Relevance, Diversity, and Exclusivity: Designing Keyword-augmentation Strategy for Zero-shot Text Classifiers

Taro Yano, Kunihiro Takeoka, Masafumi Oyamada
Data Science Laboratories, NEC Corporation
{taro_yano, k_takeoka, oyamada}@nec.com

Abstract

Zero-shot text classification involves categorizing text into classes without labeled data, typically using a pre-trained language model to compute the correlation between text and class names. This makes it essential for class names to contain sufficient information. Existing methods incorporate semantically similar keywords related to class names, but the properties of effective keywords remain unclear. We demonstrate that effective keywords should possess three properties: 1) keyword relevance to the task objective, 2) inter-class exclusivity, and 3) intra-class diversity. We also propose an automatic method for acquiring keywords that satisfy these properties without additional knowledge bases or data. Experiments on nine real-world datasets show our method outperforms existing approaches in fully zero-shot and generalized zero-shot settings. Ablation studies further confirm the importance of all three properties for superior performance.

1 Introduction

Zero-shot text classification is the process of categorizing text into classes without any training data, which is essential in scenarios where creating a large amount of labeled data is impractical. To this end, most zero-shot classification techniques utilize signals that indicate the relationship between each instance and class, such as semantic textual similarity between instances and class names (Sappadla et al., 2016; Yin et al., 2019) or the contextual word co-occurrence of the instance and the class name found in large language models like BERT (Schick and Schütze, 2021) and T5 (Sanh et al., 2022; Wei et al., 2022).

The performance of zero-shot classifiers is heavily influenced by *keywords* related to each class (including the class name itself), as these classifiers use the keywords as queries to compute the similarity between each instance and class. For example, PET (Schick and Schütze, 2021) employs a masked language model like BERT to estimate the class of a text instance by synthesizing a sentence from the text using a template such as "${text} This text is about [MASK].", calculating token probabilities at the masked position, and aggregating the probabilities of keywords related to each class (e.g., token "news" for class "News" and token "finance" for class "Economics"). This *class to related keywords mapping* is sometimes referred to as a *verbalizer* (Schick and Schütze, 2021). Since determining optimal keywords for each class is hard, several works tried to determine proper related keywords for classes using external sources such as knowledge graphs (Hu et al., 2022) or language models (Zhao et al., 2023; Shi et al., 2022).

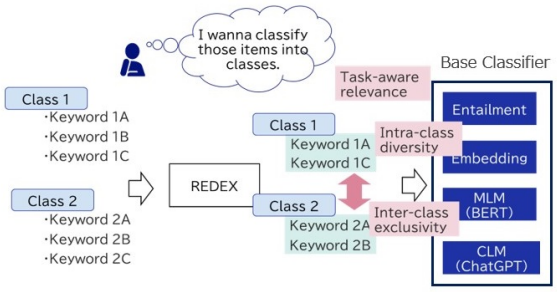

Figure 1: Overview of the proposed method REDEX. Zero-shot text classification needs proper assignment of keywords on each class. REDEX considers three properties regarding the nature of classification to assign the optimal keywords.

Regardless of whether it is manual or automatic, conventional ways to determine related keywords of each class often overlook *the nature of classification*. **(1) Keywords Relevance to the Objective:** First, the keywords attached to each class should be relevant to the classification objective, while the conventional method always attaches the same keywords for the same class name. For example, a class-keywords mapping { "Beauty" → ("mascara", "lipstick") } is suitable for product classification in E-commerce but may not be the best fit

*Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, pages 106 - 119

for movie classification. **(2) Intra-class Diversity of Keywords:** Second, the related keywords for a class should cover as broad a range of concepts as possible. Existing methods do not always consider the diversity of keywords within a class. **(3) Inter-class Exclusivity of Keywords:** Third, the related keywords for each class should be as distinct as possible, ensuring that two or more classes do not share similar keywords. For instance, the classes "Food" and "Cell Phone" might both have the keyword "apple," which can confuse zero-shot classifiers. Existing methods can produce such confusing class-keyword mappings because the keyword assignment for each class is performed independently.

In this paper, we explore the strategy of identifying optimal keywords for classes in zero-shot classifiers, considering the three properties mentioned above. Through extensive experiments, we found that considering all properties is necessary for obtaining better zero-shot classification performance in popular classifiers. To generate the optimal keywords automatically, we propose a new *generate-then-rerank* framework *REDEX* (RElevance, Diversity, EXclusivity) for keyword generation based on the concept of *maximal marginal relevance (MMR)* (Carbonell and Goldstein, 1998), which is often used in information retrieval. The extensive experiments demonstrate the effectiveness and versatility of the proposed method as it improved the performance of two types of state-of-the-art zero-shot classifiers drastically without any modifications to those methods (Zhang et al., 2022b; Yin et al., 2019; Zhang et al., 2022a; Geng and Liu, 2023; Holtzman et al., 2021) across all zero-shot settings, including generalized zero-shot text classification (GZTC) and fully-zero-shot text classification.

Our main contributions are as follows.

- We propose an automatic class-keyword mapping generation method *REDEX*, which generates keyword candidates by a generative language model and reranks them by considering three keyword properties: relevance to the objective of the classification, intra-class diversity of keywords, and inter-class exclusivity the keywords.

- Extensive experiments of REDEX for state-of-the-art zero-shot classifiers of fully or generalized zero-shot text classification in various domain datasets confirmed the effectiveness and versatility.

2 Proposed Method

2.1 Problem Setting

Zero-shot text classification is a task to estimate the optimal class $y_i \in \mathcal{K}$ of a test instance x_i, where $\mathcal{K} = \{1, 2, \ldots, K\}$ represents indices of all target classes. This paper assumes two types of zero-shot text classification: fully zero-shot setting and generalized zero-shot setting. The fully zero-shot setting provides only target class names to classify texts. The generalized zero-shot setting is where labeled data are available for a subset of target classes called seen classes, while those are not for the rest of the target classes called unseen classes. We assume that additional information, such as knowledge bases or unlabeled corpus, is unavailable.

