WC-SBERT: Zero-Shot Topic Classification Using SBERT and Light Self-Training on Wikipedia Categories

TE-YU CHI, Dept. of CSIE, National Taiwan University, Taiwan
JYH-SHING ROGER JANG, Dept. of CSIE, National Taiwan University, Taiwan

In NLP (natural language processing), zero-shot topic classification requires machines to understand the contextual meanings of texts in a downstream task without using the corresponding labeled texts for training, which is highly desirable for various applications [2]. In this paper, we propose a novel approach to construct a zero-shot task-specific model called WC-SBERT with satisfactory performance. The proposed approach is highly efficient since it uses light self-training requiring target labels (target class names of downstream tasks) only, which is distinct from other research that uses both the target labels and the unlabeled texts for training. In particular, during the pre-training stage, WC-SBERT uses contrastive learning with the multiple negative ranking loss [9] to construct the pre-trained model based on the similarity between Wiki categories. For the self-training stage, online contrastive loss is utilized to reduce the distance between a target label and Wiki categories of similar Wiki pages to the label. Experimental results indicate that compared to existing self-training models, WC-SBERT achieves rapid inference on approximately 6.45 million Wiki text entries by utilizing pre-stored Wikipedia text embeddings, significantly reducing inference time per sample by a factor of 2,746 to 16,746. During the fine-tuning step, the time required for each sample is reduced by a factor of 23 to 67. Overall, the total training time shows a maximum reduction of 27.5 times across different datasets. Most importantly, our model has achieved SOTA (state-of-the-art) accuracy on two of the three commonly used datasets for evaluating zero-shot classification, namely the AG News (0.84) and Yahoo! Answers (0.64) datasets. The code for WC-SBERT is publicly available on GitHub[1], and the dataset can also be accessed on Hugging Face[2].

CCS Concepts: • **Computing methodologies** → **Lexical semantics**; **Information extraction**; *Natural language generation*; *Language resources*.

Additional Key Words and Phrases: Zero-shot topic classification, SBERT, Wikipedia, Self-training, Contrastive learning, Knowledge graph, LLM

1 INTRODUCTION

Zero-shot topic classification is a text classification task distinguished by its ability to classify text into predefined classes without using prior labeled data. This task holds significant research value in NLP as it copes not only with known classes but also with newly appearing classes and texts from unknown domains, which can usually show a better generalization capability.

Despite the recent success achieved by Gera et al. [8] in using a self-training approach, wherein prediction results were used as labels on an unlabeled dataset, and these "pseudo-labels" were subsequently used to train

[1]https://github.com/seventychi/wc-sbert
[2]https://huggingface.co/datasets/seven-tychi/wikipedia-categories

Authors' addresses: Te-Yu Chi, d09922009@ntu.edu.tw, Dept. of CSIE, National Taiwan University, Taiwan; Jyh-Shing Roger Jang, Dept. of CSIE, National Taiwan University, Taiwan, jang@mirlab.org.

ACM 2157-6912/2024/7-ART
https://doi.org/10.1145/3678183

the model through multiple iterations. This method still leaves room for further improvement since it requires the unlabelled texts of the target dataset of the downstream task and its training process is lengthy.

The proposed WC-SBERT leverages SBERT [18] and uses the publicly available Wikipedia dataset from Hugging Face as a common ground for zero-shot classification. The textual content of the Wiki pages and their corresponding categories are referred to as "Wiki texts" and "Wiki categories", respectively, in the rest of this paper. WC-SBERT is constructed by a two-stage approach. In the first stage of pre-training, Wiki categories corresponding to the same Wiki page are positively correlated and thus are used to train a general-purpose SBERT-base model. In the second stage of self-training, a target label (label/class of the downstream task) and a Wiki category are considered positively correlated if the target label are similar to the Wiki text bearing the category, and thus can be used to perform task-specific fine-tuning of the general model obtained in the first stage. The proposed self-training is light in computing since it only use the target labels, which diverges from traditional self-training methods that require both the target labels and the unlabeled texts. The primary contributions and novelties of the proposed approach are as follows.

- **Novel pre-training**: By using Wiki categories of the same Wiki pages as training samples, We can efficiently construct a general-purpose SBERT model for subsequent task-specific fine-tuning.
- **Efficient self-training**: The propose light self-training for task-specific fine-tuning utilizes the target label and their similar Wiki categories (obtained via similar Wiki texts) for training. This method improves upon traditional self-training, which requires the downstream dataset's labels and its unlabeled texts. Our approach requires only the dataset's labels of the downstream task (thus "light self-training"), making it more aligned to the true spirit of zero-shot classification.

We have conducted extensive experiments on three topic classification benchmarks, including AG News, Yahoo! Answers, and DBpedia, to demonstrate the effectiveness and generalizability of the proposed approach, which can achieve state-of-the-art performance of zero-shot classification on the first two datasets.

2 RELATED WORK

Zero-shot topic classification is a text classification task that has the characteristic of classifying text into predefined classes without prior labeled data. We focus on open-domain zero-shot topic classification, which has better generalization capability compared to specific domains. It can not only classify text into known classes but also handle new classes and texts from unknown domains. Chang et al. [3] conducted the initial research on this task, referred to as "dataless classification" at that time. It is a method that relies solely on general knowledge for classification and does not require any domain-specific data. Since then, this task has gained significant attention.

With the advancement of deep neural networks, topic classification has experienced a significant shift towards the use of pre-trained language models (PLMs) (Yang et al., 2019; Zaheer et al., 2020; Chen et al., 2022). The first-generation PLMs, such as Word2Vec [14] and GloVe [16], relied on word embeddings and typically classified the text based on word similarity. While these models were effective in capturing the semantic meaning of words and sentences, they lacked the ability to understand complex linguistic concepts and contextual information. In contrast, the second-generation PLMs, such as BERT [6], RoBERTa [12], T5 [20], and GPT-2 [17], are based on contextual embeddings. These models become mainstream techniques in text classification since they can capture the semantic meaning of words in different contexts and can be fine-tuned for this NLP task.

Recently, Ding et al. [7] and Chu et al. [5] utilized Wikipedia data as training data to construct BERT-based classifiers. This choice is due to the vast amount of article data available in Wikipedia, which covers a wide range of general knowledge. Therefore, it is considered highly suitable for open-domain zero-shot topic classification tasks. The datasets constructed by Chu et al. and Ding et al. contain 5.75 million documents with 1.19 million categories, and 3.3 million documents with 674 top-level categories, respectively. Compared to their works, we select Wikipedia dataset from Hugging Face as data source. The dataset stands out for its substantial size,

comprehensive coverage, and meticulous categorization. With over 6.4 million documents and more than 1.5 million categories, it has shown better performance on popular topic classification datasets in our experiments.

