Learn to Adapt for Generalized Zero-Shot Text Classification

Anonymous ACL submission

Abstract

Generalized zero-shot text classification aims to classify textual instances from both previously seen classes and incrementally emerging unseen classes. Most existing methods generalize poorly since the learned parameters are only optimal for seen classes rather than for both classes, and the parameters keep stationary in predicting procedures. To address these challenges, we propose a novel Learn to Adapt (LTA) network using a variant meta-learning framework. Specifically, LTA trains multiple meta-learners by using both seen classes and virtual unseen classes to simulate a generalized zero-shot learning (GZSL) scenario in accordance with the test time, and simultaneously learns to calibrate the class prototypes and sample representations to make the learned parameters adaptive to incoming unseen classes. We claim that the proposed model is capable of representing all prototypes and samples from both classes to a more consistent distribution in the global space. Extensive experiments on five text classification datasets show that our model outperforms several competitive previous approaches by large margins. The code and the whole datasets will be available after paper publication.

1 Introduction

Text classification plays an important role in many natural language processing (NLP) applications, such as question classification, news categorization, user intent classification and so on (Minaee et al., 2021). Although a wide variety of methods have been proved successful in supervised text classification, they often break down when applied to make predictions for incrementally emerging classes without labeled training data (Pourpanah et al., 2020). Unlike zero-shot learning (ZSL) that aims to classify unseen class instances at test time (Romera-Paredes and Torr, 2015; Wang et al., 2019), generalized zero-shot learning (GZSL), that we focus on in this work, aims to classify text samples from both previous seen and emerging novel classes. Since there is a strong bias towards seen classes (Xian et al., 2017), GZSL is a more challenging yet critical problem.

Previously methods mainly focus on transductive approaches for generalized zero-shot text classification. Rios and Kavuluru (2018) use a graph convolution network to enhance the unseen class label embeddings. Zhang et al. (2019) and Song et al. (2020) generate illusion feature embeddings for unseen classes based on side information, i.e., class-level attributes or text description. More recently, Ye et al. (2020) use reinforced self-training methods to leverage unlabeled data during training stage.

With the assumption that no knowledge about unseen categories is available during the model learning phase, researchers resort to inductive approaches to handle generalized zero-shot text classification. ReCapsNet (Liu et al., 2019) uses a dimensional attention-based intent capsule network and constructs zero-shot class prototypes by similarity matrix transformation. SEG (Yan et al., 2020) exploits an outlier detection approach that can be directly applied on ReCapsNet, which discriminates the domain first, then outputs the final class label.

However, the existing methods still have two key limitations. Firstly, while the goal of these methods is to transfer beneficial knowledge for unseen classes, these models merely learn optimal parameters by minimizing the loss of instances from seen classes, regardless of explicitly calibrating the predictions on unseen classes. Therefore, domain bias towards seen classes is not fairly resolved (Vinyals et al., 2016). Secondly, although some of them take into account the inter-class relationship when constructing prototypes for unseen classes (Liu et al., 2019), the models keep static no matter what different new classes emerging in future applications. As a result, these models show...
a large quality gap between instances from seen classes and from emerging unseen classes.

To address these problems, motivated by the success of meta-learning in the few-shot learning task (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018; Finn et al., 2017), we present a novel Learn-To-Adapt network (LTA) for generalized zero-shot text classification. Concretely, the proposed LTA learns class prototypes over multiple learning episodes that mimic GZSL setting explicitly during training, making the learning setting consistent with the test environment and thereby improving generalization. Then, the model notably extends its ability from two views: prototype adaptation and sample adaption. In each episode, the LTA adjusts the representative prototypes of both seen classes and "fake" unseen classes, with the assumption that unseen class will help in calibrating representation of seen ones and thereby enable the model to learn the class sensitive representations. The updating for all prototypes is then used to generate a set of calibration parameters to guide the adaption of sample embeddings, which is designed to compensate for the shrinking features (Chen et al., 2018) that are ignored during training if they are not discriminating for seen classes, but could be critical for recognizing unseen classes. The refined sample embeddings are then classified based on similarity scores with all class prototypes. The same setting can be directly applied in test, where the LTA executes class prediction and adapts the learnt model rationally in an on-the-fly manner.