2.2 Overview

Our method *REDEX* automatically finds keywords for each target class $k \in \mathcal{K}$ to improve classification performances. Through our experiments in Section 3, we found valuable keywords in enhancing the performance should possess three properties simultaneously: the semantic relatedness to class names, the intra-class diversity, and the inter-class exclusivity. The properties represent that keywords for a class should be not only related to the class name but also be diverse to cover features of instances belonging to the class and be semantically distant from keywords of the other classes to avoid misclassification.

Figure 2 illustrates our method, which generates keyword candidates for each class and reranks them to find valuable keywords with the aforementioned properties. The first step generates keyword candidates from a generative language model to obtain diverse and task-aware candidates without auxiliary information. The second step reranks keyword candidates to select keywords with the desired properties: semantic relatedness, intra-class diversity, and inter-class exclusivity.

2.3 Keyword Candidate Generation

In the first step of our method, we use a generative language model and prompting to generate keyword candidates. Compared to the conventional methods (Hu et al., 2022; Meng et al., 2020b) that find keywords from a knowledge base or in-domain

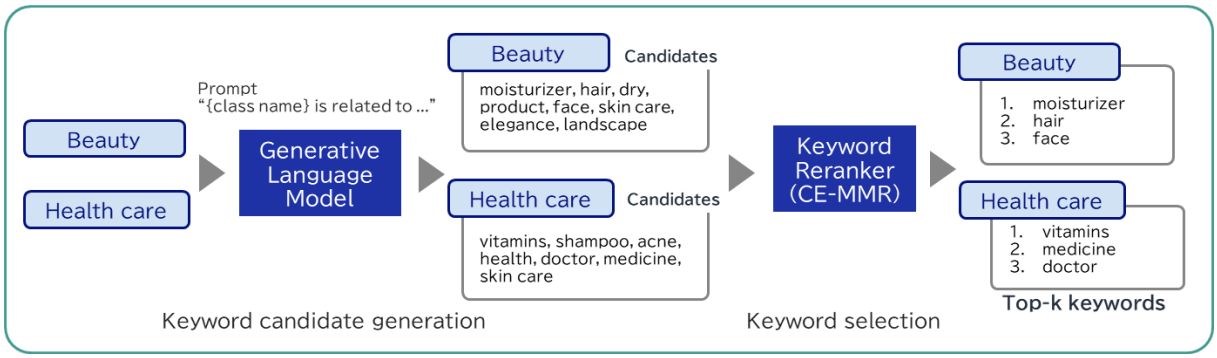

Figure 2: Overview of the proposed method REDEX for generating keyword candidates by a generative language model and reranking keyword candidates to select the suitable keywords for each class.

unlabeled data, our approach does not require any additional auxiliary information.

We manually construct prompts to input the model, such as "{class name} is related to"[1]. We then sample 20 texts using a generative language model and Nucleus Sampling (Holtzman et al., 2020). In our preliminary experiments, generating texts of more than 20 did not change most of the selected keyword candidates. We then extract phrases from generated texts by their term frequencies to acquire keyword candidates for each class. We select three times as many keyword candidates as the final number of target keywords. For details on hyperparameters and templates for generating texts, see Appendix A.

The proposed method can generate appropriate keywords by designing prompts depending on problem settings. In generalized zero-shot text classification, our method generates task-aware keywords for unseen classes using prompts that demonstrate task-aware keywords of seen classes. For instance, the task-aware keywords in the "Beauty" class in a product classification are "mascara" and "lipstick", and "elegance" and "landscapes" in a movie classification. We extract task-aware keywords for seen classes from labeled data using TF-IDF.

2.4 Reranking Keywords

Given sets of keyword candidates for classes $V = \{V_k\}_{k=1}^{|\mathcal{K}|}$, we rerank them to select suitable keywords P_k for each class k. While keyword candidates semantically relate to each class, without reranking candidate keywords, we do not capture the other properties of desirable keywords: the intra-class diversity of keywords for robust classification and the inter-class exclusivity of keywords

[1]The prompts to generate keyword candidates used in our experiments list in Appendix A.3.

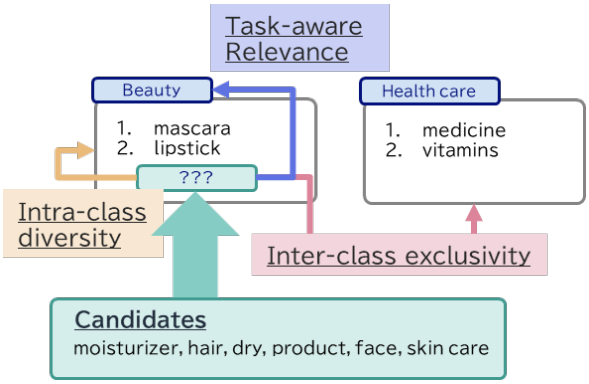

Figure 3: CE-MMR determines keywords for each class from its candidates incrementally in order of rank.

for preventing misclassification. To ensure these features of keywords, we propose class-exclusive maximal marginal relevance (CE-MMR) that extends maximal marginal relevance (MMR) for document retrieval to class-keyword reranking.