Self-training is one of the commonly used techniques in the fields of semi-supervised [24] and unsupervised learning [29]. It is characterized by utilizing model predictions to expand the training dataset and iteratively improve the model through self-training. In recent studies (Liu et al., 2023; Yang et al., 2022; van de Kar et al., 2022; Gera et al., 2022) about self-training in zero-shot text classification, Gera et al. [8] overall achieved the best performance. Self-training involves pseudo-labeling the training and test sets of the target dataset. Subsequently, the model undergoes multiple iterations of training using these pseudo-labels to enhance its understanding of the dataset. While this approach has indeed demonstrated state-of-the-art results on multiple datasets, it requires prior examination of the data in each individual target dataset. In certain commercial contexts, this may not accurately represent the absence of zero-shot data, necessitating additional computational resources to train the model on different datasets. Our study aims to improve the self-training approach in terms of dataset selection and performance.

SBERT (*Sentence-BERT*) is a focal point, primarily used for sentence-level similarity comparisons. This model, as detailed by Reimers and Gurevych [18], leverages siamese and triplet network structures to generate semantically meaningful sentence embeddings. These embeddings are typically compared using methods such as cosine similarity. Such an architecture marks a significant improvement in efficiency for similarity comparisons when contrasted with original BERT-based models.

A pivotal aspect of SBERT's application in various downstream tasks is the self-training process, wherein the selection of an appropriate loss function is critical. Our study employs both the Multiple Ranking Loss (MNR Loss) and the Online Contrastive Loss, each catering to different steps of pre-training and self-training. The MNR Loss is particularly adept when the training dataset comprises solely positive pairs and is primarily applied during the WC-SBERT pre-training stage, where negative samples are absent. Conversely, the Online Contrastive Loss is utilized for downstream tasks involving negative samples labeled differently.

Both of these loss functions operate on the principles of contrastive learning, effectively drawing positive data closer in the vector space while distancing negative data thereby facilitating the formation of clusters of similar sentences. By leveraging the architectural and training/inference optimizations of SBERT over traditional BERT models, our study utilizes the *all-mpnet-base-v2* pre-trained model from SBERT as the primary foundational model. This strategic choice underscores the advancements in sentence-level semantic analysis brought about by SBERT.

3 PROPOSED APPROACH

Previous research has extensively explored various methods and models for zero-shot topic classification, and most of the proposed approaches divide the modeling process into two stages of pre-training to obtain a general text classifier, and self-training to refine the general model into a task-specific model. Since most of the self-training stage of the proposed approaches in the literature requires the use of the unlabeled texts, which is not desirable due to the following facts:

- The texts of the target dataset (dataset for downstream task) may not be available in a true zero-shot setting.
- The BERT-based model has input length limitation which may make the training on long texts infeasible.
- The self-training of downstream classifiers is usually time-consuming when the bulky full texts are used.

To overcome these challenges, we propose a novel approach to construct a zero-shot model named WC-SBERT, aiming to achieve SOTA performance in zero-shot topic classification while addressing the aforementioned issues. The proposed approach to construct WC-SBERT can be divided into two stages:

(1) During the pre-training stage, the Wiki categories corresponding to the same Wiki pages are used to train a general SBERT-based model. The resultant model is general and it can be further fine-tuned to cater to specific downstream tasks.
(2) During the self-training stage, the aforementioned general model is further fine-tuned based on positively or negatively correlated categories and target labels. The correlation is established based on the similarity between target labels and Wiki pages.

The flowcharts of these two stages are shown in Figures 1 and 2, respectively. These two stages are detailed in the next two subsections.

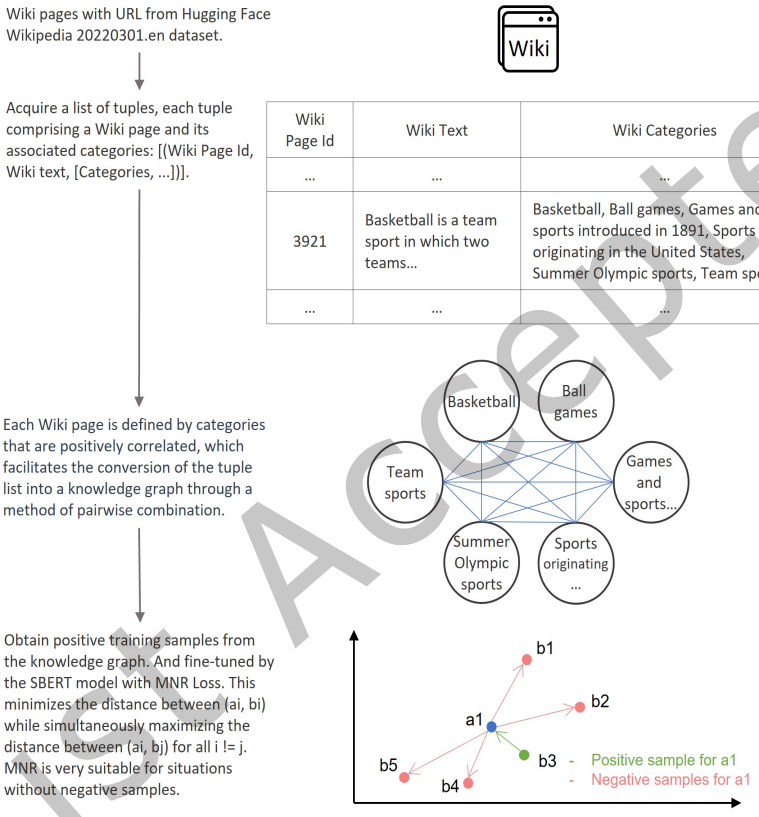

Fig. 1. Flowchart of pre-training for WC-SBERT, illustrating the process from Wikipedia dataset extraction to knowledge graph construction and the application of MNR Loss for obtaining positive and negative training samples for pre-training.

3.1 Stage 1: Pre-training

In the pre-training stage of WC-SBERT, we propose the use the categories of Hugging Face Wikipedia dataset to train a general model based on SBERT, which will be subsequently fine-tuned to a task-specific model in the second step of self-training. This method has two advantages:

- It is efficient since the categories are usually short keywords.

