SELP: A Semantically-Driven Approach for Separated and Accurate Class Prototypes in Few-Shot Text Classification

Wenxin Liang[1], **Tingyu Zhang**[1], **Han Liu**[1*], **Feng Zhang**[2]
[1]Dalian University of Technology, Dalian, China
[2]Peking University, Beijing, China
wxliang@dlut.edu.cn, zhangtingyv@mail.dlut.edu.cn,
liu.han.dut@gmail.com, zfeng.maria@gmail.com

Abstract

The meta-learning paradigm has demonstrated significant effectiveness in few-shot text classification. Currently, numerous efforts are grounded in metric-based learning, utilizing textual feature vectors for classification, with a common emphasis on enlarging inter-class distances to achieve improved classification effectiveness. However, many methods predominantly focus on enhancing the separation of prototypes without taking the semantic relationships between prototypes and class clusters into consideration. This oversight results in incomplete and inaccurate encoding of prototypes within the semantic space, affecting the generality of the learned metric space. In this paper, we propose the utilization of **S**emantically **E**nhanced **L**abels for calibrating class **P**rototypes (**SELP**), thereby obtaining prototypes that are more separated and semantically accurate. Additionally, we have devised a center loss to enhance intra-class compactness, coupled with the introduction of a simulated label distribution method to address the overfitting problem. Extensive experiments on eight few-shot text classification datasets show that the proposed method outperforms baselines significantly. Our code is available at https://github.com/tttyyyzzz-zty/SELP.git.

1 Introduction

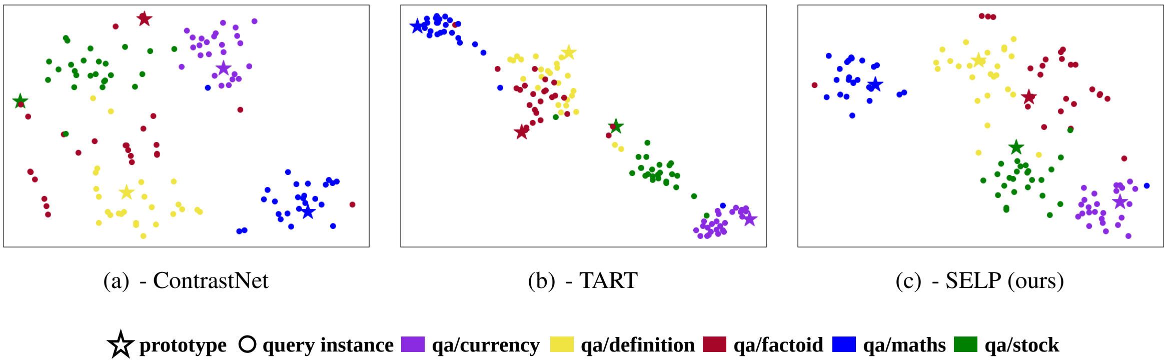
Text classification is a crucial and foundational task in natural language understanding, widely employed across various domains such as intent recognition (Dopierre et al., 2021) and sentiment analysis (Kumar and Abirami, 2021). The application of deep learning approaches to text classification tasks, combined with extensive supervised training on vast data, has demonstrated substantial efficacy in achieving superior performance in text classification endeavors (Devlin et al., 2019; Raffel et al., 2020; Song et al., 2020). However, the collection of such a substantial volume of annotated data is time-consuming and laborious, rendering it impractical across numerous real-world domains, thereby motivating few-shot text classification.

There are several methods proposed to solve few-shot text classification tasks. Fine-tuning based methods (Howard and Ruder, 2018; Shen et al., 2021) often use a pre-trained language model (PLM) and then fine-tune the model to the downstream task. A variant of this approach is prompt-based methods (Li and Liang, 2021; Schick and Schütze, 2021), which typically transform text classification tasks into closed-text formats to bridge the gap between pre-train and downstream tasks to better utilize the capabilities of the pre-trained model. However, transforming the task into a fill-in-the-blank format entails inherent limitations, demanding meticulous and precise task design, which might lack applicability across various real-world tasks. Data augmentation based methods (Liu et al., 2019b; Dopierre et al., 2021) aim to utilize auxiliary data or information for data augmentation or feature augmentation of few-shot datasets, but they may introduce new noisy data, which are more dependent on prior knowledge, and may lead to feature loss. Meta-learning based methods (Lei et al., 2023; Chen et al., 2022) aim to give the model the ability to quickly generalize to novel classes by learning on different simulated small episodes (tasks). Meta-learning based methods have shown promising results in few-shot tasks and are considered a highly promising methodology.

Although all of the above methods achieve good performance, there are still some problems with the current methods. For methods that utilize textual feature vectors for classification, their performance is highly dependent on the inter- and intra-class variance in the query set. The key to solving this problem is to encourage greater intra-class compactness and inter-class separability explicitly, and there are already some works to do so. ContrastNet

[*]Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2024, pages 9732–9741
August 11-16, 2024 ©2024 Association for Computational Linguistics


(a) - ContrastNet

(b) - TART

(c) - SELP (ours)

Figure 1: t-SNE visualization of text representation for a testing episode (N = 5, K = 1, Q =25) sampled from HWU64. Note that the 5 classes are not seen in the training set. The text representation is given by (a) ContrastNet (b) TART and (c) SELP (ours).