To consider the intra-class diversity, one can use maximal marginal relevance (MMR) (Carbonell and Goldstein, 1998), which reranks documents $\{d\}$ for a query q. MMR incrementally determines the rank of documents from top to bottom by the following scoring function:

$$S(d, q, R) = \lambda_1 \, s(d, q) - \lambda_2 \max_{d' \in R} s(d, d'), \quad (1)$$

where $s(d_1, d_2)$ is a function that returns a similarity of d_1 and d_2, R is a list of reranked documents, and $\lambda_1, \lambda_2 \in [0, 1]$ are hyperparameters controlling importance of the diversity of ranked documents and satisfy $\lambda_1 + \lambda_2 = 1$. This approach can be mapped to reranking keywords with their diversity. Considering a query and documents as a class name and its keyword candidates, MMR can be applied to the keyword reranking task. The formulation is

Algorithm 1 Reranking keywords for all classes

Require: $\mathcal{C}, V$
Ensure: P

```
INITIALIZE ∀k, P_k ← list()
for rank = 1 → max_{k∈K} |V_k| do
    for k ∈ K do
        Select p_k^rank in a deterministic way by
        arg max_{v_k ∈ V_k \ P_k} S*(c_k, v_k, {P_k'}_{k'=1}^K)
        Append p_k^rank to P_k
    end for
end for
return P
```

as follows:

$$S(c_k, v_k, P_k) = \lambda_1 s(c_k, v_k) - \lambda_2 \max_{p_k \in P_k} s(v_k, p_k), \quad (2)$$

where c_k denotes the class name of k, $v_k (\in V_k \backslash P_k)$ does a keyword candidate for class k except for P_k, and P_k does the reranked keywords of class k. Using this extended MMR, we can incrementally rerank keywords to preserve the diversity of keywords for each class and class-keyword relevance.

However, the method does not consider the inter-class exclusivity of keywords in reranking. To prevent misclassification due to assigning a similar keyword to multiple classes, we use CE-MMR, which adds the inter-class exclusivity of keywords into the above method, as illustrated in Figure 3. Put the last term for inter-class exclusivity (marked in red), the scoring function of CE-MMR

$$S^\star(c_k, v_k, P) = \alpha\, s(c_k, v_k) - \beta \max_{p_k \in P_k} s(v_k, p_k) - \gamma \max_{k' \in \mathcal{K} \backslash k} \max_{p_{k'} \in P_{k'}} s(v_k, p_{k'}), \quad (3)$$

where α, β, and γ are hyperparameters for controlling the importance of the class-keyword relatedness, intra-class diversity, and inter-class exclusivity and satisfy $\alpha + \beta + \gamma = 1$.

For reranking class keywords with the CE-MMR scoring function, we take a greedy reranking approach as shown in Algorithm 1. This algorithm repeats the following steps: calculating scores for keywords, appending the top-scored keyword for a class to a list of reranked keywords for the class, and removing the keyword from candidates.

3 Experiments

3.1 Zero-shot Text Classification

We conduct fully zero-shot experiments to demonstrate the effectiveness of our method.

3.1.1 Experimental Setup

Datasets. We use widely used benchmark datasets for topic classification and sentiment analysis. Topic classification datasets are **AG News** (Zhang et al., 2015), a collection of news articles and their topic categories, **DBpedia** (Lehmann et al., 2015) consisting of contents and their ontology classes, and **Yahoo** (Zhang et al., 2015), a collection of question-answer pairs and their topic categories. Sentiment analysis datasets are Stanford Sentiment Treebank (**SST2**) (Socher et al., 2013), a widely used benchmark, and Rotten Tomatoes (**RT**) (Pang and Lee, 2005), a collection of movie reviews and their sentiments. Statistics of datasets are shown in Appendix A.1.

Preprocessing. We use the same class names and prompt templates as the previous work Shi et al. (2022); Min et al. (2023, 2022) described in Appendix A.1. For datasets of more than 3,000 instances, due to limited computational resources, we run the experiment for three times with a randomly selected subset of 3,000 with different seeds, as in prior work (Zhao et al., 2021; Lyu et al., 2022).

Evaluation Metrics. We use accuracy to evaluate methods as in Zhao et al. (2023).

3.1.2 Compared Methods

OPT-6.7b (Zhang et al., 2022a) and **OpenLLaMA-7b** (Geng and Liu, 2023; Computer, 2023) are baseline methods that classify texts using next-token prediction with score calibration (Zhao et al., 2021; Holtzman et al., 2021) and length normalization of log-likelihood (Brown et al., 2020) techniques to improve classification accuracy as in Min et al. (2022); Holtzman et al. (2021). Also, as a compared method, we utilize NPPrompt (Zhao et al., 2023) (indicated by **w/ NPPrompt** in tables) that selects top-k similar keywords to class names from the vocabulary of the language models based on cosine similarities of token embeddings. We experimented with two variants of NPPrompt, one using the same vocabulary and embedding vectors as the base model and the other using `roberta-large` vocabulary and embedding vectors as in Zhao et al. (2023), and adopted `roberta-large`, which

Table 1: Performance on zero-shot text classification. The best scores are marked in bold. OPT and OpenLLaMA with keywords selected by our method outperform methods without keywords and by NPPrompt.

Method	AG News	DBpedia	Yahoo	SST-2	RT	Avg.
OPT-6.7b	75.8	**50.7**	33.7	55.1	58.8	54.8
w/ NPPrompt	79.6	44.9	45.9	49.8	51.8	54.4
w/ Ours	**79.7**	49.4	**49.5**	**68.5**	**69.3**	**63.3** (↑ 8.5)
OpenLLaMA-7b	**65.7**	36.1	**45.1**	74.7	70.4	58.4
w/ NPPrompt	65.3	40.7	38.8	50.9	50.0	49.1
w/ Ours	61.9	**51.3**	36.9	**77.5**	**72.7**	**60.1** (↑1.7)

Table 2: Case studies of zero-shot text classification experiments using the Yahoo dataset. Keywords in a bold font have the largest scores in the correct class.

Text	Method	Prediction	Keywords for the Correct Class
what is the name the cartoon about the french cats?	w/ NPPrompt	politics	ent, ENT, ents, enting
	w/ Ours	✓entertainment	**cartoon**, theater, sport
Please answer this chem problem for me?	w/ NPPrompt	society	science, Science, scientific, technology
	w/ Ours	✓science	**chemistry**, iphone, scientist, experiment

showed better performances. Our method (indicated by **w/ Ours** in tables) generates keyword candidates by corresponding language models and reranks them to use in inference. In our reranking, we use the cosine similarity of `roberta-large` embedding vectors as the similarity $s(\cdot, \cdot)$. We set the number of keywords to five for w/ NPPrompt and w/ Ours. As another hyperparameters of reranking, we set $\alpha = \beta = \gamma = 1/3$ for w/ Ours because small changes in these values, such as 1/3 to 1/4, barely changed the selected keywords, resulting in a minor influence on the accuracy.

