Zero-shot Text Classification via Reinforced Self-training

Zhiquan Ye[1,2], **Yuxia Geng**[1,2], **Jiaoyan Chen**[4], **Xiaoxiao Xu**[3],
Suhang Zheng[3], **Feng Wang**[3], **Jingmin Chen**[3], **Jun Zhang**[3], **Huajun Chen**[†1,2]
[1]College of Computer Science, Zhejiang University
[2]AZFT Joint Lab of Knowledge Engine [3]Alibaba Group
[4]Department of Computer Science, Oxford University
{yezq,gengyx,huajunsir}@zju.edu.cn
{jiaoyan.chen}@cs.ox.ac.uk, {xiaoxiao.xuxx}@alibaba-inc.com
{suhang.zhengsh,wf135777,jingmin.cjm,zj157077}@alibaba-inc.com

Abstract

Zero-shot learning has been a tough problem since no labeled data is available for unseen classes during training, especially for classes with low similarity. In this situation, transferring from seen classes to unseen classes is extremely hard. To tackle this problem, in this paper we propose a self-training based method to efficiently leverage unlabeled data. Traditional self-training methods use fixed heuristics to select instances from unlabeled data, whose performance varies among different datasets. We propose a reinforcement learning framework to learn data selection strategy automatically and provide more reliable selection. Experimental results on both benchmarks and a real-world e-commerce dataset show that our approach significantly outperforms previous methods in zero-shot text classification.

1 Introduction

Zero-shot learning (ZSL) is a challenging task as no labeled data is available for unseen classes during training. There are extensive works proposed in zero-shot image classification task. The main focus of these works is how to transfer knowledge from seen classes to unseen classes. To associate unseen classes with seen classes, they usually resort to semantic information such as visual attributes (Lampert et al., 2009), word embeddings of class names (Norouzi et al., 2013) and class hierarchy (Socher et al., 2013). For example, if the model has not seen any instances of "humpback whale" in the training stage, it could still make predictions at testing stage since "humpback whale" is semantically close to "killer whale" and "blue whale" in the seen class set [*], so the model is capable of transferring knowledge from seen classes to unseen classes. These methods assume that semantically similar classes share similar image features, however, they may fail in the cases where classes share low similarities.

This problem becomes even more salient in typical NLP tasks such as text classification. For example, let us consider a 10-class emotion classification task (Yin et al., 2019), in which the model is trained on class "sadness" while makes predictions on instances from class "joy". Notice that most emotions are relatively independent, which means the way we express certain emotion is pretty different from other emotions. As a result, for an unseen class we can hardly find a similar class in the seen class set. Transferring from seen classes to unseen classes can be extremely hard as matching patterns that can be shared among classes are rare.

Essentially, ZSL methods aim to learn a matching model between feature space and semantic space, which refers to text and label in text classification task respectively. Matching patterns between text and label can be roughly classified as class-invariant patterns and class-specific ones. The former refers to the patterns that are shared among classes, while the latter is dependent on a certain class. Table 1 shows an example to illustrate this definition. The string match of label and text, which is highlighted with red color, indicates a simple matching pattern that can be shared among classes. On the contrary, the words that are highlighted with blue color indicates a matching pattern that is specific to a certain class and cannot be transferred among classes easily. Imagine if the model is trained on sentence 1, it can make a correct prediction on sentence 2 while failing on sentence 3 probably.

There are mainly two ways to deal with this troublesome zero-shot learning situation, including (1) integrating more external knowledge to

[†] Corresponding Author.
[*] This example is taken from awa2 dataset, https://cvml.ist.ac.at/AwA2/.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3014–3024
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

Label	Sentence
fear	1. One day, when I realized that I was alone, I felt fear of loneliness.
guilty	2. I felt guilty when I lied to my parents.
guilty	3. I wished secretly and lied to a friend because I didn't want her to stay in my house.

Table 1: Illustration of class-invariant and class-specific matching pattern.

better describe class and build more sophisticated connections between classes (Rios and Kavuluru, 2018; Zhang et al., 2019); (2) integrating the unlabeled data to improve the generalization performance. Generally, existing works mainly adopt the former solution, while little attention is paid to the latter one. In this paper, we focus on the latter one and propose a self-training based method to leverage unlabeled data. The basic idea of self-training (McClosky et al., 2006; Sagae, 2010) is to select unlabeled instances that are predicted with high confidence and add them into the training set. It is straightforward to consider that if we add sentence 2 to training set, the model is capable of learning class-specific pattern as sentence 2 and sentence 3 share the intra-class similarity. In this way, we can mine class-specific feature through class-invariant feature.