(Chen et al., 2022) and TART (Lei et al., 2023) employ contrastive learning methods to promote mutual separation among prototypes. However, these methods primarily emphasize prototypes that are more separated, neglecting the advanced semantic relationships between prototypes and class clusters. In some cases, the computed prototypes may not effectively represent their corresponding classes. This leads to inaccuracies in encoding prototypes within the semantic space, consequently affecting the generality of the learned metric space and diminishing model generalization. We propose to address the aforementioned issues by employing labels to calibrate prototypes, aiming to obtain separable and more accurate prototypes. While some existing work utilizes labels for classification (Luo et al., 2021; Du et al., 2023), they do not manipulate the prototypes. Moreover, some methods resort to leveraging external knowledge bases to augment label semantics due to the limited information contained in labels (Zhang et al., 2022). We contend that this approach is intricate, and the obtained labels lack genuine instance-based information, resulting in limited generalization capability.

In this study, we introduce a simple yet effective approach that utilizes a prompt pool to enrich the semantic content of labels. This method ensures that labels encapsulate richer information about the underlying semantics of instances within their associated classes. Subsequently, we utilize these enriched labels to calibrate class prototypes, encouraging increased distinctiveness among prototypes and bolstering their semantic accuracy. Additionally, we introduce a center loss to enhance intra-class compactness and a simulated label distribution method to mitigate the overfitting problem. Figure 1 illustrates the comparison between our approach and two strong baseline methods. It can be observed that our method effectively pulls back the prototypes to the locations where class clusters aggregate, achieving better intra-class compactness.

2 Related Work

2.1 Fine-Tuning Based Methods

Fine-tuning based methods typically use pre-trained language models (PLMs) and then fine-tune them on specific downstream tasks to obtain models adapted to the downstream task. Some methods (Howard and Ruder, 2018; Shen et al., 2021) aim to preserve the transferability of the model by applying distinct learning rates to each layer during training. However, fine-tuning based methods suffer from the overfitting problem due to the scarcity of training data. Recently, prompt-based approaches have achieved promising results. PET (Schick and Schütze, 2021) models classification problems as fill-in-blank problems, bridging the gap between upstream and downstream tasks. LM-BFF (Gao et al., 2021) uses the T5 (Raffel et al., 2020) for automated template generation, and in turn combines examples with the current text, enriching the contextual information. However, transforming the task into a fill-in-the-blank format entails inherent limitations, demanding meticulous and precise task design, which might lack applicability across various real-world tasks.

2.2 Data Augmentation Based Methods

The core idea of data augmentation based approaches is to utilize auxiliary data or information for data augmentation or feature augmentation of few-shot datasets. Dopierre et al. (2021) propose a short text paraphrase model, which produces different paraphrases of the original text as data augmentation. TPN (Liu et al., 2019b) employs a transductive approach by constructing an undirected graph that integrates all unlabeled and labeled data, obtaining labels for all unlabeled data through label propagation. Way-DE (Liu et al., 2023) assumes a Gaussian distribution for each class and utilizes the original support set along with the nearest minority query samples to estimate the mean and covariance. Subsequently, it augments labeled samples by sampling from the estimated distribution. While data augmentation remains a reliable method for addressing few-shot problems, it may inadvertently introduce new noisy data, become more reliant on prior knowledge, and potentially lead to feature loss.

2.3 Meta-Learning Based Methods

Meta-learning aims to enhance the multi-tasking generalization of the model by sampling episodes on the seen classes so that it can be quickly adapted to novel classes. Existing meta-learning based methods can be divided into three types. (1) Optimization-based methods, such as MAML (Finn et al., 2017) and MAML++ (Antoniou et al., 2019), aim to learn a good model initialization parameter that can be quickly adapted to a new task within a few steps of gradient updating. (2) Model-based methods, such as MANN (Santoro et al., 2016) and Meta-ticket (Chijiwa et al., 2022), develop specific model architectures that enable rapid adaptation to novel tasks. (3) Metric-based methods like prototypical network (Snell et al., 2017) and induction network (Geng et al., 2019), utilize a metric function to calculate the distance or similarity between different samples, which in turn determines the category of the sample to be predicted. Current metric-based methods focus on generating separable feature representations, ContrastNet (Chen et al., 2022) through supervised contrastive learning and two unsupervised contrastive learning approaches at the task level and instance level, achieves more discriminative prototype representations and alleviates the issue of overfitting. TART (Lei et al., 2023) transforms the class prototypes to per-class fixed reference points in task-adaptive metric spaces and uses a discriminative reference regularization to further maximize divergence between transformed prototypes. Meta-learning stands as a promising approach for addressing few-shot tasks.

3 Model

3.1 Problem Formulation

In this paper, we follow the traditional N-way K-shot setting. Specifically, let $\mathcal{C}_{\text{train}}$, $\mathcal{C}_{\text{val}}$, $\mathcal{C}_{\text{test}}$ denote the disjoint set of training classes, validation classes and test classes, and they have no overlapping classes, i.e., $\mathcal{C}_{\text{train}} \bigcap \mathcal{C}_{\text{val}} \bigcap \mathcal{C}_{\text{test}} = \emptyset$.