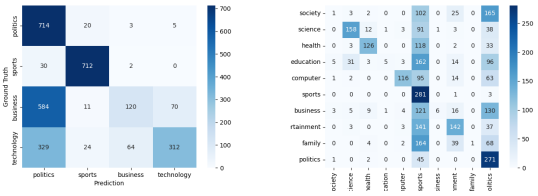

Figure 4: Error analysis for experimental results using OpenLLaMA-7b with our keywords. The left and right figures correspond to the AG News and Yahoo dataset results, respectively. OpenLLaMA-7b prefers to predict specific classes incorrectly due to the bias of giving higher scores to keywords of those incorrect classes.

3.1.3 Results

Overall Performances. Table 1 shows the experimental results of zero-shot text classification. In comparison to the baseline, our proposed method demonstrates an average accuracy improvement of 8.5 points (8.9 points compared to w/ NPPrompt) in OPT-6.7b, 1.7 points against the baseline (11.9 points compared to w/ NPPrompt) in OpenLLaMA-7b.

For some task-model combinations (Yahoo, AG News and OpenLLaMA-7b), the proposed method underperforms the vanilla OpenLLaMA-7b. To understand the reason for this, we show the confusion matrix in Figure 4. The figure shows that when the proposed method performs poorly, OpenLLaMA-7b prefers to predict specific classes incorrectly. For the case of the AG News dataset, OpenLLaMA-7b with our keywords prefers the "politics" class. We believe this is due to the bias of OpenLLaMA-7b to give higher scores to keywords in the "politics" class. In practice, we observed that OpenLLaMA-7b gave a high score to the keywords of the "politics" class even though the keywords seem to have no relationship with the input text. Although our proposed method subtracts the null prompt score to reduce the biases as in Zhao et al. (2021); Holtzman et al. (2021), there is still room for improvement regarding the score calibration method to alleviate the problem.

Case Studies. To further understand the disparity between NPPrompt and our method, we analyze the selected keywords and predictions on the Yahoo dataset. Table 2 shows that our diverse keywords can encourage a classifier to make a prediction based on the relatedness between a text and various semantics of the class. For example, the proposed method gives a high score to the keyword "chemistry" in the "science" class for the input text "Please answer this chem problem for me?". Thus,

Table 3: Relationship between the properties of keywords and accuracy. While considering only intra-class diversity or inter-class exclusivity underperform the vanilla model, considering both outperform in most cases.

Method	AG News	DBpedia	Yahoo	SST-2	RT	Avg.
OPT-6.7b	75.8	50.7	33.7	55.1	58.8	54.8
w/ Sim	**84.2**	**64.8**	47.6	56.0	58.3	62.1
w/ Sim + Exc	75.7	53.0	49.0	**68.6**	64.3	62.1
w/ Sim + Div	74.3	52.4	48.1	55.1	52.3	56.4
w/ Sim + Exc + Div	79.7	49.4	**49.5**	68.5	**69.3**	**63.3**

the proposed method correctly classifies the input text into the "science" class, while NPPrompt, which does not have the keyword "chemistry", fails to correctly classify the input text.

3.1.4 Analysis

To confirm the effectiveness of intra-class diversity and inter-class exclusivity in keyword reranking, we conduct experiments with varying keyword reranking methods. We compare four variants of CE-MMR with OPT-6.7b and vanilla OPT-6.7b as a baseline. For CE-MMR, we turn on and off three terms in Equation 3, where we denote the first, second, and third terms by **Sim**, **Div**, and **Exc**.

Table 3 shows the results. On average, Sim + Exc + Div, which considers intra-class diversity, inter-class exclusivity, and similarity to class names, achieves the highest accuracy. In sentiment analysis datasets, we find that inter-class exclusivity of keywords is more critical than intra-class diversity by comparing Sim+Exc to Sim+Div. This result suggests that when class names are antonyms such as "great" and "terrible", models are prone to give confusing keywords unless inter-class exclusivity is taken into account. Sim achieves the best results in the topic classification AG News and DBpedia. This result indicates that similarity is more important for some datasets and assigning reranking weights to exclusivity is sometimes semi-optimal. In practical applications, we can select the values of α, β, and γ according to the accuracy of the validation data. In addition, Sim + Div showed lower performance for all data in the zero-shot setting, while Sim + Exc + Div showed the best on average. This result suggests that it is not sufficient to consider only intra-class diversity, but it is essential to simultaneously consider inter-class exclusivity in order to achieve high accuracy.

3.2 Generalized Zero-shot Text Classification

We conduct experiments to confirm that our proposed method is also effective for the generalized zero-shot classification setting.

3.2.1 Datasets

We use four publicly available multi-class text classification datasets, including topic classification, intent classification, and emotion classification. The topic classification datasets are **Amazon** (McAuley et al., 2015), a collection of reviews for products and their categories, and **WoS** (Kowsari et al., 2017), a collection of academic papers and their research areas. The intent classification dataset is **Snips** (Coucke et al., 2018) that contains crowdsourced queries and their intent, such as "Book Restaurant". The emotion dataset is **Emotion** (Bostan and Klinger, 2018), a widely used benchmark for zero-shot text classification (Yin et al., 2019; Ye et al., 2020), a collection of short sequences and their emotion labels such as "joy" and "sad".

Preprocessing. We randomly select 50% from all classes as seen classes, the other 25% as unseen classes, and the other 25% as validation classes. Then, training data is selected from seen classes, validation data from seen and validation classes, and test data from the seen and unseen classes.