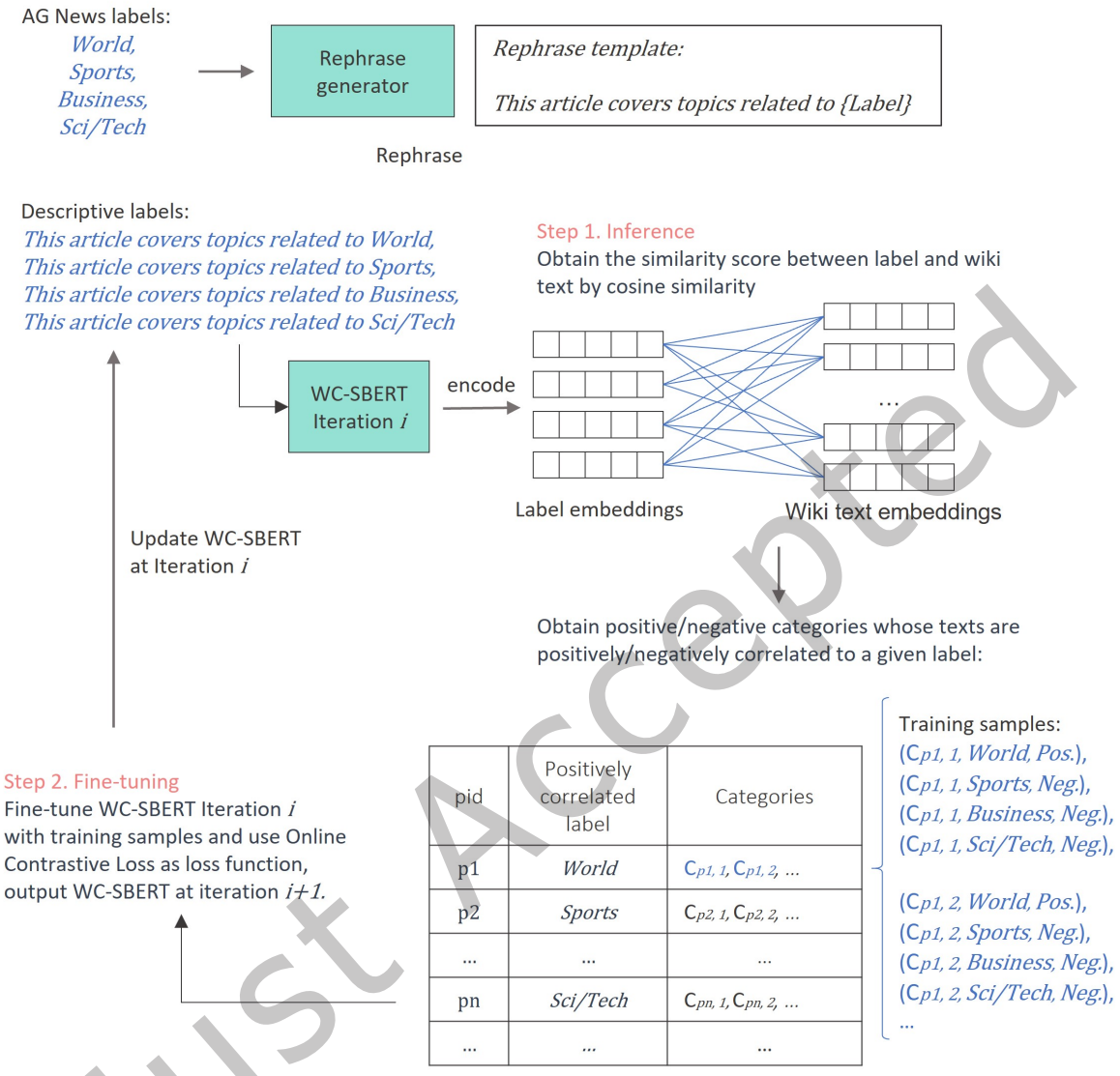

Fig. 2. Flowchart of self-training for WC-SBERT (using the AG News dataset as an example).

- The categories are numerous and quite diversified, which can serve as a balanced dataset for training a general-purpose of text classifiers.

The Wikipedia dataset is available at Hugging Face, but we still need to retrieve the categories of each Wiki pages with provided URLs to form the extended dataset *wiki-category* used in this study. Intuitively, categories corresponding to the same Wiki page are semantically associated, so they can be used to train the general model to acquire world knowledge. We can combine these associated categories in pairs (non-repetitive combinations

using Python's *itertools.combinations* using the pseudo code shown in Algorithm 1) to create the SBERT training dataset.

The steps involved in training set construction and loss function selection are detailed next.

(1) **Define training tuples**: Given a Wiki page p and its corresponding set of categories C_p, we can construct a set of page-category tuples as follows:

$$\{(p, C_p) \mid p \in P\}, \tag{1}$$

where P is a set of Wiki pages and C_p is the set of associated categories within page p. The set of all categories C is defined as

$$C = \cup_p C_p, p \in P. \tag{2}$$

(2) **Construct the knowledge graph**: From the above page-category tuples, we can construct a knowledge graph G where nodes represent categories and edges represent their associated same-page relationship. In other words, categories within C_p will be connected by an edge since they correspond to the same Wiki page.

$$G = (N, E) \text{ with } N = C \text{ and } E = \{(c_i, c_j) \mid c_i \text{ and } c_j \text{ belong to the same Wiki page}\} \tag{3}$$

Here, N is the set of categories, and E is the set of edges representing same-page association between categories

(3) **Generate training samples**: Positive training samples are generated from the connected nodes in the graph:

$$T = \{(c_i, c_j) \in E\} \tag{4}$$

That is, the set of training samples T consists of category pairs (c_i, c_j) connected by edges in the knowledge graph G.

(4) **Apply MNR loss**: To fine-tune the SBERT model, we use the Multiple Negative Ranking (MNR) Loss. This loss function helps to minimize the distance between positive pairs while maximizing the distance between non-associated pairs, ensuring that the model learns to distinguish between related and unrelated categories effectively:

$$\text{MNR Loss:} \quad \min_{\theta} \sum_{(a_i, b_i) \in T} [d_\theta(a_i, b_i) - \min_{b_j \neq b_i} d_\theta(a_i, b_j) + \text{margin}] \tag{5}$$

Through the training SBERT using same-page categories, we obtain a general-purpose base model, which is referred to as the pre-trained *WC-SBERT*. We shall further perform self-training on the pre-trained *WC-SBERT* using the target labels of the downstream task, as explained in the next subsection.