In the training phase, we construct training episodes $\mathcal{N}_{\text{train}}$ from $\mathcal{C}_{\text{train}}$, and each task contains a support set $\mathcal{S}$ and a query set $\mathcal{Q}$. We randomly select N classes of K samples each to form the support set, i.e., $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N\times K}$, where x_i is a data sample, y_i is the class label. The query set consists of a portion of the remaining samples from these N classes, i.e., $\mathcal{Q} = \{(x_j, y_j)\}_{j=1}^{N\times q}$, where q is the number of queries.

In the validation and testing phases, we use the same approach to construct $\mathcal{N}_{\text{val}}$ and $\mathcal{N}_{\text{test}}$ from $\mathcal{C}_{\text{val}}$ and $\mathcal{C}_{\text{test}}$, respectively. But in the testing phase, as the labels of queries are unknown in the testing stage, the query set in a test task can be represented as $\mathcal{Q} = \{x_j\}_{j=1}^{N\times q}$. A meta-learner is trained on such episodes that attempt to classify the texts in the query set $\mathcal{Q}$ on the basis of few labeled texts in the support set $\mathcal{S}$.

3.2 Overview

Our approach employs BERT (Devlin et al., 2019) as a text encoder and utilizes a prompt pool to enhance the semantic content of labels. Subsequently, the enhanced labels are used to calibrate text prototypes, aiming for more distinctive and semantically accurate text representations. Additionally, we design a center loss to enhance intra-class compactness and introduce a label distribution estimation method to mitigate overfitting issues. The overall model structure is shown in Figure 2. All notations in Figure 2 will be defined in the rest of this section.

3.3 Class-Dependent Prompt

Prompt Pool In contrast to using only a single initialization for the prompt, Wang et al. (2022) propose to learn a prompt pool. Prompt pool can extract more knowledge about the task than a single prompt and is more suitable for hard and complex tasks. The prompt pool comprises T trainable prompts, which are dynamically updated across tasks to acquire transferable meta-knowledge. Specifically, we define the prompt pool

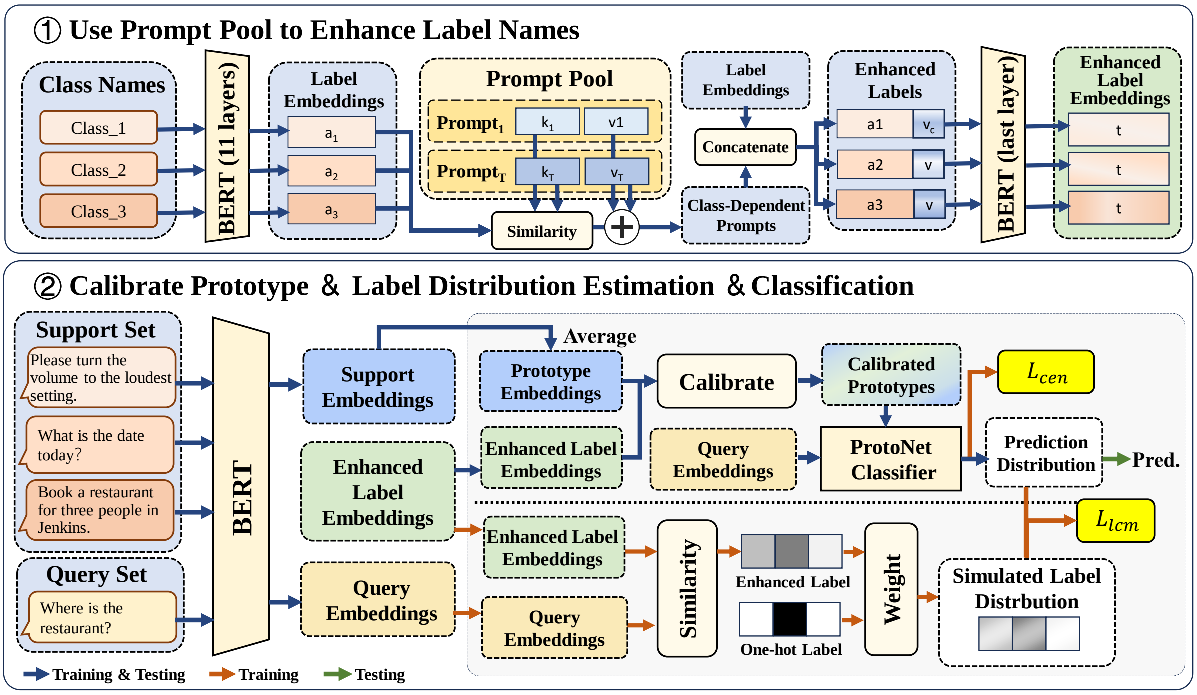


Figure 2: Illustration of the pipeline of SELP for a 3-way 1-shot task with one query example. First, a Prompt is computed for each label name using Prompt Pool, which is then connected with the corresponding label name. After obtaining embeddings for label names, support set, and query set, the prototypes are calibrated using label names. Simultaneously, a simulated label distribution is computed for each query statement, followed by the calculation of both the center loss and label distribution loss.