3.2.2 Compared Methods

We evaluate our methods, several baselines for GZSTC, and a method for a fully supervised setting as a reference. **LabelSim** (Sappadla et al., 2016) uses word embeddings to calculate similarities between an instance and class names. **LTA** (Zhang et al., 2022b) is a meta-learning method that rehearse on fake unseen classes selected from seen classes. **Entailment** (Yin et al., 2019) treats text classification tasks as textual entailment that predict whether a given text entails "This text is about {class name}." using a pre-trained language

Table 4: Harmonic mean accuracies of seen and unseen classes on generalized zero-shot text classification (seen and unseen class accuracies in the brackets). Bold values indicate the best results among GZSTC methods. Notice that LTA splits seen classes into fake seen and fake unseen classes, which is not applicable for datasets with a small number of seen classes, such as WoS and Snips. † Averaged on only Amazon and Emotion datasets.

Method	Amazon	WoS	Snips	Emotion	Avg
LabelSim	7.95$_{(7.83,\ 8.08)}$	40.5$_{(29.4,\ 65.3)}$	70.6$_{(75.7,\ 66.1)}$	6.46$_{(10.0,\ 22.3)}$	32.2$_{(33.8,\ 36.4)}$
LTA	53.5$_{(69.5,43.5)}$	N/A	N/A	42.7$_{(37.9,\mathbf{48.9})}$	†48.1$_{(53.7,46.2)}$
w/ Ours	66.6$_{(58.2,\mathbf{77.7})}$	N/A	N/A	35.6$_{(30.3,43.3)}$	†51.1$_{(44.2,60.5)}$
Entailment	63.2$_{(89.1,49.0)}$	83.1$_{(\mathbf{92.8},75.3)}$	98.9$_{(\mathbf{99.8},98.1)}$	46.5$_{(\mathbf{72.0},34.3)}$	72.9$_{(\mathbf{88.4},64.1)}$
w/ Ours	**77.3**$_{(\mathbf{92.0},66.7)}$	**86.3**$_{(92.0,\mathbf{81.3})}$	**99.2**$_{(99.4,\mathbf{98.9})}$	**56.4**$_{(69.0,47.6)}$	**79.8**$_{(88.1,\mathbf{73.6})}$
Fully Supervised					
BERT	92.4$_{(92.7,92.0)}$	92.8$_{(89.2,96.7)}$	99.7$_{(99.9,99.6)}$	61.6$_{(68.4,54.0)}$	86.6$_{(87.5,85.5)}$

Table 5: Effectiveness of considering intra-class diversity and inter-class exclusivity on harmonic mean accuracies with seen and unseen class accuracies in brackets.

Method	Amazon	WoS	Snips	Emotion	Avg
No Reranking					
Term-Frequency	71.4$_{(91.6,\ 58.5)}$	79.8$_{(92.8,\ 69.9)}$	98.9$_{(100,\ 97.8)}$	54.4$_{(67.6,\ 45.5)}$	76.1$_{(88.0,\ 67.9)}$
Reranking					
w/ Sim	**77.3**$_{(92.3,\ 66.5)}$	75.9$_{(92.5,64.4)}$	97.7$_{(\mathbf{100},\ 95.5)}$	31.4$_{(68.1,\ 20.4)}$	70.6$_{(88.2,\ 61.7)}$
w/ Sim + Exc	74.0$_{(\mathbf{92.7},\ 61.5)}$	76.1$_{(\mathbf{93.2},\ 64.3)}$	97.4$_{(\mathbf{100},\ 94.9)}$	25.3$_{(\mathbf{74.6},\ 15.2)}$	68.2$_{(\mathbf{90.1},\ 59.0)}$
w/ Sim + Div	68.7$_{(92.2,\ 54.8)}$	83.8$_{(92.4,\ 76.7)}$	92.1$_{(87.4,\ 97.3)}$	**60.3**$_{(69.7,\ \mathbf{53.1})}$	76.2$_{(85.4,\ 70.5)}$
w/ Sim + Exc + Div	**77.3**$_{(92.0,\ \mathbf{66.7})}$	**86.3**$_{(92.0,\ \mathbf{81.3})}$	**99.2**$_{(99.4,\ \mathbf{98.9})}$	56.4$_{(69.0,\ 47.6)}$	**79.8**$_{(88.1,\ \mathbf{73.6})}$

model. In addition to these baselines, we denote our method combined with baselines as **w/ Ours**. We combine Entailment, LTA and the proposed method by simply replacing a class name with the keyword expanded class name "{class name} such as {keyword1}, {keyword2}, {keyword3} , {keyword4}" because we found this simple method to be sufficient for improving performance, as it requires the same order of computation as the vanilla method.

To find out how much room for improvement is left compared to the *fully supervised setting*, we compare **BERT** trained on the training data for seen classes and training data for unseen classes that is not available for GZSTC methods.

3.2.3 Experimental Setup

Evaluation Metrics. We use accuracies of seen and unseen classes and their harmonic mean as evaluation metrics as in Zhang et al. (2022b). We use the harmonic mean to measure overall performances since there is a trade-off between seen and unseen class accuracy.

Implementation Details. We use `bert-base-uncased` (Devlin et al., 2019) as a pre-trained language model for Entailment, Entailment w/ ours, LTA, and Supervised BERT. For Entailment, we do not conduct pre-finetuning on an NLI dataset suggested in the original paper (Yin et al., 2019) since the original BERT without pre-finetuning shows better performances in our experiments. Our method uses GPT-J-6B (Wang and Komatsuzaki, 2021) as a generative language model. Furthermore, for reranking keywords, we use the cosine similarity of embeddings obtained by the BERT encoder as similarities in Equation 3 and select the top-4 keywords per class. For LabelSim, we use the bi-gram of public fastText (Grave et al., 2018) embeddings trained on the Wikipedia corpus. Please refer to Appendix A for other implementation details.

Hyperparameters. To validate the model, we use validation data that consists of labeled data of seen classes and validation classes. Search spaces and determined values of hyperparameters are described in Appendix A.2.