Algorithm 1 Pseudo code of generating SBERT training samples

```
train_samples ← []
for data in wiki_category do
    categories ← data["categories"]
    for pair in combinations(categories, 2) do
        train_samples.append(InputExample(texts=[pair[0], pair[1]]))
    end for
end for
```

3.2 Stage 2: Self-training

To further fine-tune the base model obtained in stage 1 for the downstream task, we use identify categories and labels that are either positively or negatively correlated, then use them for training, as shown in Figure 2. A category and a target label is defined as positively correlated if the target label is close enough to the category's Wiki page. Otherwise, they are negatively correlated. We can then use these positive/negative correlation between categories and labels to fine-tune the general model obtained in the pre-training stage. More specifically, there are two steps involved in this stage:

- **Inference step:** Given a Wiki page, we can use its texts to find similar target labels based on cosine similarity of their embeddings. (This can be done efficiently, as detailed later.) In notation, given a Wiki page p, its positively correlated labels can be defined as a set shown next:

$$L_p = \{l \mid cos(p, l) > t, p \in P \, l \in L\},$$

  where $cos(p, l)$ is the cosine similarity between the embeddings of Wiki page p and label l, P is the set of Wiki pages, L is the set of target labels, and t is the threshold of the cosine similarity score.
- **Fine-tuning step:** Positively and negatively associated categories of a given target label are identified based on the aforementioned similarity. These positive/negative training samples are used to train the base model using online contrastive loss []. In notation, the set of positively correlated categories and labels can be defined as a set shown next:

$$\{(c, l) \mid c \in C_p, l \in L_p, \forall p \in P\}.$$

  Similarly, the set of negatively correlated categories and labels can be defined as a set shown next:

$$\{(c, l) \mid c \in C_p, l \in L - L_p, \forall p \in P\}.$$

The above two steps are iterated until a stopping criterion is met. To make the comparison step more efficient, we can use the general base model to encode the first 200 words of each Wiki page (6,458,670 pages in total) and store its embedding in an h5 format [] for subsequent computation of cosine similarity. This can significantly reduce the computing time for cosine similarity. Note that this Wiki text embedding is only done once, based on the observation of Merchant et al. [13] which states that BERT-based models, when fine-tuned, do not suffer significant catastrophic effects on out-of-domain embeddings.

4 DATASETS

4.1 Training dataset

This study utilizes the Wikipedia dataset from Hugging Face for both pre-training and self-training of the proposed WC-SBERT. The dataset consists of a total of 6,458,670 records, with fields including Wiki page ID, URL, title, and text. The Wikipedia categories facilitate the organization of articles by topic, grouping similar articles together. Since this dataset does not include the categories for each page, a separate web scraping program is designed in this study to retrieve the categories for each page. A total of 1,563,194 categories are collected, and the combination of these categories with the original dataset is named the *wiki-category* dataset. (The authors have also uploaded these categories to Hugging Face.) Listing 1 shows an example of one page's data.

4.2 Target datasets for zero-shot evaluation

We have three target datasets as the downstream tasks for the evaluation of WC-SBERT for zero-shot topic classification.

- **AG News**: The AG News dataset [28] consists of news titles and descriptions, categorized into four classes: *World*, *Sports*, *Business*, and *Sci/Tech*. This study utilizes the AG News dataset from Hugging Face, with a

```
{
    ``id": 3921,
    ``url": ``https://en.wikipedia.org/wiki/Basketball",
    ``title": ``Basketball",
    ``text": ``Basketball is a team sport...",
    ``categories": ["Ball games", ``Team sports", ``..."]
}
```

Listing 1. Wikipedia sample data format

total of 7,600 data points in the test set. *Sci/Tech* is further divided into two classes (*Science* and *Technology*) for classification purposes.

- **Yahoo! Answers**: The Yahoo! Answers dataset [28] mainly comprises questions and answers from the Yahoo! platform. In this study, Yahoo! Answers dataset from Hugging Face is employed, containing 60,000 data points in the test set. There are ten main categories for the topics: *Society & Culture*, *Science & Mathematics*, *Health*, *Education & Reference*, *Computers & Internet*, *Sports*, *Business & Finance*, *Entertainment & Music*, *Family & Relationships*, and *Politics & Government*. Each label is individually treated as two labels for classification (split by the '&' character), and the output is mapped back to the original label.
- **DBpedia**: The DBpedia dataset [1] is derived from structured information extracted from Wikipedia. In this study, the DBpedia dataset *dbpedia_14* from Hugging Face is used as the experimental subject, consisting of 70,000 data points in the test set. It includes 14 distinct categories, namely *Company*, *EducationInstitution*, *Artist*, *Athlete*, *OfficeHolder*, *MeanOfTransportation*, *Building*, *NaturalPlace*, *Village*, *Animal*, *Plant*, *Album*, *Film*, and *WrittenWork*. We will split the partial classification into words by separating them with a space as input: *EducationInstitution* to *Education institution*, *OfficeHolder* to *Office holder*, *MeanOfTransportation* to *Mean of transportation*, *NaturalPlace* to *Nature place* and *WrittenWork* to *Written work*.

5 EXPERIMENTS

This study prioritizes conducting relevant experiments using the AG News dataset as the target dataset. The findings and techniques are then applied to the other two target datasets. The experiment setup are shown in Table 1.

Table 1. **The setup used our experiments.**

Parameters	**Value**
Original model	SBERT (all-mpnet-base-v2)
Batch size for training	256
Max sequence length (token length)	128
Epochs	1
Loss function for pre-training	Multiple Negative Ranking (MNR) Loss
Loss function for self-training	Online Contrastive Loss

5.1 Ablation Study

To measure the impact of different training stages, we conducted an ablation study. This subsection presents the performance comparison among the original SBERT (all-mpnet-base-v2), the WC-SBERT pre-trained model, and the WC-SBERT self-training model.

We used the AG News dataset with the same setup for all models to ensure a fair comparison. These models, as shown next, were evaluated on various metrics, including accuracy, precision, recall, and F1-score.

- **Original SBERT**: The original SBERT model is used as the baseline for comparison.
- **WC-SBERT with pre-training only**: This is the model obtained in the pre-training stage, as described in Section 3.1.
- **WC-SBERT**: The is the final WC-SBERT model after two stages of pre-training and self-training, as described in Section 3.2.

Table 2 shows the performance comparison among the original SBERT, the WC-SBERT with pre-training only, and the overall WC-SBERT.

Table 2. Performance comparison among the original SBERT, WC-SBERT with pre-training only, and WC-SBERT

Model	Accuracy	Precision	Recall	F1-score
Original SBERT	0.654	0.653	0.654	0.648
WC-SBERT with pre-training only	0.673	0.685	0.673	0.661
WC-SBERT	**0.735**	**0.766**	**0.735**	**0.730**

The results indicate that each stage of pre-training or self-training contributes to improved performance. In particular, WC-SBERT with pre-training only outperforms the original SBERT model across all metrics. Furthermore, the final WC-SBERT achieves the highest performance across all metrics, demonstrating that the proposed two-stage approach of pre-training and self-training based on Wiki categories is highly effectively for zero-shot topic classification.