as follows:

$$\boldsymbol{V} = \{\boldsymbol{V}_1, \boldsymbol{V}_2, ..., \boldsymbol{V}_T\}, \boldsymbol{V}_i \in \mathbb{R}^{L\times d_h}, \tag{1}$$

where T is the number of prompts in the pool, L is the length of each prompt and d_h is the word embedding dimension. Each prompt is attached to a learnable key $\boldsymbol{k}_j$:

$$\boldsymbol{K} = \{\boldsymbol{k}_1, \boldsymbol{k}_2, ..., \boldsymbol{k}_T\}, \boldsymbol{k}_i \in \mathbb{R}^{d_h}. \tag{2}$$

Prompt Enhanced Labels Here, we are not using the prompt pool to build prompt templates, but rather to use the prompt pool to enhance labels. Prompt pool is able to learn meta-knowledge that is transferable across tasks, in each episode, we can compute a prompt for each class name through the attention mechanism. Specifically, given a class c, the weight between class name a_c and prompt $\boldsymbol{V}_j$ is computed as:

$$w_j^c = \frac{\boldsymbol{k}_j^\top \boldsymbol{a}_c^{cls}}{\sum_{j'=1}^{T} \boldsymbol{k}_{j'}^\top \boldsymbol{a}_c^{cls}}, \tag{3}$$

where $\boldsymbol{a}_c = E(a_c) \in \mathbb{R}^{L_c\times d_h}$ is the label name features of class c, obtained from the pre-trained language model(i.e., $E(\cdot)$). And $\boldsymbol{a}_c^{cls} \in \mathbb{R}^{d_h}$ is a standalone token embedding encompassing the entire semantic label (we employ the CLS token from BERT here). Then the class-dependent prompt for class c is generated by weighted all the prompt values:

$$\boldsymbol{v}_c = \sum_{j=1}^{T} w_j^c \boldsymbol{V}_j. \tag{4}$$

Because the prompt for each class is computed based on the class name, it is task-independent, and this design also allows for quick adaptation and computation for novel classes. Following the computation of a class-specific prompt, we concatenate the class name with the generated prompt to obtain a new label name $\boldsymbol{t}_c = [\boldsymbol{v}_c; \boldsymbol{a}_c]$.

It is worth noting that, to ensure the enhanced label representations and text representations remain in the same space, we further pass their representations through an additional encoder, mapping them into a unified space. Specifically, in our implementation, we employ the first 11 layers of BERT as the first text encoder (i.e., $E(\cdot)$) and use the last layer of BERT as the second text encoder (i.e., $f_e(\cdot)$). Consequently, we obtain the label representation $\boldsymbol{t}' = f_e(\boldsymbol{t})$ and the text representation $\boldsymbol{x} = f_e(E(x))$ within the same vector space.

Calibrated Prototype Representation To address the issue of insufficient separation and accuracy of prototypes, we calibrate the prototypes using enhanced labels. Specifically, we first obtain the class prototype $\boldsymbol{p}_c$ of each class through the support set

$$\boldsymbol{p}_c = \frac{1}{K} \sum_{(x_i, y_i) \in S_{train}} \mathbb{I}(y_i = c)\boldsymbol{x}_i, \tag{5}$$

where $\boldsymbol{x}_i \in \mathbb{R}^{d_h}$ is the text features, $\mathbb{I}(\cdot)$ is the indicator function. We calibrate it with semantically enhanced labels to get the final class prototype $\boldsymbol{p}'_c$:

$$\boldsymbol{p}'_c = \alpha \times \boldsymbol{p}_c + (1 - \alpha) \times \boldsymbol{t}'_c. \tag{6}$$

Here, α is a hyperparameter ranging between 0 and 1.

3.4 Center Loss

In the previous step, we have achieved improved separation among class prototypes. Now, our goal is to ensure greater compactness within each class. We propose a simple regularized loss:

$$L_{cen} = \frac{1}{2N_q} \sum_{i=1}^{N_q} ||\boldsymbol{x}_i - \boldsymbol{p}'_{y_j}||_2, \tag{7}$$

where N_q is the number of query samples in an episode. With such a simple loss, it is possible to force each sample to move closer to the prototype of the class, making the compactness within each class increase.

3.5 Label Distribution Estimation

Due to the scarcity of training samples, few-shot text classification tasks often suffer from the overfitting problem. One contributing factor to model overfitting is the inadequacy of the current one-hot encoding of labels, which fails to fully capture the relationships between instances and labels. This inadequacy arises from the non-complete independence among labels, and instances may, in practice, be associated with multiple labels. This results in the model exhibiting overconfidence on the seen classes, consequently leading to diminished generalization performance on unseen classes.