3.2.4 Results

Table 4 shows the results of the end-to-end experiments. In comparison, Entailment overperformes

Table 6: Effectiveness of task-aware keyword generation on harmonic mean accuracies of seen and unseen classes (seen and unseen class accuracies in the brackets). Bold values indicate the best results among methods.

Generation Method	Amazon	WoS	Snips	Emotion	Avg
Language Model	**78.5**$_{(\mathbf{90.5}, \mathbf{69.3})}$	84.5$_{(\mathbf{92.5}, 77.8)}$	98.0$_{(\mathbf{100}, 96.1)}$	47.8$_{(\mathbf{69.7}, 36.4)}$	77.2$_{(\mathbf{88.2}, 69.9)}$
In-Context	77.3$_{(92.0, 66.7)}$	**86.3**$_{(92.0, \mathbf{81.3})}$	**99.2**$_{(99.4, \mathbf{98.9})}$	**56.4**$_{(69.0, \mathbf{47.6})}$	**79.8**$_{(88.1, \mathbf{73.6})}$

the other baselines, and Entailment and LTA with our extension overperformes methods without using our extension on average. The results suggest that the Entailment method generalizes better than the dual-encoder approach (LTA), as pointed out in the few-shot settings in Müller et al. (2022). Also, the results suggest that keywords selected with our method help improve unseen class accuracy due to the keywords complementing the lack of information on unseen classes. Compared to the result of the fully supervised method, there is a little room for improvement.

3.2.5 Analysis

We analyze the contribution of each component of our method by conducting additional experiments.

Keyword Reranking Methods. To confirm the effectiveness of reranking keywords by the intra-class diversity and inter-class exclusivity in the generalized zero-shot settings, we conduct ablation studies on reranking methods similar to Section 3.1.4. We use the Entailment method without reranked keywords as a baseline and compare four reranking methods to the baseline.

Table 5 shows the comparison results of keyword reranking methods. Consistent with the analysis in Section 3.1, the method considering all the characteristics is the best among compared methods on average. An inconsistent trend with the fully zero-shot setting is that intra-class diversity is more important than inter-class exclusivity in the generalized zero-shot setting. We hypothesize that the classifier learns to ignore noisy keywords and concentrate only on relevant ones through model training.

Keyword Candidate Generation Methods. To study the effectiveness of task-aware keywords described in Section 2.3 compared to task-unaware keywords, we compare keyword candidate generation technique that uses in-context demonstrations of class name and keyword pairs of seen classes (**In-Context**) to generate task-aware keywords and keyword candidates generation technique that uses only class names to generate task-unawware keyword candidates (**Language Model**). Implementation details are described in Appendix A. In the experiment, we use our keyword reranking method described in Section 2.4 to rerank keyword candidates and Entailment as the base classifier. Table 6 shows the experimental results to confirm the effectiveness of task-aware keywords. In-Context outperforms Language Model by 2.6 points on the harmonic mean of accuracies on average. This result indicates that task-aware keywords generated with in-context learning are more effective than task-unaware keywords generated with only class names.

4 Related Work

Zero-shot Text Classification. Zero-shot text classification is a text classification task in a special situation where some target classes do not have any training data. Existing methods for zero-shot text classification decide the class y of an input instance x based on the relationship between a class name and an instance (Sappadla et al., 2016; Yin et al., 2019) such as semantic similarity. Recent methods use a pre-trained language model (PLM) to calculate the similarity (Holtzman et al., 2021; Xia et al., 2022; Sun et al., 2022) of the class and the instance. For example, Schick and Schütze (2021) transforms similarity calculations into the predictions of masked token probabilities such as "Good movie! [SEP] The sentiment of this review is [MASK].". If the likelihood of "great" is higher than "bad" for "[MASK]", one can classify "Good movie!" into the positive class. At this time, it is necessary to associate the vocabulary of the PLMs and the target classes.

When training data for a part of target classes are available, the task is *generalized zero-shot text classification* (GZSTC). Similar to zero-shot text classification, Pushp and Srivastava (2017) predicts the relatedness between texts and classes by a trained neural network with the training data, and Yin et al. (2019) proposes a textual entailment-based method with PLMs, where textual entailment-based meth-

ods show effectiveness in other zero-shot tasks such as stance detection (Xu et al., 2022) and ultra-fine entity typing (Li et al., 2022). LTA (Zhang et al., 2022b) applies meta-learning for GZSTC, which learns how to adapt the encoder to new classes by episodic training on fake unseen classes selected from seen classes.

In another line of work, when a large amount of unlabeled data for target classes is available, the task is called *weakly supervised text classification* and has been studied in (Meng et al., 2018; Mekala and Shang, 2020; Mekala et al., 2022; Zhang et al., 2021; Wang et al., 2021). X-Class (Wang et al., 2021) uses class-adaptive embedding representations of instances to obtain high-quality pseudo-labeled data. Zhang et al. (2023) proposes PIEClass that iteratively trains two types of classifiers, a prompt-based classifier, and a head-token classifier, to correct pseudo-label errors with each other. Since the existing weakly supervised text classification methods require a large amount of in-domain unlabeled data that are unavailable for unseen classes in zero-shot scenarios, those methods are not applicable in our problem settings.

Class-keyword Mapping Construction. What keywords are associated with the target classes is crucial. In PET (Schick and Schütze, 2021), a mapping from keywords to classes is designed by users. For instance, in sentiment analysis, the word "terrible" is associated with the negative class, and "great" is associated with the positive class. However, since manually constructing class-keyword mappings is costly, methods to automate the process have been proposed (Schick et al., 2020; Shin et al., 2020; Shi et al., 2022; Hu et al., 2022; Zhao et al., 2023). If training data is available, they can be utilized in the construction method (Schick et al., 2020; Shin et al., 2020). When a large amount of unlabeled corpus is available, weakly supervised methods (Meng et al., 2020c,a,b) are practical to acquire keywords. LOTClass (Meng et al., 2020b) masks class names in unlabeled data and obtains mask tokens predicted by the mask language model as keywords associated with the class names. As in our problem setting, when both labeled and unlabeled data are unavailable, one approach is to select words that resemble the class name based on embedding similarity (Zhao et al., 2023).