In addition to the ablation study, we evaluated the performance of the models on the other two target datasets. For these datasets, we primarily used accuracy as the evaluation metric to maintain consistency. The results and comparisons are discussed in the following subsections.

5.2 Use of descriptive labels

In recent research, techniques such as prompt tuning [19] [22], including GPT instruction-tuning [27], have been shown to significantly enhance the accuracy of zero-shot or few-shot classification. SBERT, known for its contextual understanding capabilities, can leverage its strength to enhance the encoding ability of its sentence embeddings. Our objective is to explore whether prompt tuning can also improve the similarity matching in WC-SBERT, which is exclusively trained on the target label set. In this experiments, we used a simple rephrase generator that transforms the original label into a descriptive one (conceptually akin to a hard prompt in [11]) with the template: *"This article covers topics related to [label]"*. The experimental results indicate that using only the descriptive label can significantly enhance the predictive accuracy to 0.836, a 17.3% increase from 0.713 using the original labels. In the subsequent experiments, we shall use descriptive labels as the input for WC-SBERT during self-training and model inference.

5.3 Accuracy evaluation on parameters for self-training

In our experiments, two main parameters influence the accuracy of the self-training stage, that is, the similarity score threshold (t) and the iteration count. Here We attempt to observe the impact of different thresholds and iteration count on accuracy. As revealed in Figure 3, a threshold that is too low (t=0.75) results in an excessive number of positive correlation training samples in each iteration, leading to a significant increase in training samples. This not only increases the training duration but also degrades the accuracy. On the other hand, a higher threshold (where t=0.9 results in no training samples being obtained) may lead to an insufficient number of training samples, thereby inhibiting effective learning and causing the accuracy to plateau.

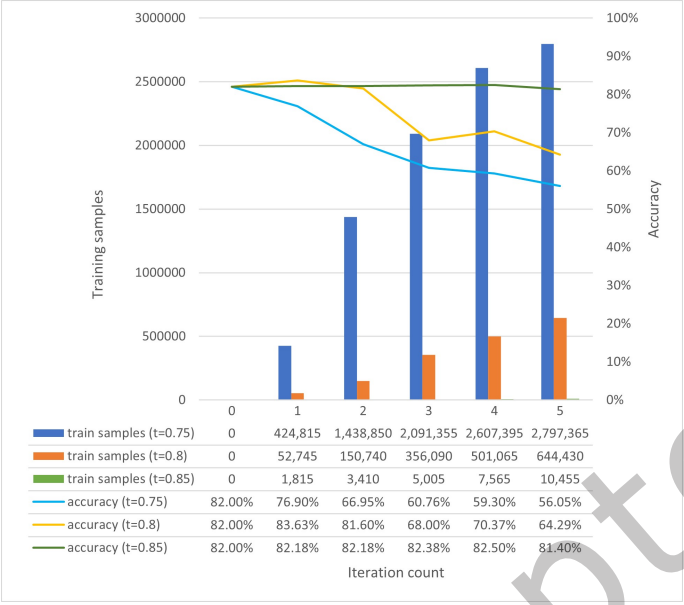

Fig. 3. Evaluation on AG News with various parameter settings for self-training

5.4 Self-training iterations analysis

Figure 3 indicates that the first iteration of self-training significantly improves the model's accuracy, while further iterations tend to degrade instead. This phenomenon can be attributed to several factors:

(1) **Accumulation of noise**: As the self-training process relies on the model's predictions to generate new training data, errors in these predictions can propagate and amplify over multiple iterations, leading to reduced accuracy.
(2) **Lack of novel information**: After the initial iteration, the model may not gain significant new information from additional, as the training data becomes fixed and overfiting occurs.

5.5 Descriptive labels for all datasets

Based on the aforementioned experiments on AG News, we have concluded that the optimal results for downstream tasks are obtained by using descriptive labels in conjunction with appropriate thresholds and iteration counts. We applied the same descriptive label template: *"This article covers topics related to [label]"* to both the Yahoo! Answers and DBPedia datasets, setting the threshold (t) to be greater than or equal to 0.7 and the iteration count to be 5 or fewer, yielded the highest accuracy for each dataset, as shown in Table 3.

Table 3. Evaluation results of different target datasets with descriptive labels

Dataset	**AG News**	**Yahoo! Answers**	**DBpedia**
Threshold (t)	0.85	0.85	0.85
Iteration (i)	1	1	1
Accuracy	0.836	0.637	0.747

In Table 3, both AG News and Yahoo! Answers have achieved state-of-the-art performance while DBPedia has not. Consequently, we conducted an error analysis of the prediction results for DBPedia using a confusion matrix shown in Table 4 to discover the potential reasons. In this table, out of 5,000 samples of *Animal*, 3,132 instances are misclassified as *Plant*, indicating there may be no clear distinction between animals and plants in the descriptions of Wikipedia pages. Moreover, some ambiguity may arise in some labels. For instance *Artist* can encompass professionals in various fields, including painters, musicians, and producers. *Album* refers to music-related albums, and *Film* may cover music and other audiovisual content as well. It is understandable that confusion and ambiguity may arise for these labels' definitions.

Table 4. **Confusion matrix for DBpedia.** The predicted labels are presented as columns, while the actual labels are presented in rows. The rightmost column displays the accuracy for each label.

Label index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	*Acc.*(%)
0 (Company)	2714	80	35	21	39	808	179	37	16	45	161	448	239	178	54.28
1 (EducationInstitution)	36	4737	38	2	20	7	28	19	42	20	24	6	14	7	94.74
2 (Artist)	46	47	955	9	90	1	14	12	5	37	24	2158	369	1233	19.10
3 (Athlete)	38	31	2	4555	41	153	12	2	57	9	5	20	9	66	91.10
4 (OfficeHolder)	131	76	8	16	4180	20	12	11	350	28	18	5	9	136	83.60
5 (MeanOfTransportation)	53	1	0	3	1	4758	52	82	12	28	2	1	5	2	95.16
6 (Building)	280	505	118	2	31	223	2093	337	1325	15	19	5	36	11	41.86
7 (NaturalPlace)	0	2	0	0	0	32	47	4792	125	0	2	0	0	0	95.84
8 (Village)	0	16	3	0	1	2	0	509	4465	0	4	0	0	0	89.30
9 (Animal)	0	0	0	0	0	0	0	724	0	1144	3132	0	0	0	62.64
10 (Plant)	8	0	0	0	0	2	0	2	0	1	4985	0	2	0	99.70
11 (Album)	1	0	2	1	0	0	0	0	0	1	0	4988	5	2	99.76
12 (Film)	2	0	1	0	0	1	0	0	1	19	0	76	4891	9	97.82
13 (WrittenWork)	290	103	111	4	206	21	35	39	139	435	201	116	254	3046	60.92