Inspired by Guo et al. (2021), we aim to transform the current one-hot encoding, a "hard label," into a "soft label" by modeling the relationships between instances and labels. This approach is employed to facilitate the model in learning as much information as possible from the instances, thereby alleviating overfitting concerns. Specifically, in a specific episode, for each query sample, we calculate its similarity with each label:

$$y^{(c)} = \mathrm{softmax}(\boldsymbol{x}_i \boldsymbol{T}^\top \boldsymbol{W} + \boldsymbol{b}), \tag{8}$$

where $\boldsymbol{T} = [\boldsymbol{t}'_1, \boldsymbol{t}'_2, ..., \boldsymbol{t}'_N]$ is the set of labels that have been enhanced in this episode. The similarity we calculated is then weighted with the original one-hot label $y^{(t)}$, and after weighting, we get the simulated label distribution (SLD) that we ultimately wish to obtain.

$$y^{(s)} = \mathrm{softmax}(\beta y^{(t)} + y^{(c)}). \tag{9}$$

Based on the prototypes obtained from Eq.6, we calculate the distance between each sample and the prototypes as the predicted label of the sample:

$$p(y = c|x_q) = \frac{\exp\left(-d(\boldsymbol{x}_q, \boldsymbol{p}'_c)\right)}{\sum_{i=1}^{N} \exp\left(-d(\boldsymbol{x}_q, \boldsymbol{p}'_i)\right)}, \tag{10}$$

$$y^{(p)} = \mathrm{softmax}([p_1, p_2, ..., p_N]), \tag{11}$$

where $\boldsymbol{x}_q$ is a query instance. We use the Kullback–Leibler divergence as the loss function to measure the difference between $y^{(p)}$ and $y^{(s)}$.

$$\begin{aligned} \mathcal{L}_{lcm} &= \textit{KL-divergence}(y^{(s)}, y^{(p)}) \\ &= \sum_{i=1}^{N} y_i^{(s)} \log\left(\frac{y_i^{(s)}}{y_i^{(p)}}\right) \end{aligned} \quad . \tag{12}$$

3.6 Objective and Prediction

Overall Objective During the training phase, we amalgamate the computed losses $\mathcal{L}_{cen}$ in Eq.7 and $\mathcal{L}_{lcm}$ in Eq.12. Given that different losses may exhibit varying scales, and manually adjusting weights can be a challenging and expensive process, we employ uncertainty weights (Kendall et al., 2018) to automatically compute the weight for each loss. The overall objective is:

$$\mathcal{L} = \frac{1}{2\sigma_1^2}\mathcal{L}_{cen} + \frac{1}{2\sigma_2^2}\mathcal{L}_{lcm} + \log \sigma_1 \sigma_2. \tag{13}$$

Testing In the testing phase, given an N-way K-shot task, we first compute the corresponding prompt for each label, then compute prototypes for each class using the samples in the support set, then calibrate the resulting prototypes with the labels, and finally compute the distance from the samples in the query set to each prototype for classification.

Dataset	#samples	#tokens per sample (mean±std)	#tokens per class name (mean±std)	#classes (train/valid/test)
HuffPost (Bao et al., 2020)	36900	13 ± 4	1 ± 1	20 / 5 / 16
Amazon (He and McAuley, 2016)	24000	152 ± 33	3 ± 1	10 / 5 / 9
Reuters (Bao et al., 2020)	620	207 ± 148	1 ± 1	15 / 5 / 11
20News (Lang, 1995)	18828	357 ± 528	5 ± 3	8 / 5 / 7
Banking77 (Casanueva et al., 2020)	13083	14 ± 9	6 ± 3	25 / 25 / 27
HWU64 (Liu et al., 2019b)	11036	7 ± 3	4 ± 1	23 / 16 / 25
Liu57 (Liu et al., 2019b)	25478	8 ± 4	1 ± 1	18 / 18 / 18
Clinc150 (Larson et al., 2019)	22500	9 ± 3	3 ± 2	50 / 50 / 50

Table 1: Dataset statistics.

4 Experiments

4.1 Datasets

In line with previous work (Liu et al., 2023), our experiments will be conducted on eight commonly used datasets, including four intent detection datasets: Banking77, HWU64, Clinc150, and Liu57, and four news or review classification datasets: HuffPost, Amazon, Reuters, and 20News. The detailed statistics for all datasets are summarized in Table 1. We provide a brief introduction to the dataset:

HuffPost (Bao et al., 2020) consists of news headlines published on HuffPost between 2012 and 2018.

Amazon (He and McAuley, 2016) consists of 142.8 million user reviews across 24 product categories. Following (Han et al., 2021), we use a subset, selecting 1,000 reviews for each category.

Reuters (Bao et al., 2020) is collected shorter Reuters articles in 1987. Following (Bao et al., 2020), we employ a subset of 31 classes.

20News (Lang, 1995) is comprised of informal discourse from news discussion forums, covers 18828 documents under 20 topics, is a subset of 20 news groups.

Banking77 (Casanueva et al., 2020) is a fine-grained intent classification dataset specific to banking domain.

HWU64 (Liu et al., 2019a) is a fine-grained intent classification dataset.

Liu57 (Liu et al., 2019a) is a highly imbalanced intent classification dataset collected on Amazon Mechanical Turkcollected from Amazon Mechanical Turk.

Clinc150 (Larson et al., 2019) comprises 150 intents and 23,700 examples spanning 10 domains.

4.2 Baselines

We compare the proposed few-shot text classification models with the following baselines:

Prototypical Network (Snell et al., 2017) learns class prototype representations and employs nearest-neighbor classification, emphasizing instance similarity to prototypes.