In summary, our contributions include: (i) We propose a novel Learn to Adapt (LTA) network for generalized zero-shot text classification which is capable of adapting incrementally between seen classes and emerging unseen classes at test time. (ii) We propose a methodology for calibrating both prototypes and samples to deduce a global representation space, efficiently avoiding over-fitting on seen classes. (iii) Experimental results on five generalized zero-shot text classification datasets show that our method outperforms previous methods with a large margin.

2 Related Work

Generalized Zero-Shot Learning  The challenge of zero-shot learning (ZSL) has been the focus of attention in recent years, especially in the applications of image classification (Socher et al., 2013; Xian et al., 2017; Wang et al., 2018, 2019), intent classification (Xia et al., 2018; Liu et al., 2019; Yan et al., 2020), and question classification (Fu et al., 2018). Different from ZSL, generalized zero-shot learning (GZSL) that attempts to categorize instances from both seen and unseen classes is a more realistic condition that matches with practical applications. For example, a question classifier for question answering system has to classify not only the questions ever asked but also new questions incrementally emerging from the users.

There are two key issues that GZSL has to address: (1) how to incrementally learn beneficial knowledge for unseen classes from seen ones, and (2) how to tackle the domain bias caused by the extremely imbalanced data of seen and unseen domains.

To alleviate the first issue, some of the earliest works on ZSL attempt to learn a matching model between instance embedding and class prototype embeddings represented by extra information including class-level attribute, text description, or their combinations (Frome et al., 2013; Jinseok Nam, 2016; Zhu et al., 2019; Xia et al., 2018). In a similar vein, other methods (Wang et al., 2018; Rios and Kavuluru, 2018; Si et al., 2020) also investigate the semantic relationship between the side information for obtaining better prototype representation.

The key problem of the second issue is that the model is trained with data from the seen classes and the parameters are actually optimized on seen domain, thus they are not aware of unseen classes. Assuming the extra information about unseen classes is available, another prominent approach attempts to use generative models to generate virtual samples or features for unseen domains (Xian et al., 2018; Schönfeld et al., 2019; Zhang et al., 2019; Song et al., 2020). By using synthesized samples, the generative approaches can convert GZSL problem to the conventional supervised learning problem where biases towards seen classes are largely alleviated. Nevertheless, these models are trained using data from seen classes and fails to incrementally adapt to emerging new classes. Additionally, studies also extend to exploit the unlabeled data for unseen classes (Xian et al., 2019; Rahman et al., 2019; Ye et al., 2020).

However, these models assume that they have access to the extra information about the unseen classes, which is not very realistic since often neither the test data nor their label descriptions is
available at the training phase (as supposed in this work). In contrast, our model can involve all classes (seen and unseen) jointly during inference, essentially it is trained towards continuous generalization for new classes, hence it is capable to adapt to incoming new class dynamically.

**Episode-Based Training in GZSL** Our approach is primarily based on episodic training/meta-learning that has been widely used in few-shot learning (FSL) (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018). The primitive goal of episodic training is to quickly learn a meta-task from a small number of sampled classes and supporting sets. A particular advantage of episodic learning is that, by constructing meta-tasks, the setting of training is consistent with that of test, which is essential for classification problems.

Studies extend to exploit episodic training in the "generalized" settings. Gidaris and Komodakis (2018); Ye et al. (2019); Shi et al. (2019) utilize weight generators or relationships to update representative prototypes in generalized FSL (GFSL). Yu et al. (2020) use a generative network to generate unseen prototypes in GZSL. These methods only consider the prototype adaptation while the sample embeddings are still static whatever the unseen classes are. On the contrary, Bao et al. (2020) uses distributional signatures to update sample embeddings in GFSL. Considering that distributional signatures can be equal for two different tasks, our method uses a novel semantic update extractor to update samples following the prototype adaptation rather than statistical information.

A compelling property of our method is that it tackles knowledge transferring and domain bias simultaneously in an adaptive episodic training framework by adapting both prototypes and sample embeddings, and draws a fast adaption to the novel classes without the cost of dramatic damage in discriminating the seen classes.