In earlier experiments, we realize the importance of descriptive labels on SBERT's for topic classification. As a result, we shall try to define a specific descriptive label for each label in DBpedia. In particular, we shall use a template **"This [description] described in this content is [label]"** and add explicit statements/keywords for the customized descriptions for each label, aiming to enhance discriminability by distinguishing the characteristics of each label. For example, the customized descriptive label of *Athelete* is **"This person who plays sport described in this content is an athlete"**. Other customized descriptive labels of DBpedia can be referred to Table 5. By using the customized descriptive labels, we can increase the accuracy from 0.747 to 0.881. The accuracy is good, but still not as good as the results achieved by Gera et al. which requires the text part of the downstream tasks. In contrast, our approach only use Wikipedia dataset and the customized descriptive labels to achieve a comparable performance.

Table 5. **Customized descriptive labels for DBpedia.**

Label	**Customized descriptive labels**
Company	This topic is describing this company
EducationInstitution	This school, university described in this content is an education institution
Artist	This musician, painter, singer, writer, author described in this content is an artist
Athlete	This person who plays sport described in this content is an athlete
OfficeHolder	This person who holds a position or office in a government described in this content is an officeholder
MeanOfTransportation	This vehicles, ridden, trains and other conveyances described in this content is transportation
Building	This man-made structure described in this content is a building
NaturalPlace	This natural landforms, bodies of water, vegetation, rocks, forests, rivers, lakes, mountains, oceans, grasslands described in this content is a natural place
Village	This town, small settlement or community described in this content is a village
Animal	This organism described in this content is an animal
Plant	This organism described in this content is a plant
Album	This music or recorded tracks described in this content is an album
Film	This movie described in this content is a film
WrittenWork	This books, essays, poems or literatures described in this content is a written work

After applying the customized descriptive labels, we observed a significant reduction in the number of instances initially misclassified as plant instead of animal, decreasing from 3,132 to 487, as shown in Table 6. This indicates the effectiveness of the customized descriptive labels for zero-shot topic classification.

Table 6. **Confusion Matrix for DBpedia with Descriptive Labels.** The predicted labels are presented as columns, while the actual labels are presented in rows. The rightmost column displays the test accuracy for each label after applying descriptive labels.

Label index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	*Acc.*(%)
0 (Company)	3382	99	120	8	40	525	121	24	12	36	67	283	148	135	67.64
1 (EducationInstitution)	34	4850	28	1	18	1	22	3	18	7	10	0	5	3	97.00
2 (Artist)	67	19	3687	27	144	4	20	6	2	14	7	181	51	771	73.74
3 (Athlete)	7	1	11	4816	130	14	1	0	18	0	0	1	0	1	96.32
4 (OfficeHolder)	50	40	46	8	4814	5	5	2	6	4	0	0	0	20	96.28
5 (MeanOfTransportation)	333	0	1	0	1	4639	10	6	1	4	2	0	1	2	92.78
6 (Building)	145	330	20	0	15	100	3976	61	311	13	4	1	2	22	79.52
7 (NaturalPlace)	0	0	0	0	0	6	87	4624	281	1	0	0	0	1	92.48
8 (Village)	0	16	4	0	4	0	7	585	4379	1	0	0	0	4	87.58
9 (Animal)	2	0	2	29	0	12	0	292	0	4175	487	0	0	1	83.50
10 (Plant)	8	0	0	0	0	2	1	83	0	15	4890	0	1	0	97.80
11 (Album)	1	0	12	1	0	0	0	0	0	0	0	4974	6	6	99.48
12 (Film)	8	1	9	2	0	2	4	0	0	4	0	105	4805	60	96.10
13 (WrittenWork)	543	109	30	4	47	15	12	31	105	147	123	40	141	3653	73.06

We also applied the customized descriptive labels to both AG News and Yahoo! Answers by using specific descriptions for some labels. Specially, to increase the distinction between two ambiguous categories, we injected reverse implications by using the term "not." For example, in the case of AG News, we modified the original prompt to "This topic is talk about World, not Business." All customized descriptive labels are presented in Table 12 and 13 in the appendix. The accuracy improves accordingly, as shown in Table 7.

Table 7. Accuracy of the original and customized descriptive labels.

Dataset	**AG News**	**Yahoo! Answers**	**DBpedia**
Original descriptive label	0.836	0.637	0.747
Customized descriptive label	**0.840**	**0.638**	**0.881**

5.6 Efficiency comparison of self-training

The proposed light self-training is an iterative procedure, where each iteration consists of two steps: inference on the target labels (to obtain Wiki pages of similar texts) and and fine-tuning using the categories of positively correlated Wiki pages. Here we compare the efficiency between WC-SBERT and the current best model in Gera et al. [8] on three datasets, where the iteration count is 2.

To ensure the experiment's fairness, all tests are conducted using the same hardware environment with the following specifications: CPU: Intel(R) Xeon(R) Gold 6154 (8 cores), GPU: NVIDIA Tesla V100 * 2, RAM: 128 GB, OS: Ubuntu 20.04 LTS. The time costs are presented in the following tables:

- Table 8: Time cost comparison during the inference step.
- Table 9: Time cost comparison during the the self-training step.
- Table 10: Overall time cost comparison.

From these comparison tables, we can observe that WC-SBERT demonstrates significant improvements in efficiency in both steps in self-training:

Table 8. **Time cost during the inference step in self-training.**

Dataset	Model	Iteration 0		Iteration 1		Total Time	Time / Samples
		Time (sec.)	Samples	Time (sec.)	Samples	(sec.)	(sec.)
AG News	Self-Gera	204	7,600	207	7,600	411	0.027
	WC-SBERT	63	6,458,670	64	6,458,670	127	9.832×10^{-6}
DBPedia	Self-Gera	7,939	70,000	8,077	70,000	16,016	0.114
	WC-SBERT	79	6,458,670	83	6,458,670	162	1.254×10^{-5}
Yahoo! Answers	Self-Gera	13,201	58,966	11,525	58,966	24,726	0.210
	WC-SBERT	80	6,458,670	82	6,458,670	162	1.254×10^{-5}

Table 9. **Time cost during the fine-tune step in self-training.**

Dataset	Model	Iteration 0		Iteration 1		Total Time	Time / Samples
		Time (sec.)	Samples	Time (sec.)	Samples	(sec.)	(sec.)
AG News	Self-Gera	110	800	106	800	216	0.135
	WC-SBERT	174	52,745	274	150,740	448	0.002
DBPedia	Self-Gera	395	2,800	391	2,800	786	0.140
	WC-SBERT	165	25,172	150	29,036	315	0.006
Yahoo! Answers	Self-Gera	221	2,000	218	2,000	439	0.110
	WC-SBERT	159	37,188	155	40,248	314	0.004

Table 10. **Self-training performance comparison results between WC-SBERT and Self-Gera.** The Self-training time is the result of adding the total time for each dataset in Table 8 and Table 9.