However, directly applying traditional self-training method to zero-shot learning may encounter some problems: (1) traditional self-training methods use manually designed heuristics to select data, so manual adjustment of selection strategy is costly (Chen et al., 2018). (2) due to the severe domain shift (Fu et al., 2015), traditional self-training method may not provide reliable selection. To alleviate these problems, we present a reinforcement learning framework to learn data selection policy, which can select unlabeled data automatically and provide more reliable selection.

The contributions of our work can be summarized as follows:

- We propose a self-training based method to leverage unlabeled data in zero-shot text classification. Our method is capable of alleviating the domain shift problem and enabling transferring between classes sharing low similarities and connections.

- We propose a reinforcement learning framework to learn data selection policy automatically instead of using manually designed heuristics.

- Experimental results on both benchmarks and a real-world e-commerce dataset show that our method outperforms previous methods with a large margin of 15.4% and 5.4% on average in generalized and non-generalized ZSL respectively.

2 Related Work

2.1 Zero-shot Learning

Zero-shot learning has been widely studied in image classification, in which training classes and testing classes are disjoint (Lampert et al., 2013; Larochelle et al., 2008; Rohrbach et al., 2011). The general idea of zero-shot learning is to transfer knowledge from seen classes to unseen classes (Wang et al., 2019). Most methods focus on learning a matching model between image feature space and class semantic space, such as visual attributes (Lampert et al., 2009), word embeddings of class names (Socher et al., 2013), class hierarchy (Socher et al., 2013).

For zero-shot text classification, similar methods have been adopted. (Dauphin et al., 2013) associated text with class label through semantic space, which is learned by deep neural networks trained on large amounts of search engine query log data. (Nam et al., 2016) proposed an approach to embed text and label into joint space while sharing word representations between text and label. (Pushp and Srivastava, 2017) proposed three neural networks to learn the relationship between text and tags, which are trained on a large text corpus. (Rios and Kavuluru, 2018) incorporated word embeddings and hierarchical class structure using GCN (Kipf and Welling, 2016) for multi-label zero-shot medical records classification. (Zhang et al., 2019) proposed a two-phase framework together with data augmentation and feature augmentation, in which four kinds of semantic knowledge (word embeddings, class descriptions, class hierarchy, and knowledge graph) were incorporated.

These works benefit from large training corpus and external semantic knowledge, however, none of these works have tried to leverage unlabeled unseen data in zero-shot text classification, namely transductive zero-shot learning (Xian et al., 2018). There exists some work to utilize unlabeled data in image classification to alleviate domain shift problem, including (Fu et al., 2012; Rohrbach et al., 2013; Li et al., 2015; Fu et al., 2015), etc. As far as we know, our work is the first to explore transductive zero-shot learning in text classification.

2.2 Self-training

Self-training is a widely used algorithm in semi-supervised learning (Triguero et al., 2015). The basic process of self-training is to iteratively select high-confidence data from unlabeled data and add these pseudo-labeled data to training set. Self-training has shown its effectiveness for various natural language processing tasks, including text classification (Drury et al., 2011; Van Asch and Daelemans, 2016), name entity recognition (Kozareva et al., 2005), parsing (McClosky et al., 2006, 2008; Huang and Harper, 2009). However, there are two main drawbacks of self-training. Firstly, its data selection strategy is simply confidence-based, which may not provide reliable selection (Chen et al., 2011) and cause error accumulation. Secondly, self-training relies on pre-defined confidence threshold which varies among datasets and manual adjustment is costly.

2.3 Reinforcement Learning for Data Selection

There have been some works applying reinforcement learning to data selection in semi-supervised learning, including active learning (Fang et al., 2017), self-training (Chen et al., 2018), co-training (Wu et al., 2018). These works share a similar framework which uses deep Q-Network (Mnih et al., 2015) to learn a data selection strategy guided by performance change of model. This process is time-consuming as the reward is immediate which means the classifier is retrained and evaluated after each instance is selected. Reinforcement learning has also been applied in relation extraction to alleviate the noisy label problem caused by distant supervision. (Feng et al., 2018; Qin et al., 2018) proposed a policy network to automatically identify wrongly-labeled instances in training set. Earlier, (Fan et al., 2017) proposed an adaptive data selection strategy, enabling to dynamically choose different data at different training stages.