MAML (Finn et al., 2017) optimizes for rapid adaptation to new tasks through iterative training across multiple tasks, acquiring generic initial parameters for quick learning and generalization across diverse tasks.

Induction Network (Geng et al., 2019) uses dynamic routing to represent and condense samples within categories into class-level representations, facilitating query sample classification.

HATT (Gao et al., 2019) utilizes a hybrid attention mechanism, encompassing both instance-level and feature-level attention, to enhance robustness and expedite the iteration speed of the model.

DS-FSL (Bao et al., 2020) focuses on learning the relationship between the importance of words and distributional signatures, thereby weighting to obtain a more refined sample representation.

MLADA (Han et al., 2021) is a meta-learning framework integrated with an adversarial domain adaptation network, aiming to improve the adaptive ability of the model and generate high-quality text embedding for new classes.

ContrastNet (Chen et al., 2022) through supervised contrastive learning and two unsupervised contrastive learning approaches at the task level and instance level, achieved more discriminative prototype representations and alleviated the issue of overfitting.

TPN (Liu et al., 2019b) employs a transductive approach by constructing an undirected graph that integrates all unlabeled and labeled data, obtaining labels for all unlabeled data through label propaga-

Method	HuffPost		Amazon		Reuters		20News		Average	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Prototypical Networks	35.7	41.3	37.6	52.1	59.6	66.9	37.8	45.3	42.7	51.4
MAML	35.9	49.3	39.6	47.1	54.6	62.9	33.8	43.7	40.9	50.8
Induction Networks	38.7	49.1	34.9	41.3	59.4	67.9	28.7	33.3	40.4	47.9
HATT	41.1	56.3	49.1	66.0	43.2	56.2	44.2	55.0	44.4	58.4
DS-PSL	43.0	63.5	62.6	81.1	81.8	96.0	52.1	68.3	59.9	77.2
MLADA	45.0	64.9	68.4	86.0	82.3	**96.7**	59.6	77.8	63.9	81.4
ContrastNet	51.8	67.8	73.5	83.6	88.5	94.6	70.9	80.5	71.2	81.6
TPN	50.6	69.5	76.0	84.9	**91.4**	93.1	63.0	69.4	70.3	79.2
TART	45.7±1.5	68.7±2.0	71.7±6.6	83.8±3.2	87.6±0.9	95.2±0.7	72.0±4.2	83.6±3.4	69.3±3.3	82.8±2.3
Way-DE	51.9±2.4	71.7±2.4	76.1±6.3	87.4±3.2	90.6±0.7	95.2±0.8	71.0±4.0	83.2±4.1	72.4±3.4	84.4±2.6
SELP (Ours)	**66.1±4.0**	**73.0±3.1**	**80.4±5.0**	**87.9±3.3**	91.3±2.4	95.6±1.6	**77.5±4.2**	**85.2±3.6**	**78.8±3.9**	**85.4±2.9**

Table 2: The 5-way 1-shot and 5-shot average accuracy on news or review classification datasets.

Method	Banking77		HWU64		Liu57		Clinc150		Average	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
PROTAUGMENT	86.9	94.5	82.4	91.7	84.4	92.6	94.9	98.4	87.2	94.3
PROTAUGMENT (bigram)	88.1	94.7	84.1	92.1	85.3	93.2	95.8	98.5	88.3	94.6
PROTAUGMENT (unigram)	89.6	94.7	84.3	92.6	86.1	93.7	96.5	98.7	89.1	94.9
ContrastNet	91.2	96.4	86.6	92.6	85.9	93.7	96.6	98.5	90.1	95.3
TPN	90.4	94.8	83.7	91.5	86.6	93.2	97.1	98.1	89.5	94.4
TART	89.5±1.0	94.7±0.5	85.4±1.7	93.4±0.9	87.9±2.0	94.5±1.0	96.4±0.6	98.7±0.2	89.8±1.3	95.3±0.7
Way-DE	90.5±1.6	95.4±1.0	87.1±1.9	93.4±1.1	90.4±2.2	**95.5±1.1**	98.0±0.5	**99.3±0.2**	91.5±1.6	95.9±0.9
SELP (Ours)	**92.1±1.0**	**96.5±0.4**	**89.8±1.3**	**93.9±0.8**	**91.7±1.4**	94.9±0.9	**98.2±0.5**	99.1±0.2	**92.9±1.1**	**96.1±0.6**

Table 3: The 5-way 1-shot and 5-shot average accuracy on intent detection datasets.

tion.

PROTAUGMENT (Dopierre et al., 2021) is a data augmentation technique that uses a model to generate paraphrases of short texts. It applies an unsupervised loss at the instance level on the vanilla prototypical network (Snell et al., 2017). PROTAUGMENT (unigram) and PROTAUGMENT (bigram) employ different strategies for word paraphrasing.

TART (Lei et al., 2023) transform the class prototypes to per-class fixed reference points in task-adaptive metric spaces and use a discriminative reference regularization to further maximize divergence between transformed prototypes.

Way-DE (Liu et al., 2023) assumes a Gaussian distribution for each class and utilizes the original support set along with the nearest minority query samples to estimate the mean and covariance. Subsequently, it augments labeled samples by sampling from the estimated distribution.