**3 Methodology**

**3.1 Problem Definition**

Formally, let \( Y^s = \{y^s_1, ..., y^s_{K^s}\} \) and \( Y^u = \{y^u_1, ..., y^u_{K^u}\} \) denote \( K^s \) seen classes and \( K^u \) unseen classes respectively, and \( \mathcal{Y} = Y^s \cup Y^u \) denote the global label space with \( Y^s \cap Y^u = \emptyset \). Suppose we have a collection of training samples \( \mathcal{D}^s = \{(x^i, y^s_i, a^s_i)\}_{i=1}^M \), that consists of \( M \) samples from \( K^s \) seen classes, where \( x^i \in \mathcal{X}^s \) represents \( i \)-th text utterance, \( y^s_i \) is and \( a^s_i \) are its one-hot class label and class-level textual description, respectively. At the test time, provided with a class description set \( \mathcal{A}^u = \{a^u_j\}_{j=1}^{K^u} \) for unseen classes, the GZSL task is to classify the test instance into either a seen or an unseen class.

**3.2 Overview**

**Encoder** An textual input \( x \) with \( T \) words is encoded by a BERT (Devlin et al., 2018) into a sequence of hidden vectors \( \mathbf{H} = [h_1, h_2, ..., h_T] \in \mathbb{R}^{T \times d_h} \), where \( d_h \) is the hidden dimension. The text embedding \( f(x) \in \mathbb{R}^{d_h} \) is then obtained by averaging over the \( T \) hidden vectors.

**Training** In the training stage, we introduce an episodic learning paradigm, which trains the model by simulating multiple generalized zero-shot text classification tasks on seen classes. Following the principle that train and test conditions must match (Vinyals et al., 2016) and recent studies on "generalized" setting (Gidaris and Komodakis, 2018; Shi et al., 2019; Ye et al., 2019; Bao et al., 2020; Verma et al., 2020; Yu et al., 2020), each episode involves an \( N^s \)-way \( K \)-shot learning task for seen classes, denoted as \( \mathcal{D}^s_i = \{(x^i_j, y^s_j, a^s_j)\}_{j=1}^{N^s \times K} \) with \( K \) labelled instances for each of \( N^s \) classes randomly sampled from the seen data \( \mathcal{D}^s \) in the \( i \)-th episode, and a \( N^u \)-way \( K \)-shot learning task for "fake" unseen classes, denoted as \( \mathcal{D}^u_i = \{(x^u_j, y^u_j, a^u_k)\}_{k=1}^{N^u} \) which is also from \( \mathcal{D}^s \), with \( N^s + N^u \leq C^s \). More precisely, let \( \mathcal{Y}^s \) and \( \mathcal{Y}^u \) denote the sampled seen class space and sampled "fake" unseen class space respectively, with \( \mathcal{Y}^s \subset \mathcal{Y}^s \) and \( \mathcal{Y}^u \subset \mathcal{Y}^u \), and \( \mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset \). For a new query instance \( x \), the generalized zero-shot learning model performs

\[
\hat{y} = \arg \max_{y \in \mathcal{Y}^s \cup \mathcal{Y}^u} p(y|x, \mathcal{D}^s_i, \mathcal{D}^u_i) \tag{1}
\]

The model is designed to maintain a globally joint class prototype space as well as dynamic adaptation to unseen classes with zero labeled instances, whose detailed implement is described as follows. 

**3.3 Prototype Adaptation**

The proposed LTA network first introduces a learnable look-up table \( \mathbf{S} \in \mathbb{R}^{C^s \times d_h} \) from which to extract the seen prototypes \( \mathbf{S}_i \in \mathbb{R}^{N^s \times d_h} \) on demand. The \( \mathbf{S} \) is initialized using the supervised classifier by reducing the error on the training samples from the random initialization. The virtual unseen prototypes \( \mathbf{U}_i \) is produced by the BERT encoder
\[ f(\cdot) \text{ using their corresponding class descriptions:} \]
\[ U_t = [f(a_y)]_{y \in \mathcal{Y}_r} \in \mathbb{R}^{N_c \times d_h}. \]

Then the joint prototype matrix \( R \) is obtained by concatenating \( S_t \) and \( U_t \), \( R = [S_t, U_t] \in \mathbb{R}^{C^* \times d_h} \) with \( r_j \) as the \( j \)-th prototype. Then \( R \) is fed into an inter-class Transformer encoder (Vaswani et al., 2017) to explicitly model the updates for the representations of both seen prototypes and novel prototypes:

\[
Z = \text{TransformerEncoder}(R)
\]
\[
\hat{R} = R + Z
\]

where \( Z \in \mathbb{R}^{C^* \times d_h} \) highlights the adjustment after mutual reflections, and the updated prototypes \( \hat{R} \in \mathbb{R}^{C^* \times d_h} \) is regarded as the calibrated representative prototypes of both seen and unseen categories, with \( \hat{r}_j \) as the adjusted \( j \)-th prototype. The self-attentions used in Transformer is agile to capture the inter-class relationship of seen and unseen classes and thereby it is beneficial to derive globally discriminative prototypes. The prototypes simultaneously update both seen and unseen classes, which enables the model to represent and discriminate the newly incoming categories in an on-the-fly manner.

