer 11: token “<s>”

Figure 6: Attention map for a movie review example. The original text is "A wonderful movie!" and the verbalizers are "Bad." and "It's Good.". The model is SSTuning-base with 2 classes.

ACL 2023 Responsible NLP Checklist

A For every submission:

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

☒ A2. Did you discuss any potential risks of your work?
We are working on text classification, which classifies text into a certain category. This should not have any potential risk.

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

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

B ☑ Did you use or create scientific artifacts?

Sec 3 Experiment Setup

☑ B1. Did you cite the creators of artifacts you used?
Sec 3 Experiment Setup

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We only use public datasets, which do not need a license.

☒ 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)?
We only use publicly available datasets, which should not have an issue.

☒ 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?
We only use publicly available datasets, which are commonly used.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Sec 3 Experiment Setup

☑ 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.
Appendix A.2 Evaluation Datasets

C ☑ Did you run computational experiments?

Section 3 and 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix B.1 Experiment setup

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Sec 4 Results and Analysis

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

☑ 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.)?
Sec 3.4 Implementation Details

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

Left blank.

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

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

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

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

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