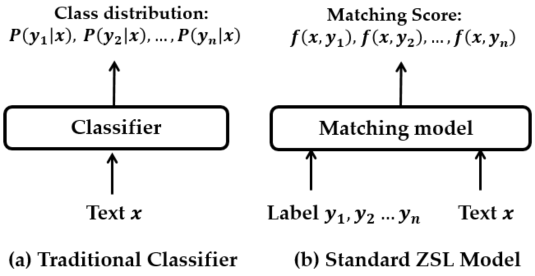

Class distribution:
$P(y_1|x), P(y_2|x), \ldots, P(y_n|x)$
Classifier
Text x
Matching Score:
$f(x, y_1), f(x, y_2), \ldots, f(x, y_n)$
Matching model
Label $y_1, y_2 \ldots y_n$
Text x
(a) Traditional Classifier
(b) Standard ZSL Model

Figure 1: Illustration of the traditional classifier and standard ZSL model.

3 Methodology

3.1 Problem Formulation and Overview

Here we first formalize the zero-shot text classification problem. Let $\mathcal{Y}^s$ and $\mathcal{Y}^u$ denote seen and unseen class set respectively, where $\mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset, \mathcal{Y}^s \cup \mathcal{Y}^u = \mathcal{Y}$. Suppose there is $\mathcal{D}^s = \{(x_i^s, y_i^s)\}_{i=1}^N$ for seen classes and $\mathcal{D}^u = \{x_i^u, y_i^u\}_{i=1}^M$ for unseen classes, where x_i represents i-th text and y_i represents the corresponding label. As shown in Figure 1, ZSL method turns a classification problem into a matching problem between text and class label. During training, we learn a matching model $f(x, y; \theta)$ from seen classes $\mathcal{D}^s$ and then make predictions on unseen classes:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} f(x, y; \theta) \ , \tag{1}$$

where θ refers to the parameter of f. For transductive ZSL, both labeled seen data $\mathcal{D}^s$ and unlabeled unseen data $\mathcal{D}^u = \{x_i^u\}_{i=1}^M$ are available during training.

To tackle zero-shot text classification, a reinforced self-training framework is developed in this work. Figure 2 shows an overview of our reinforced self-training framework for zero-shot text classification. The goal of our framework is to select high quality data from unseen classes automatically by agent and use these data to augment the performance of the base matching model. Specifically, we first train the base matching model on seen class data and make predictions on unseen class data. To make it more efficient, the agent performs data selection from a subset of unlabeled data instead of all unlabeled data at each iteration. We rank the instances by prediction confidence and take a certain ratio of instances from

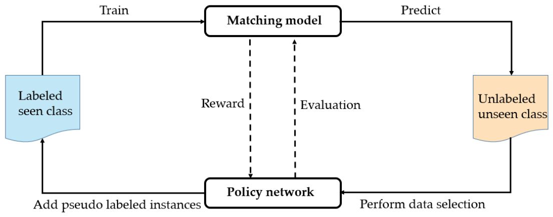


Figure 2: Overview of our reinforced self-training framework for zero-shot text classification.

it at each iteration. The agent is responsible for selecting data from this subset and filter negative instances. The reward is determined by the performance of matching model in validation set. We will introduce the details of our method in the following subsections.

3.2 The Base Matching Model

Our RL-based data selection framework is model-agnostic, which means any matching model is compatible. Here we adopt the widely recognized pre-trained model BERT (Devlin et al., 2018) as the base matching model. For seen classes, given text x and label y, we generate $\{(x, y')|y' \in \mathcal{Y}^s\}$ as training instances, in which (x, y') is a positive training instance if $y' = y$. We take the text as premise and transform the label into its corresponding hypothesis provided in (Yin et al., 2019). Therefore, the input sequence of BERT is packed as "[CLS] x [SEP] hypotheis of y' [SEP]", where [CLS] and [SEP] are special start and separator tokens, as shown in Figure 3. BERT encoder is composed of multi-layer bidirectional transformers (Vaswani et al., 2017). We use the hidden vector $c_{x,y'} \in \mathbb{R}^H$ corresponding to [CLS] in the final layer as the aggregate representation. We add a linear layer and compute loss as below:

$$p_{x,y'} = \sigma(W^T c_{x,y'} + b), \tag{2}$$

$$\mathcal{L} = \left\{ \begin{array}{ll} -log(p_{x,y'}) & y' = y \\ -log(1 - p_{x,y'}) & y' \neq y \end{array} \right. , \tag{3}$$

where W and b are parameters of the linear layer, $W \in \mathbb{R}^H$, $b \in \mathbb{R}$, H is the hidden dimension size, and $p_{x,y'}$ indicates the matching score between x and y', $\sigma(\cdot)$ is sigmoid function.

3.3 Reinforcement Learning for Self-training

The conventional self-training method simply selects data predicted with high confidence, which is confidence-based. We formalize the data selection as a sequential decision-making process and introduce a RL framework to combine confidence-based strategy and performance-driven strategy. We describe the whole process in Algorithm 1 . The details of the RL modules are described below.

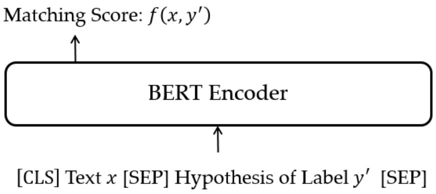

Matching Score: $f(x, y')$
BERT Encoder
[CLS] Text x [SEP] Hypothesis of Label y' [SEP]

Figure 3: BERT as the base matching model.

3.3.1 State

For each text x, we get prediction scores $\{p_{x,y'}|y' \in \mathcal{Y}^u\}$. The label y^* with maximum matching score is considered as the pseudo label. For time step t, the current state s_t consists of 2 parts: the prediction confidence p_{x,y^*}, the representation of arriving instance c_{x,y^*}. We take the hidden vector corresponding to [CLS] as the representation of current instance (x, y^*). The policy network takes p_{x,y^*} and c_{x,y^*} as input and outputs the probability whether to select or not.

3.3.2 Action

At each step, the agent is required to take action for the current instance(x, y^*) – whether to select it or not. At time step t, $a_t = 1$ means the agent accepts the current instance and adds it to training set; $a_t = 0$ means rejection. The action value is obtained through sampling from the policy network's output $P(a|s_t)$.

3.3.3 Reward

If wrongly-labeled instances are added into training set, it will degrade the performance of the matching model. Therefore the function of reward is to guide the agent to select the instances that are consistent with training set. The reward is determined by the performance of the matching model on validation set, which consists of 2 parts: seen validation set $\mathcal{D}^s_{dev}$ and unseen validation set $\mathcal{D}^u_{dev}$. $\mathcal{D}^u_{dev}$ comes from the pseudo labeled data, which guides newly-selected data to be consistent with previously-selected data. More specifically, after each batch of selection, we train the matching model using the selected instances, and evaluate on validation set. We use macro-F1 as the evaluation metric. Assume there are N_3 batches in one episode, we get two F sequences $F^s = \{F_1^s, F_2^s, ..., F_{N_3}^s\}$ for seen validation set and $F^u = \{F_1^u, F_2^u, ..., F_{N_3}^u\}$ for unseen validation set. For batch k, the reward is formulated as:

$$r_k = \frac{(F_k^s - \mu^s)}{\sigma^s} + \lambda \cdot \frac{(F_k^u - \mu^u)}{\sigma^u} \,, \qquad (4)$$

where λ controls the weight of seen class and unseen class, μ and σ represent the mean and standard deviation of F, respectively.

3.3.4 Policy Network

We adopt a multi-layer perceptron (MLP) as the policy network. The policy network receives states: the prediction confidence p_{x,y^*} and the representation of arriving instance c_{x,y^*}, then output the probability for each action.

$$z_t = ReLU(W_1^T c_{x,y^*} + W_2^T p_{x,y^*} + b_1), \qquad (5)$$

$$P(a|s_t) = softmax(W_3^T z_t + b_2) \,. \qquad (6)$$

We use ReLU as the activation function, W_1, W_2, W_3, b_1, b_2 are the parameters of MLP, and $P(a|s_t)$ is the probability of actions.