### 3.4 Sample Adaptation

As been discussed in (Chen et al., 2018), the zero-shot learning tasks are prone to produce semantics loss, where some features would be discarded during training if they are not discriminating for seen classes, but critical for recognizing unseen classes. We observe that the similar problem is exacerbated in GZSL task due to the extreme unbalance between seen and unseen classes. We tackle this problem by introducing sample adaption following the trajectories of prototypes adaption. In concrete, we apply a semantic update extractor via attention mechanism to capture synchronous updating of the prototypes:

\[
F = ZW_1
\]
\[
A = \text{Softmax}(W_3\text{ReLU}(W_2F^T))
\]
\[
C = AF
\]

where \( W_1 \in \mathbb{R}^{d_h \times d_h}, W_2 \in \mathbb{R}^{d_c \times d_h}, W_3 \in \mathbb{R}^{d_c \times d_h} \) are trainable parameters, \( A \) denotes the attention weight matrix and \( C \in \mathbb{R}^{d_c \times d_h} \) extracts different semantic components with \( c_j \) is the \( j \)-th semantic components. To offset the semantic loss mentioned above, we compare the attention score for each \( h_t \) to get most related semantic adjustment and reconstruct the contribution of each feature:

\[
e_t = \text{Softmax}(\max_j (\frac{h_t c_j}{||h_t|| \cdot ||c_j||}))
\]
\[ g(x) = \sum_{t=1}^{T} e_t h_t \] (8)

where the self-attention weight \( e_t \in \mathbb{R}^T \) is used to re-weight the \( t \)-th word of sample \( x \) to be classified, and \( \beta \) is a scalar to control the differentiation of attention scores. In this way, the different attention weight discriminate the importance of words rather than averaging them.

One notable reason of choosing of the above feature-level calibration is that, in classification task, the encoder is trained to produce feature embeddings that collapses to its ground-truth prototype, therefore the adjustment of feature embedding should cater to the adjustment of a reliable global prototype space. In addition, since this calibration is applied after the encoding, it reduces the complicated parameter tuning for a massive encoder (e.g., BERT), which elegantly helps the GZSL task to fast adapt to the incoming test instances.

### 3.5 Loss function

With the adapted prototypes \( \hat{R} \) and the adapted sample \( g(x) \), a Softmax classifier is used with cosine similarity:

\[
p(\hat{y} = y \mid x) = \frac{\exp(s(g(x), \hat{r}_y))}{\sum_y \exp(s(g(x), \hat{r}_y))} \] (9)

where \( s(a, b) = \frac{a \cdot b}{\|a\| \|b\|} \) is cosine similarity with temperature \( \tau \). Finally the model is trained to minimize the losses across all episodes:

\[
\mathcal{L} = -\frac{1}{N} \sum_i \mathcal{L}_i \] (10)

where \( \mathcal{L}_i \) is the loss of the \( i \)-th episode:

\[
\mathcal{L}_i = \frac{1}{M} \sum_{(x, y, a) \in D_i^u \cup D_i^s} \log p(\hat{y} = y \mid x) \] (11)

The training process of LTA is summarized in Algorithm 1.

### 4 Experiments

#### 4.1 Datasets

**Intent Classification Datasets.** We collect four intent classification datasets. (1) **SNIPS-SLU** (Coucke et al., 2018), a widely used benchmark for English GZSL intent detection with 5 seen intents and 2 unseen intents. (2) **SMP-18** (Zhang et al., 2017), a Chinese dialogue corpus for user intent detection with 24 seen intents and 6 unseen intents. (3) **ATIS** (Hemphill et al., 1990), an English airline travel domain dataset, from which we extract 17 intents with at least 5 samples, and split them into 12 seen intents and 5 unseen intents. (4) **CLINC** (Larson et al., 2019) is a recently published intent detection dataset includes 22,500 in-scope queries covering 150 intent classes from 10 domains. We randomly split the 150 intents into 120 seen intents and 30 unseen intents.