4.3 Implementation Details

Evaluation Metric We follow Liu et al. (2023) to use accuracy to assess the performance of our model. The datasets used in our experiments, as provided by Chen et al. (2022), consist of five random class partitions for each dataset. All reported results are averages obtained across these five partitions.

Parameter Settings We follow Liu et al. (2023) to conduct experiments on 5-way 1-shot and 5-shot settings. Across the news or review classification datasets, we employed bert-base-uncased model as the feature extractor. Our reported average accuracy is based on 1000 episodes sampled from the test set, with each episode comprising 25 query instances. Across the intent detection datasets, we employed a further pre-trained BERT model provided in Dopierre et al. (2021) as the feature extractor. Our reported average accuracy is based on 600 episodes sampled from the test set, with each episode comprising 5 query instances. We set the number of prompts T, to 8, and the length of the prompt is chosen from {8, 16} using the validation set. In the simulation of label distribution, we set β to 4.0. We optimize the models using AdamW with an initialized learning rate of 2e-5. In the prototype calibration stage, we opt for distinct values of α for each dataset from {0.5,0.6,0.7,0.8} based on the performance of the validation set.

4.4 Main Results

Tables 2 and 3 report the experimental results for the news or review classification task and the intent detection task. Most baseline results are taken from Liu et al. (2023). The results for TART in Table 2 and Table 3 are obtained from our re-run of their

Method	HuffPost		HWU64	
	1-shot	5-shot	1-shot	5-shot
Ours	**66.1±4.0**	**73.0±3.1**	**89.8±1.3**	**93.9±0.8**
-$\mathcal{L}_{lcm}$	65.6±3.9	72.1±2.8	87.5±1.5	93.2±0.9
-$\mathcal{L}_{cen}$	63.1±3.1	71.8±2.6	88.9±1.4	92.9±0.7
-$PromptPool$	62.4±3.6	70.1±3.0	86.7±1.3	92.7±0.9
-$Label$	35.7	41.3	82.4	91.7

Table 4: Ablation study results.

experiment. The best results are highlighted in bold.

From Table 2, it is evident that our approach achieves state-of-the-art performance on most datasets, particularly excelling in the 1-shot setting with an average improvement of 6.4% over other baselines. In the 1-shot setting of the HuffPost dataset, our method outperforms the best baseline by 14.2%. The reason behind this lies in the 1-shot scenario, where prototypes, being computed from a single sample, are more prone to deviate from class clusters. Our method effectively pulls back these deviated prototypes, leading to superior performance. This underscores the superiority of our approach in handling few-shot tasks. Besides, we notice that the performance gains for 1-shot intent detection are smaller than the gains for 1-shot text classification. We believe there are two main reasons for that: (1) The baselines of the intent detection datasets are very high, and there is not much room for improvement. (2) The texts in the intent datasets are shorter and provide limited information, which is a limitation for our method.

As depicted in Table 3, our method demonstrates consistent effectiveness, outperforming baselines on the majority of datasets. Similarly, our approach exhibits more pronounced improvements in the 1-shot setting, 1.6% in 1-shot on Banking77, and 2.7% in 1-shot on HWU64, respectively This further validates the effectiveness of our method for both long and short texts.

4.5 Visualization

In Figure 1, we employ t-SNE to visualize the query and prototype representations generated by ContrastNet, TART, and our method. We sample 5 classes from the test set of HWU64, sampling one prototype and 25 samples for each class. From Figure 1(a), ContrastNet roughly distinguishes class textual representations. However, the prototype representations sometimes significantly deviate from the class clusters, leading to a substantial impact on the classification accuracy of ContrastNet. As illustrated in Figure 1(b), TART maps the representations to another space, resulting in a different distribution compared to the other two methods. It can be observed that, although TART attempts to separate the prototypes as much as possible, the query samples are intermingled without clear boundaries. In Figure 1(c), our approach demonstrates that similar classes still cluster together. However, class prototypes are drawn back to their respective semantic regions through label calibration. Furthermore, the intra-class cohesion is relatively enhanced, and inter-class boundaries are clearer, affirming the effectiveness of our method.

4.6 Ablation Study

To demonstrate the impact of each component of the model, we conduct ablation experiments on the HuffPost and HWU64 datasets, as shown in Table 4.

The SELP -$\mathcal{L}_{lcm}$ donates that we do not apply $\mathcal{L}_{lcm}$ (Eq.12) and directly use one-hot label and cross-entropy loss. The decrease in the model's performance indicates the effectiveness of the designed LCM loss.

The SELP -$\mathcal{L}_{cen}$ donates that we do not apply $\mathcal{L}_{cen}$ (Eq.7) and solely employ $\mathcal{L}_{lcm}$ as the loss function for model updates. There is also a decline in the effectiveness of the model.

The SELP -$PromptPool$ donates that we excluding the prompt pool, we directly employ the embeddings of the raw labels to calibrate the prototypes and compute the overall loss for model updates. The model exhibits a significant performance decline, indicating that the prompt pool has learned meta-knowledge, which enhances the representation of labels.

When removing the label information (SELP-$Label$), the model regresses to the original prototype network. And the model performs the worst, indicating the strong guidance role of label information in classification.