3.3.5 Optimization

To learn an optimal data selection policy, we aim to maximize the expected total reward, which can be formulated as:

$$J(\phi) = E_{P_\phi(a|s)}[R(s,a)] \,, \qquad (7)$$

where $R(s,a)$ is the state-action value function and ϕ is the parameter of policy network. We update the ϕ via policy gradient (Sutton et al., 2000),

$$\phi \leftarrow \phi + \eta \nabla_\phi \tilde{J}(\phi) \,, \qquad (8)$$

where η is the discount learning rate. For a batch B_k, we sample an action a_t for each state s_t according to policy $P_\phi(a|s)$. After one episode , we compute rewards $\{r_k\}_{k=1}^{N_3}$ by Equation 4. The gradient can be approximated by

$$\nabla_\phi \tilde{J}(\phi) = \frac{r_k}{|B_k|} \sum_{t=1}^{|B_k|} \nabla_\phi logP(a_t|s_t) \,, \qquad (9)$$

where $|B_k|$ is the number of instances in one batch, r_k is the reward of batch B_k, the parameter of policy network is updated after each episode.

Algorithm 1 Reinforced self-training for zero-shot text classification

```
Require: labeled seen data $\mathcal{D}^s = \{(x_i^s, y_i^s)\}_{i=1}^N$,
    unlabeled unseen data $\mathcal{D}^u = \{(x_i^u)\}_{i=1}^M$, seen
    validation set $\mathcal{D}_{dev}^s$.
Initialize pseudo-labeled data $\mathcal{D}^p \leftarrow \emptyset$
for $i = 1 \rightarrow N_1$ do    //iteration $i$
    Train matching model $f$ with instances
    from $\mathcal{D}^s$ and $\mathcal{D}^p$.
    Make prediction on $\mathcal{D}^u$, get confidence $P$.
    Get a subset $\Omega$ from $\mathcal{D}^u$ by ranked confi-
    dence $P$.
    for $j = 1 \rightarrow N_2$ do    //episode $j$
        if early stop criteria is met then
            break
        end if
        Shuffle $\Omega = \{B_1, B_2, ..., B_{N_3}\}$.
        for $k = 1 \rightarrow N_3$ do    //batch $k$
            Get a batch $B_k$ from $\Omega$.
            Decide action for each instance in
            $B_k$, get selected instances $B_k^p$.
            Train model $f'$ with $B_k^p$.
            Evaluate on $\mathcal{D}_{dev}^s$ and $\mathcal{D}_{dev}^u$,
            get $F_k^s$, $F_k^u$.
        end for
        Compute rewards $\{r_k\}_{k=1}^{N_3}$ by equa-
        tion 4.
        // update policy network
        for $k = 1 \rightarrow N_3$ do
            $\phi \leftarrow \phi + \eta \frac{r_k}{|B_k|} \sum_{t=1}^{|B_k|} \nabla_\phi logP(a_t|s_t)$
        end for
    end for
    $\mathcal{D}_i^p \leftarrow \cup_{k=1}^{N_3} B_k^p$
    $\mathcal{D}^p \leftarrow \mathcal{D}^p \cup \mathcal{D}_i^p$
    $\mathcal{D}^u \leftarrow \mathcal{D}^u \setminus \mathcal{D}_i^p$
    $\mathcal{D}_{dev}^u \leftarrow \mathcal{D}^p$.
end for
```

4 Experiments

4.1 Datasets

We use two kinds of datasets for our experiments. The first comes from the recently released benchmarks for zero-shot text classification (Yin et al., 2019), including 3 datasets: topic, emotion and situation classification. Considering that some texts in situation dataset has multiple labels, we remove texts with multiple labels and keep single-label texts. To keep consistent with Equation 1, "none" type is not included in unseen classes. Datasets are prepared with two versions of partitions with non-

		Seen class		Unseen class
		#Train	#Valid	#Test
Topic	I	650000	5000	50000
	II	650000	5000	50000
Emotion	I	20465	2405	5101
	II	14204	1419	8901
Situation	I	2428	240	689
	II	1747	173	1102
E-commerce	I	9000	1000	5000
	II	9000	1000	5000

Table 2: Statistics of text classification Datasets, where I and II refer to two ways of partitions respectively described in (Yin et al., 2019).

overlapping labels so as to get rid of the models over-fitting on one of them.