**Question Classification Dataset.** In order to draw a comprehensive analysis of the proposed method, we construct a question classification task from the Quora Question Pairs dataset \(^1\), which is aimed to identify duplicate questions. We collect questions with at least 5 duplicate samples into classes. In each class, we choose the question with minimum words as the label description, which is widely used in real-world question-answering systems (Sakata et al., 2019). Table 1 summarizes all datasets statistics. It is worth to note that intents in ATIS are highly unbalanced with flight accounts for about 87% of training data.

**Dataset Settings.** Following (Siddique et al., 2021), we use random seen/unseen classes for 10 runs instead of manual selection used in (Yan et al., 2020), which leads to more fair results because every class could be unseen class. We randomly take

\(^1\)www.kaggle.com/c/quora-question-pairs
Table 1: Dataset statistics. “FS” indicates “few-shot”, “BAL” indicates “balance”, “IBAL” indicates “imbalance”. The “avg #samples” indicates the average number of samples per class.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#classes</th>
<th>seen samples</th>
<th>unseen samples</th>
<th>total samples</th>
<th>#sent len</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNIPS</td>
<td>5</td>
<td>13802 1384</td>
<td>6313</td>
<td>20115</td>
<td>9.10</td>
<td>BAL</td>
</tr>
<tr>
<td>SMP</td>
<td>24</td>
<td>2460 60</td>
<td>2400 360</td>
<td>4860</td>
<td>4.83</td>
<td>FS</td>
</tr>
<tr>
<td>ATIS</td>
<td>12</td>
<td>4972 245</td>
<td>22500 105</td>
<td>24755</td>
<td>11.44</td>
<td>IBAL</td>
</tr>
<tr>
<td>Clinc</td>
<td>120</td>
<td>22500 105</td>
<td>43100 2050</td>
<td>66600</td>
<td>8.23</td>
<td>BAL</td>
</tr>
<tr>
<td>Quora</td>
<td>1360</td>
<td>17394 7</td>
<td>1360 340</td>
<td>18754</td>
<td>10.46</td>
<td>FS</td>
</tr>
</tbody>
</table>

70% samples of each seen class as the training set, and the remaining 30% samples of each seen class as the seen test and take all the samples of unseen classes as the unseen test. All the textual labels of the same class are regarded as the description for this class.

4.2 Baseline Methods

To validate the benefits of the proposed LTA, we compare against with other approaches in three aspects:

**Supervised Learning Methods.** To show the performances on seen classes with supervised learning instead of ZSL/GZSL setting, we use (1) **BiLSTM** (Schuster and Paliwal, 1997) and (2) **BERT** (Devlin et al., 2018) as the encoder with a linear softmax classifier.

**Metric Learning Methods.** Metric-based embedding methods are commonly used as baselines for ZSL/GZSL. Thus we introduce three different metric learning methods: (1) **EucSoftmax**: We adapt (Snell et al., 2017) that uses squared Euclidean distance as the metric and softmax classifies; (2) **Zero-shot DNN**: We adapt (Kumar et al., 2017) that uses squared Euclidean distance and triplet loss to maintain a margin for different classes. We choose the label embedding(prototype) as the anchor and the closest sample as negative sample in each triplet tuple; (3) **CosT** (Gidaris and Komodakis, 2018) refers to Cosine Distance with temperature scalar $s(a, b) = \tau \cos(a, b)$ where $\tau$ is a learnable temperature scalar to dynamically control the peakiness of the probability distribution generated by the Softmax.

**SOTA Methods.** We also compare our model with two recent state-of-the-art (SOTA) methods: (1) **ReCapsNet** (Liu et al., 2019) uses a dimensional attention-based intent capsule network and a matrix transformation method for ZSL/GZSL. (2) **SEG** (Yan et al., 2020) is an outlier detection approach that can be directly applied on ReCapsNet. SEG acts as a domain discriminator which first determines whether a test sample belongs to seen classes or unseen classes and then classifies in their own domain. RIDE (Siddique et al., 2021) is not considered because they use outer knowledge not available in our settings and the data they used is limited within the intent detection task due that the labels in their method need to be a combination of "action" and "object".