While the conventional methods select keywords for each class independently, the attached keywords ignore the nature of classification as described in Figure 1. To avoid choosing such keywords, our proposed method selects keywords carefully by considering intra-class diversity and inter-class exclusivity of keywords.

5 Conclusion

This paper proposes a novel method for improving zero-shot text classification that finds keywords related to classes properly. Our method generates diverse keyword candidates by a generative language model and reranks the candidates by an extended maximal marginal relevance method to acquire the keywords that are diverse within a class and exclusive among different classes. Experimental results on fully zero-shot and generalized zero-shot text classification tasks demonstrate the effectiveness of the proposed method.

6 Limitations

We used a limited variety of language models in the experiments, but further work will be needed to confirm that our results are maintained for other models, such as multi-lingual models or larger-sized models. Even if we use our proposed method, it is still necessary to provide appropriate seed class names manually. Also, our proposed method is applicable to few-shot learning, so we need to investigate whether the proposed method is effective in these settings.

7 Acknowledgements

We thank three anonymous reviewers for their helpful comments and suggestions.

References

Laura Ana Maria Bostan and Roman Klinger. 2018. An analysis of annotated corpora for emotion classification in text. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2104–2119. Association for Computational Linguistics.

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*.

Jaime G. Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, pages 335–336. ACM.

Together Computer. 2023. Redpajama-data: An open source recipe to reproduce llama training dataset.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinyang Geng and Hao Liu. 2023. Openllama: An open reproduction of llama.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomás Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based TF-IDF procedure. *CoRR*, abs/2203.05794.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2240, Dublin, Ireland. Association for Computational Linguistics.

Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. 2017. Hdltex: Hierarchical deep learning for text classification. In *16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, Cancun, Mexico, December 18-21, 2017*, pages 364–371. IEEE.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Bangzheng Li, Wenpeng Yin, and Muhao Chen. 2022. Ultra-fine entity typing with indirect supervision from natural language inference. *Transactions of the Association for Computational Linguistics*, 10:607–622.

Xinxi Lyu, Sewon Min, Iz Beltagy, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. Z-ICL: zero-shot in-context learning with pseudo-demonstrations. *CoRR*, abs/2212.09865.

Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 43–52. ACM.

Dheeraj Mekala, Chengyu Dong, and Jingbo Shang. 2022. LOPS: learning order inspired pseudo-label selection for weakly supervised text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4894–4908. Association for Computational Linguistics.

Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 323–333. Association for Computational Linguistics.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Zihan Wang, Chao Zhang, Yu Zhang, and Jiawei Han. 2020a. Discriminative topic mining via category-name guided text embedding. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2121–2132. ACM / IW3C2.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 983–992. ACM.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020b. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9006–9017. Association for Computational Linguistics.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Yu Zhang, Chao Zhang, and Jiawei Han. 2020c. Hierarchical topic mining via joint spherical tree and text embedding. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1908–1917. ACM.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330, Dublin, Ireland. Association for Computational Linguistics.

Sewon Min, Weijia Shi, Mike Lewis, Xilun Chen, Wentau Yih, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2023. Nonparametric masked language modeling. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2097–2118, Toronto, Canada. Association for Computational Linguistics.

Thomas Müller, Guillermo Pérez-Torró, and Marc Franco-Salvador. 2022. Few-shot learning with Siamese networks and label tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8532–8545, Dublin, Ireland. Association for Computational Linguistics.

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.

Pushpankar Kumar Pushp and Muktabh Mayank Srivastava. 2017. Train once, test anywhere: Zero-shot learning for text classification. *CoRR*, abs/1712.05972.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. 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.

Prateek Veeranna Sappadla, Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016. Using semantic similarity for multi-label zero-shot classification of text documents. In *24th European Symposium on Artificial Neural Networks, ESANN 2016, Bruges, Belgium, April 27-29, 2016*.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5569–5578, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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*, pages 255–269, Online. Association for Computational Linguistics.

Weijia Shi, Julian Michael, Suchin Gururangan, and Luke Zettlemoyer. 2022. Nearest neighbor zero-shot inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3254–3265, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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.

Yi Sun, Yu Zheng, Chao Hao, and Hangping Qiu. 2022. NSP-BERT: A prompt-based few-shot learner

through an original pre-training task —— next sentence prediction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3233–3250, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Zihan Wang, Dheeraj Mekala, and Jingbo Shang. 2021. X-class: Text classification with extremely weak supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3043–3053. Association for Computational Linguistics.

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, Virtual Event, April 25-29, 2022*. OpenReview.net.

Mengzhou Xia, Mikel Artetxe, Jingfei Du, Danqi Chen, and Veselin Stoyanov. 2022. Prompting ELECTRA: Few-shot learning with discriminative pre-trained models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11351–11361, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hanzi Xu, Slobodan Vucetic, and Wenpeng Yin. 2022. Openstance: Real-world zero-shot stance detection. In *Proceedings of the 26th Conference on Computational Natural Language Learning, CoNLL 2022, Abu Dhabi, United Arab Emirates (Hybrid Event), December 7-8, 2022*, pages 314–324. Association for Computational Linguistics.

Zhiquan Ye, Yuxia Geng, Jiaoyan Chen, Jingmin Chen, Xiaoxiao Xu, SuHang Zheng, Feng Wang, Jun Zhang, and Huajun Chen. 2020. Zero-shot text classification via reinforced self-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3014–3024, Online. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Lu Zhang, Jiandong Ding, Yi Xu, Yingyao Liu, and Shuigeng Zhou. 2021. Weakly-supervised text classification based on keyword graph. 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 2803–2813. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022a. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

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.

Yiwen Zhang, Caixia Yuan, Xiaojie Wang, Ziwei Bai, and Yongbin Liu. 2022b. Learn to adapt for generalized zero-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 517–527, Dublin, Ireland. Association for Computational Linguistics.