Dataset	Metric	Self-Gera	WC-SBERT	Time ratio (Self-Gera / WC-SBERT)
AG News	Self-training Time (sec.)	627	575	**1.09**
	Inference Time / Sample (sec.)	0.027	9.832×10^{-6}	**2,746.14**
	Fine-tune Time / Sample (sec.)	0.135	0.002	**67.5**
DBPedia	Self-training Time (sec.)	16,802	477	**35.22**
	Inference Time / Sample (sec.)	0.114	1.254×10^{-5}	**9,090.91**
	Fine-tune Time / Sample (sec.)	0.140	0.006	**23.33**
Yahoo! Answers	Self-training Time (sec.)	25,165	476	**52.86**
	Inference Time / Sample (sec.)	0.210	1.254×10^{-5}	**16,746.41**
	Fine-tune Time / Sample (sec.)	0.110	0.004	**27.5**

- During the inference step, WC-SBERT employs 6.45 million wiki text samples for inference, achieving a remarkable reduction in time across different datasets, with an efficiency improvement factor of 2,746 to 16,746 times.
- In the fine-tuning step, WC-SBERT utilizes Wiki categories for fine-tuning, and despite the larger training dataset, WC-SBERT's overall fine-tuning time remains lower than that of Self-Gera, resulting in an efficiency boost of 23.33 to 67.5 times.

In summary, Self-Gera's self-training time is also 1.09 to 52.86 times longer than WC-SBERT.

5.7 GPT experiment

In recent research, it has been demonstrated that LLMs, such as GPT, are highly effective in comprehending and addressing classification problems [15]. We have also used the currently popular GPT-3.5-turbo model provided by OpenAI API to test the effects of zero-shot topic classification. To inference directly on AG News, we used the prompt template shown in Listing 2.

```
There are 4 classes below:
    World, Sports, Business, Sci/Tech
This text is belong to which class: {Text}
```

Listing 2. Prompt template for GPT 3.5 on AG News

Using the aforementioned prompt for inference on the GPT-3.5 model, we obtained an accuracy of 0.710, which is not quite satisfactory. More discussion about the results are presented in next subsection.

Moreover, we also tried to use GPT-3.5-turbo and its fine-tune provided by OpenAI API. However, due to its high cost, we only used 6,700 Wikipedia entries (split into 5,700 entries for training and 1,000 for validation) to fine-tune the GPT-3.5-turbo model. We then randomly selected 300 entries from the Yahoo! Answers as a test set. If we fine-tune the GPT-3.5-turbo model with all 5,700 training data entries at once, the resulting test accuracy is 0.55, which is far from state of the art. However, OpenAI API also offers the feature to fine-tune an already fine-tuned model, meaning that you can fine-tune a model a second time with a smaller dataset. By dividing the 5,700 entries into 5,000 for the first round of fine-tuning and 700 for the second round, we achieved a test accuracy of 0.62. Of course, due to the high cost of using the Curie model, we could only test on a few hundred entries. In summary, this GPT fine-tuning approach is quite challenging when applied to large-scale datasets for zero-shot topic classification.

5.8 Analysis and conclusions of experiments

From the above experiments, a summary of analysis and conclusions can be drawn here.

- SBERT-based similarity models, when integrated with labels as the objective for both pre-training and self-training, continue to demonstrate sensitivity to context. Employing descriptive labels results in enhanced accuracy.
- For the accuracy of self-training, selecting an appropriate threshold (t) is instrumental in obtaining an optimal number of training samples, which in turn contributes to an increase in accuracy. Generally, the best accuracy is achieved within 1 to 2 iterations.
- Compared to the current best model in Gera et al. [8], WC-SBERT demonstrates significant improvements in efficiency in both steps in self-training, as shown in Tables 8, 9, and 10.
- GPT-3.5 does not achieve satisfactory accuracy even after fine-tuning. Moreover, its cost is higher compared to other models, which makes it not suitable for zero-shot topic classification.

Table 11. Accuracy comparison on three target datasets using the proposed model WC-SBERT and three baselines of Ding et al. [7], Gera et al. [8], and GPT-3.5. (Due to the cost of GPT-3.5, we do not perform GPT-3.5 inference on the DBPedia test set, which consists of a total of 70,000 samples.)

Dataset	AG News	Yahoo! Answers	DBpedia
WC-SBERT	**0.840**	**0.638**	0.881
Wiki-Ding	0.796	0.573	0.902
Self-Gera	0.814	0.620	**0.945**
GPT-3.5	0.710	0.597	N/A

A list of accuracy on three target datasets using the proposed model WC-SBERT and three baselines of Ding et al. [7], Gera et al. [8], and GPT-3.5 are shown Table 11 for easy comparison. The proposed model WC-SBERT achieves SOTA on the first two datasets, but not the third. The potential reasons for not achieving SOTA on the DBpedia dataset can be analyzed as follows.

(1) **Ambiguity in label definitions**: The DBpedia dataset contains labels with ambiguous definitions, leading to confusion during classification. For instance, categories like "Artist" and "Album" can overlap significantly, causing the model to misclassify entries. In our experiment using descriptive labels, we observed that many errors were due to these overlapping and unclear label definitions. For example, the label "NaturalPlace" in DBpedia includes a wide range of entities from forests to rivers, which can be challenging for the model to classify accurately. The presence of such diverse and overlapping categories contributes to the lower performance on this dataset.
(2) **Alignment with true zero-Shot tasks**: Compared to traditional self-training, our results are less favorable since traditional self-training involves fine-tuning with both target labels and unlabeled texts, which allows traditional self-training to have a certain degree of task understanding. In contrast, WC-SBERT is more general-purpose and adheres more closely to the true spirit of zero-shot tasks, where the model does not see any texts from the target task during training.