4.3 Experimental Setup

**Evaluation Metrics.** We basically use accuracy (Acc) to estimate the performances on seen and unseen test sets. Besides, we adopt Macro-F1 (F1) rather than Micro-F1 to better evaluate the performances on imbalanced and few-shot datasets, because Macro-F1 gives the same weight of F1 scores for each class. For overall assessments, we adopt the widely used Harmonic Mean (HM) of Acc and F1 on Seen and Unseen, because the overall Acc and F1 scores are influenced by the ratio of seen and unseen test set sizes.

**Implementation Details.** We use pretrained BERT-base encoder with $d_h = 768$ on intent classification datasets and BiLSTM with 128 hidden size each direction on Quora dataset as basic encoder. The scalars of our model is set to be $\tau = 10.0$, $\beta = 10.0$, $d_a = d_h$, which is trained via Adam (Kingma and Ba, 2015) optimizer, with learning rates $10^{-5}$ for BERT, $10^{-4}$ for BiLSTM and $10^{-3}$ for the others. During training, we set $K = 5$ and $C^u = [2, 2, 2, 10, 20]$, $d_r = [4, 16, 32, 64, 64]$ for SNIPS, SMP, ATIS, CLINC and Quora datasets, respectively. The learnable $R$ is initialized from the prototypes trained from metric learning methods, and is used as our basic baseline. We also conduct an ablation study to investigate the effectiveness of each proposed component. As depicted in Table 2 and Table 3, "w/o Init" refers to the model that randomly initialize $R$. "w/o SA" refers to the model that only uses prototype adaptation without "Sample Adaptation". "w/o" means none of the adaptation steps is applied.

4.4 Results

The results on four intent datasets and Quora dataset are given in Table 2 and Table 3. Our
Table 2: Results (in %) on four intent benchmarks. The Top1 results of GZSL methods are highlighted in bold and underline for Top2 results, the same below.

<table>
<thead>
<tr>
<th>Model</th>
<th>SNIPS-NLU</th>
<th>SMP-18</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Seen</td>
<td>Unseen</td>
</tr>
<tr>
<td></td>
<td>Acc F1</td>
<td>Acc F1</td>
</tr>
<tr>
<td>Bi-LSTM</td>
<td>98.23</td>
<td>98.23</td>
</tr>
<tr>
<td>BERT</td>
<td>98.91</td>
<td>98.91</td>
</tr>
<tr>
<td>EucoSoftmax</td>
<td>81.09</td>
<td>65.50</td>
</tr>
<tr>
<td>Zero-shot DNN</td>
<td>81.09</td>
<td>65.28</td>
</tr>
<tr>
<td>CosT</td>
<td>91.68</td>
<td>75.76</td>
</tr>
<tr>
<td>ReCapsNet + SEG</td>
<td>96.26</td>
<td>67.70</td>
</tr>
<tr>
<td>LTA (Ours)</td>
<td>74.05</td>
<td>74.11</td>
</tr>
<tr>
<td>w/o Init</td>
<td>82.57</td>
<td>75.22</td>
</tr>
<tr>
<td>w/o SA</td>
<td>67.31</td>
<td>70.56</td>
</tr>
<tr>
<td>w/o A</td>
<td>75.26</td>
<td>71.82</td>
</tr>
</tbody>
</table>

Table 3: Results (in %) on Quora question classification dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>ATIS</th>
<th>CLINC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Seen</td>
<td>Unseen</td>
</tr>
<tr>
<td></td>
<td>Acc F1</td>
<td>Acc F1</td>
</tr>
<tr>
<td>Bi-LSTM</td>
<td>93.24</td>
<td>79.51</td>
</tr>
<tr>
<td>BERT</td>
<td>97.18</td>
<td>97.31</td>
</tr>
<tr>
<td>EucoSoftmax</td>
<td>67.67</td>
<td>16.11</td>
</tr>
<tr>
<td>Zero-shot DNN</td>
<td>63.56</td>
<td>23.12</td>
</tr>
<tr>
<td>CosT</td>
<td>98.02</td>
<td>59.55</td>
</tr>
<tr>
<td>ReCapsNet + SEG</td>
<td>86.19</td>
<td>23.88</td>
</tr>
<tr>
<td>LTA (Ours)</td>
<td>93.75</td>
<td>40.90</td>
</tr>
<tr>
<td>w/o Init</td>
<td>96.28</td>
<td>63.15</td>
</tr>
<tr>
<td>w/o SA</td>
<td>89.96</td>
<td>47.48</td>
</tr>
<tr>
<td>w/o A</td>
<td>90.20</td>
<td>51.74</td>
</tr>
<tr>
<td>w/o A</td>
<td>94.94</td>
<td>63.25</td>
</tr>
</tbody>
</table>

proposed methods achieve the overall best performance compared to baselines.