Yunyi Zhang, Minhao Jiang, Yu Meng, Yu Zhang, and Jiawei Han. 2023. PIEClass: Weakly-supervised text classification with prompting and noise-robust iterative ensemble training. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12655–12670, Singapore. Association for Computational Linguistics.

Xuandong Zhao, Siqi Ouyang, Zhiguo Yu, Ming Wu, and Lei Li. 2023. Pre-trained language models can be fully zero-shot learners. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15590–15606, Toronto, Canada. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

A Other Experimental Details.

A.1 Datasets

Table 7 and Table 8 are statistics of datasets used in experiments of Section 3.1 and Section 3.2, respectively.

Table 9 shows class names and templates used in our experiments.

Table 7: Statistics of datasets used in the zero-shot experiments.

	#Instances	#Classes	Domain
AG News[2]	3000	4	News
DBpedia[3]	3000	14	Wikipedia
Yahoo[4]	3000	10	Yahoo Answers
SST-2[5]	872	2	Movie review
RT[6]	1066	2	Movie review

Table 8: Statistics of datasets used in the experiments before splitting into seen and unseen classes.

	#Instances	#Classes	Domain
Amazon[7]	24,000	24	Product Review
WoS[8]	46,985	7	Academic Paper
Snips	13,802	7	Voice Assistant
Emotion	36,463	10	Mixed

A.2 Hyperparameters

Table 10 shows the hyperparameters used for model training in Section 3.2. We use the same values of hyperparameters in the original papers, except for parameters that the original papers use different values for different datasets. We use the same hyperparameters as the vanilla LTA or Entailment for our methods combined with LTA or Entailment.

In addition to the hyperparameters described in the table, parameters that are unique for each method are set as follows.

LTA We use the hyperparameters for LTA in the original paper as $d_h = 768, d_a = 768, \alpha = 10.0, \tau = 10.0, N^{s_i} = N^{u_i} = 2, K = 5, d_r = 32$.

Entailment For the template to generate a hypothesis, we use "This text is about {class name}." as suggested in the original paper.

Ours When generating sequences that contain keyword candidates in our method, the temperature parameters that control the generation probabilities, top_p parameter (the threshold for top-p sampling), and generation length are manually set to 0.9, 0.8, and 16, respectively. We generate 20 sequences for each class and extract 24 keyword candidates with the highest Term-Frequency value per class.

A.3 Templates for Generating Keyword Candidates

Zero-Shot Text Classification. We use the following templates to generate keyword candidates for experiments in Section 3.1.

- "{class name} such as "
- "{class name}: "
- "examples of {class name} are "
- "{class name} also "
- "{class name} and "

Generalized Zero-Shot Text Classification. We use the following templates to generate keyword candidates for experiments in Section 3.2.

- "{class name} such as {keyword candidate1}, {keyword candidate2}, $\cdots$",
- "{class name}: {keyword candidate1}, {keyword candidate2}, $\cdots$",
- "examples of {class name} are {keyword candidate1}, {keyword candidate2}, $\cdots$",

where a {keyword candidate} is a keyword of the seen class extracted from training data. We concatenate the class-keyword pairs of several seen classes with a line break "\n" in between and add instructions of the same format to generate the unseen class keywords on the last line. When retrieving keyword candidates from training data for seen classes, we aggregate training data for each class and use TF-IDF to retrieve class-specific keywords, which is similar to class-based TF-IDF (Grootendorst, 2022).

Table 9: Templates and class names used in our experiments.

Dataset	Class Name	Template
AG News	"politics", "sports", "business", "technology"	"{text}topic: "
DBpedia	"Company", "school", "Artist", "Athlete", "OfficeHolder", "transportation", "Building", "Mountain", "Village", "Animal", "Plant", "Album", "Film", "book"	"{title}{content}{title} is a "
Yahoo	"society", "science", "health", "education", "computer", "sports", "business", "entertainment", "amily", "politics"	"{question title}topic: "
SST-2	"terrible", "great"	"{text}It was "
RT	"terrible", "great"	"{text}It was "
Amazon	(seen) "Apps for Android", "Baby", "Beauty", "Clothing Shoes and Jewelry", "Digital Music", "Electronics", "Movies and TV", "Patio Lawn and Garden", "Pet Supplies", "Tools and Home Improvement", "Toys and Games", "Video Games" (unseen) "Amazon Instant Video", "CDs and Vinyl", "Cell Phones and Accessories", "Grocery and Gourmet Food", "Kindle Store", "Office Productsy" (valid) "Automotive", "Books", "Health and Personal Care", "Home and Kitchen", "Musical Instruments", "Sports and Outdoors"	"{text}This text is about {class name}"
WoS	(seen) "Civil Engineering", "Computer Science", "Mechanical Engineering" (unseen) "Electrical Engineering", "Medical Science" (valid) "Psychology", "biochemistry"	"{text}This text is about {class name}"
Snips	(seen) "book", "movie", "playlist", (unseen) "music", "restaurant" (valid) "search", "weather"	"{text}This text is about {class name}"
Emotion	(seen) "anger", "fear", "love", "no emotion" (unseen) "disgust", "sadness", "shame" (valid) "guilt", "joy", "surprise"	"{text}This text is about {class name}"

Table 10: Hyperparameters for fine-tuning. Notice that the batch size of LTA (step2) is determined by K, N^{s_i}, and N^{u_i}.

Hyperparameter	LTA (step1)	LTA (step2)	Entailment
# of maximum epochs	10	300	3
Model selection	early stopping (3epochs)	early stopping (30epochs)	best epoch
Learning rate	1e-3	1e-5	1e-5
Scheduler	None	None	linear
Optimizer	Adam	Adam	AdamW
Adam epsilon	1e-08	1e-08	1e-08
Adam beta weights	0.9, 0.999	0.9, 0.999	0.9, 0.999
Weight decay	0.0	0.0	0.01
Batch size	64	N/A	32