6 CONCLUSIONS AND FUTURE WORK

This paper introduces WC-SBERT, a zero-shot topic classification model which leverages a Wikipedia dataset to align closely with zero-shot scenarios. The proposed WC-SBERT can achieve SOTA performance on AG News and Yahoo! Answers datasets. It is also more efficient then previous approaches using self-training. The success of WC-SBERT can be largely attributed to the following factors:

- Use of Wiki categories that appear in the same Wiki page for pre-training.
- Use of Wiki categories that are positively/negatively correlated to a target label for self-training.
- Use of customized description labels for enhancing accuracy.

Despite achieving outstanding performance on effectiveness and efficiency in zero-shot topic classification, the proposed approach still has bottlenecks and challenges that need further exploration in the future, as explain next.

- **Threshold selection**: Choosing an appropriate threshold (t) for the self-training stage is crucial for model accuracy. We observed in our experiments that different thresholds significantly impact the model accuracy. However, we have not yet effectively determined the optimal threshold. Future work will focus on developing automated methods to determine the best threshold.
- **Descriptive label design**: The design of descriptive labels significantly affects the classification performance of the model. We found that for some labels (e.g., "NaturalPlace" in DBpedia), ambiguous label definitions lead to misclassification. We will further investigate how to optimize the design of descriptive labels to reduce ambiguity and increase model accuracy.
- **Dataset representativeness**: Our study primarily uses Wikipedia data for pre-training, but whether this data is sufficiently generalizable to be applied to various tasks remains uncertain. For one thing, Wikipedia seldom has expressive texts with emotions. Therefore, there is room for improvement when applying WC-SBERT to downstream tasks such as sentiment analysis. For instance, for sentiment analysis tasks involving consumer reviews and emotions, such as those from IMDB and Amazon, we may need to use emotion-rich datasets to create a general-purpose zero-shot model as the pre-trained model for WC-SBERT.

ACKNOWLEDGMENTS

The authors would like to thank the support from the Featured Area Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (113L900901/113L900902/113L900903), Taiwan.

REFERENCES

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web*, Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 722–735.

[2] Lorenzo Bongiovanni, Luca Bruno, Fabrizio Dominici, and Giuseppe Rizzo. 2023. Zero-Shot Taxonomy Mapping for Document Classification. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*. 911–918.

[3] Ming-Wei Chang, Lev Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of Semantic Representation: Dataless Classification. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2* (Chicago, Illinois) *(AAAI'08)*. AAAI Press, 830–835.

[4] Qianben Chen, Richong Zhang, Yaowei Zheng, and Yongyi Mao. 2022. Dual Contrastive Learning: Text Classification via Label-Aware Data Augmentation. *CoRR* abs/2201.08702 (2022). arXiv:2201.08702 https://arxiv.org/abs/2201.08702

[5] Zewei Chu, Karl Stratos, and Kevin Gimpel. 2020. Natcat: Weakly Supervised Text Classification with Naturally Annotated Datasets. *CoRR* abs/2009.14335 (2020). arXiv:2009.14335 https://arxiv.org/abs/2009.14335

[6] 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[7] Hantian Ding, Jinrui Yang, Yuqian Deng, Hongming Zhang, and Dan Roth. 2022. Towards Open-Domain Topic Classification. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*. Association for Computational Linguistics, Hybrid: Seattle, Washington + Online, 90–98. https://doi.org/10.18653/v1/2022.naacl-demo.10

[8] 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, Abu Dhabi, United Arab Emirates, 1107–1119. https://aclanthology.org/2022.emnlp-main.73

[9] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652* (2017).

[10] Chaoqun Liu, Wenxuan Zhang, Guizhen Chen, Xiaobao Wu, Anh Tuan Luu, Chip Hong Chang, and Lidong Bing. 2023. Zero-Shot Text Classification via Self-Supervised Tuning. arXiv:2305.11442 [cs.CL]

[11] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 61–68.

[12] 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 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[13] Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to bert embeddings during fine-tuning? *arXiv preprint arXiv:2004.14448* (2020).

[14] Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *Proceedings of Workshop at ICLR* 2013 (01 2013). https://www.researchgate.net/publication/234131319_Efficient_Estimation_of_Word_Representations_in_Vector_Space

[15] Alejandro Peña, Aythami Morales, Julian Fierrez, Ignacio Serna, Javier Ortega-Garcia, Iñigo Puente, Jorge Cordova, and Gonzalo Cordova. 2023. Leveraging large language models for topic classification in the domain of public affairs. In *International Conference on Document Analysis and Recognition*. Springer, 20–33.

[16] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[17] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019). https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf

[18] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. http://arxiv.org/abs/1908.10084

[19] Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676* (2020).

[20] Sonish Sivarajkumar and Yanshan Wang. 2022. HealthPrompt: A Zero-shot Learning Paradigm for Clinical Natural Language Processing. arXiv:2203.05061 [cs.CL]

[21] 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, Abu Dhabi, United Arab Emirates, 7508–7520. https://aclanthology.org/2022.emnlp-main.509

[22] Yuki Wang, Wei Wang, Qi Chen, Kaizhu Huang, Anh Nguyen, and Suparna De. 2023. Prompt-based Zero-shot Text Classification with Conceptual Knowledge. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, Vol. 4. Association for Computational Linguistics, 30–38.

[23] 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, Abu Dhabi, United Arab Emirates, 7042–7055. https://aclanthology.org/2022.emnlp-main.474

[24] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. 2021. A Survey on Deep Semi-supervised Learning. *CoRR* abs/2103.00550 (2021). arXiv:2103.00550 https://arxiv.org/abs/2103.00550

[25] 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*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf

[26] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for Longer Sequences. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 17283–17297. https://proceedings.neurips.cc/paper_files/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf

[27] Ruohong Zhang, Yau-Shian Wang, and Yiming Yang. 2023. Generation-driven Contrastive Self-training for Zero-shot Text Classification with Instruction-tuned GPT. arXiv:2304.11872 [cs.CL]

[28] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf

[29] Yang Zou, Zhiding Yu, B.V.K. Vijaya Kumar, and Jinsong Wang. 2018. Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-Training. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

A APPENDIX

Table 12. **Description of labels on AG News.**

Label	Descriptive labels
World	This article covers topics related to World not Business
Sports	This article covers topics related to Sports
Business	This article covers topics related to Business not World
Science	This article covers topics related to Science
Technology	This article covers topics related to Technology

Table 13. **Description of labels on Yahoo! Answers.** We only provided descriptions for a few categories in our analysis.

Label	Descriptive labels
Society	This topic is talk about Society not Family or Relationships
Education	This topic is talk about Education not Science or Mathematics