Detailed and interesting observations can also be derived from the results: (1) Supervised Metric-Learning methods as the basic baselines, achieve comparable results on Seen Test for all datasets. However, it suffers from the domain bias problem and the performance drops with a large margin on Unseen Test, where the task is complex due to the imbalanced and few-shot scenarios. (2) The performances on SNIPS-NLU and SMP-18 of ReCapsNet and SEG are worse than those in their original paper although we use the open-source code, this is because we random split the test unseen classes which makes it more challenging. Besides, these methods fail to recognize unseen samples well on datasets with large scale of categories, yielding worse 0% Acc and F1 on Quora. The most likely reason is that ReCapsNet uses label embedding similarities to construct unseen prototypes in capsule network, which imposes a non-trivial computational and memory burden. (3) Our method shows its privilege for all datasets. In particular, with the help of continuous adapting ability, it observes smaller gap between seen and unseen domain, which proves the adaptation on testing phase effectively works. Although the performance on seen domain drops slightly, the proposed LTA outperforms the competitive metric-learning baselines by 9.54% HM Acc and 12.90% HM F1 average on the whole datasets, indicating that our model fairly balances the seen and unseen classes.

Ablation Study. To better understand the contribution of each component of our method, we explore three variants of LTA. We can observe that LTA with both prototype adaptation and sample adaptation outperforms those without adaptations in all cases. Generally “LTA w/o SA” with only prototype adaptation achieves better performance compared to “LTA w/o A”. The “LTA w/o Init” has relatively stable performances.
The model was trained under a consistent setting with testing. In particular, it efficiently alleviated the bias towards seen classes by utilizing both prototype adaptation and sample adaptation. Experiments on five text classification datasets validated that our model achieved compelling results on both seen classes and unseen classes, meanwhile the sample representations are independently clustered by the adapted prototype and easy to be distinguished.

To further study how the sample adaptation works, we select an representative case "when is it time for a tire change" and show its attention weights used as calibration parameters in (c). The case is still misclassified after the prototype adaptation due to the common word "time" and "change" also appear in seen classes. After the sample adaptation, however, it can be seen that the word "tire" which is a key word for classifying, get the most attention while the other confusing words are not. This result suggests that calibrating the attention weights is useful for acquiring a prototype aware representation which helps the sample adaptation.

4.5 Results on Emerging Unseen Classes

As the partition of seen / unseen classes is fixed in previous experiments, in order to study the robustness of the proposed adaption method, we conduct the experiment across unseen class sets of different scales. Specifically, we select 70 classes as seen classes and 10 classes as validating unseen classes. The testing unseen classes are randomly sampled from the remaining 70 classes and each experiment is repeated 50 times with different sampling sets for a more stable result. Figure 3 (a) shows the HM accuracy on all classes as the number of the unseen classes grows. We can see that our LTA model outperforms the metric baseline and ablation models in all cases, where the well performance is mainly attributed to the improvements on unseen classes as shown in Figure 3 (b). These results suggest that our adaptation method is robust and effective for adapting to new classes as well as improving the overall performance on all classes.

4.6 Visualization

To demonstrate how our adaptation method works, we further visualize the encoded representation via PCA in Figure 2. When there is no unseen class, seen classes (yellow and red) is discriminative enough. But when the new class "tire change" (purple) comes, it is ambiguous with class "oil change when" (red). We observe that the seen and unseen class prototypes are updated to be far away from each other after prototype adaptation as shown in (a), which eases the domain bias problem. However, the performance is unsatisfactory since the sample representations are still not discriminative no matter how prototype updates. As we can see, with the sample adaptation as shown in (b), the sample representations are independently clustered by the adapted prototype and easy to to be distinguished.

This paper proposed a novel adaptive meta-learning network for generalized zero-shot text classification. The model was trained under a consistent setting with testing. In particular, it efficiently alleviated the bias towards seen classes by utilizing both prototype adaptation and sample adaptation. Experiments on five text classification datasets validated that our model achieved compelling results on both seen classes and unseen classes, meanwhile...
was capable of fast adapting to new classes.

References