5 Conclusion

In this paper, we employ a prompt pool to compute a distinctive prompt for each label. This prompt encompasses learned transferable meta-knowledge, providing additional semantic information related to instances. Subsequently, the prototypes are calibrated using these enhanced labels, resulting in prototypes that are more separated, semantically accurate, and closer to their respective class clusters. Additionally, we introduce a straightforward center loss to enhance intra-class compactness and apply a simulated label distribution method to mitigate overfitting issues. Experimental results affirm the effectiveness of our proposed approach.

Limitations and Potential Risks

Limitations In this paper, we employed BERT as the text encoder. While our proposed approach is applicable to any PLM that provides text representations, we did not conduct experiments on other PLMs due to time and efficiency constraints. Additionally, our method relies on meta-learning and therefore requires at least one available training episode. In future work, we aim to further investigate the application of labels in the context of small-sample and zero-shot scenarios.

Potential Risks Our research is dedicated to investigating how to enhance natural language understanding under low-resource conditions, with a focus on improving the performance of text classification. Our efforts contribute to an uplift in text classification efficiency, with no inherent risks posed to society or individuals.

Acknowledgment

This work was supported by National Natural Science Foundation of China (No. 62106035, 62206038), and Chunhui Project Foundation of the Education Department of China (No. HZKY20220419).

References

Antreas Antoniou, Harrison Edwards, and Amos J. Storkey. 2019. How to train your MAML. In *International Conference on Learning Representations (ICLR)*.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot text classification with distributional signatures. In *International Conference on Learning Representations (ICLR)*.

Inigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, page 38.

Junfan Chen, Richong Zhang, Yongyi Mao, and Jie Xu. 2022. Contrastnet: A contrastive learning framework for few-shot text classification. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 10492–10500.

Daiki Chijiwa, Shin'ya Yamaguchi, Atsutoshi Kumagai, and Yasutoshi Ida. 2022. Meta-ticket: Finding optimal subnetworks for few-shot learning within randomly initialized neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 25264–25277.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186.

Thomas Dopierre, Christophe Gravier, and Wilfried Logerais. 2021. Protaugment: Intent detection meta-learning through unsupervised diverse paraphrasing. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2454–2466.

Jiangshu Du, Wenpeng Yin, Congying Xia, and Philip S. Yu. 2023. Learning to select from multiple options. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 12754–12762.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, pages 1126–1135.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3816–3830.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 6407–6414.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3904–3913.

Biyang Guo, Songqiao Han, Xiao Han, Hailiang Huang, and Ting Lu. 2021. Label confusion learning to enhance text classification models. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 12929–12936.

Chengcheng Han, Zeqiu Fan, Dongxiang Zhang, Minghui Qiu, Ming Gao, and Aoying Zhou. 2021. Meta-learning adversarial domain adaptation network for few-shot text classification. In *Findings of the Association for Computational Linguistics (Findings of ACL)*, pages 1664–1673.

Ruining He and Julian J. McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *The Web Conference (WWW)*, pages 507–517.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 328–339.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Computer Vision and Pattern Recognition (CVPR)*, pages 7482–7491.

J. Ashok Kumar and S. Abirami. 2021. Ensemble application of bidirectional LSTM and GRU for aspect category detection with imbalanced data. *Neural Computing Applications (Neural Comput. Appl.)*, pages 14603–14621.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *International Conference on Machine Learning (ICML)*, pages 331–339.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1311–1316.

Shuo Lei, Xuchao Zhang, Jianfeng He, Fanglan Chen, and Chang-Tien Lu. 2023. TART: improved few-shot text classification using task-adaptive reference transformation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 11014–11026.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4582–4597.

Han Liu, Feng Zhang, Xiaotong Zhang, Siyang Zhao, Fenglong Ma, Xiao-Ming Wu, Hongyang Chen, Hong Yu, and Xianchao Zhang. 2023. Boosting few-shot text classification via distribution estimation. pages 13219–13227.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019a. Benchmarking natural language understanding services for building conversational agents. In *International Workshop on Spoken Dialogue Systems (IWSDS)*, pages 165–183.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. 2019b. Learning to propagate labels: Transductive propagation network for few-shot learning. In *International Conference on Learning Representations (ICLR)*.

Qiaoyang Luo, Lingqiao Liu, Yuhao Lin, and Wei Zhang. 2021. Don't miss the labels: Label-semantic augmented meta-learner for few-shot text classification. In *Findings of the Association for Computational Linguistics (Findings of ACL)*, pages 2773–2782.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, pages 140:1–140:67.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning (ICML)*, pages 1842–1850.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *European Chapter of the Association for Computational Linguistics: Main Volume (EACL)*, pages 255–269.

Zhiqiang Shen, Zechun Liu, Jie Qin, Marios Savvides, and Kwang-Ting Cheng. 2021. Partial is better than all: Revisiting fine-tuning strategy for few-shot learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 9594–9602.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 4077–4087.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 16857–16867.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. 2022. Learning to prompt for continual learning. In *Computer Vision and Pattern Recognition (CVPR)*, pages 139–149.

Haoxing Zhang, Xiaofeng Zhang, Haibo Huang, and Lei Yu. 2022. Prompt-based meta-learning for few-shot text classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1342–1357.