To further evaluate our method in real-world scenario, we construct a new dataset from e-commerce platform, where texts consist of user search queries. For seen classes $\mathcal{Y}^s$, it consists of the categories of product that users click on after searching. For unseen classes $\mathcal{Y}^u$, it consists of the pre-defined user preference classes. User preference refers to the product's attribute that users prefer, such as the efficacy of cosmetic products, the style of furniture. The user preference and product category are disjoint so it can be formalized as a zero-shot learning problem. We annotate 10-class user preference dataset for evaluation and there is 1000 instances for each class. Following (Yin et al., 2019), we created two versions of unseen classes each with 5 classes that do not overlap. The statistics of datasets are shown in Table 2.

4.2 Implementation Details

We use the BERT-Base (Devlin et al., 2018) as our base matching model, with 12-layer transformer blocks, 768-dimension hidden state, 12 attention heads and total 110M parameters. We use the pre-trained BERT-Base-Uncased[*] for the English benchmarks and BERT-Base-Chinese[†] for e-commerce dataset. For training stage, we use Adam (Kingma and Ba, 2014) for fine-tuning with β_1 as 0.9, β_2 as 0.999. The max sequence length of BERT input is set to 64. For other hyper-parameters, we set learning rate as 5e-5, ratio $\delta = size(\Omega)/M$ as 0.2, iteration number N_1 as 5 and episode number N_2 as 20. We select weight λ among $\{1, 2, 5, 10\}$. For baselines, we adopt 300-dim GloVe vectors (Pennington et al., 2014) for English words and 300-dim word vectors from (Li et al., 2018) for Chinese words.

Policy network pre-train is widely used by reinforcement learning based methods to accelerate the training of RL agent (Silver et al., 2016; Xiong et al., 2017; Qin et al., 2018). We use seen class data to pre-train the agent, enabling the agent to distinguish negative instances. We set early stop criteria to avoid overfitting to seen class data.

4.3 Baseline Methods

We compare our method with the following baselines: (1) **Word2vec** measures how well a label matches the text by computing cosine similarity of their representations. Both the representations of text and labels are average of word embeddings. (2) **Label similarity** (Veeranna et al.) uses word embeddings to compute semantic similarity as well, which computes the cosine similarity between class label and every n-gram (n=1,2,3) of the text, and takes the max similarity as final matching score; (3) **FC** and **RNN+FC** refers to the architecture 1 and architecture 2 proposed in (Pushp and Srivastava, 2017).

We also compare multiple variants of our models: (1) **BERT** refers to the base matching model without self-training and RL; (2) **BERT+self-training** refers to the traditional self-training method, which selects instances with high confidence. However, confidence threshold has great impact on performance. With different thresholds, the number of selected instances differs, resulting in performance change of the model. To provide a fair comparison, we record the number of instances k selected in every iteration in RL selection process. For self-training, we select top k instances for every iteration. (3) **BERT+RL** refers to full model of our methods.

We use macro-F1 as evaluation metric in our experiments since datasets are not well balanced. We report the results in two ZSL setting: generalized and non-generalized. In non-generalized ZSL, at test time we aim to assign an instance to unseen class label ($\mathcal{Y}^u$). While in generalized ZSL, class label comes from both unseen and seen classes ($\mathcal{Y}^s \cup \mathcal{Y}^u$). The harsh policy in testing (Yin et al., 2019) is not adopted in our experiments.

[*] https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip

[†] https://storage.googleapis.com/bert_models/2018_11_03/chinese_L-12_H-768_A-12.zip

	Topic		Emotion		Situation		E-commerce	
	I	II	I	II	I	II	I	II
Word2vec	35.50	35.33	4.77	11.45	40.67	36.33	53.09	55.47
Label similarity	34.62	36.14	10.63	16.89	54.56	37.45	59.04	55.89
FC	19.45	22.46	27.36	8.31	24.33	25.01	26.40	22.45
RNN+FC	9.68	13.41	15.45	3.15	15.58	14.09	25.76	18.15
BERT	57.07	45.50	16.86	10.21	60.23	34.15	58.05	66.47
BERT+self-training	72.21	62.90	31.96	**19.72**	69.00	49.30	65.14	76.72
BERT+RL	**73.41**	**65.53**	**36.98**	19.38	**73.14**	**52.44**	**70.63**	**80.32**

Table 3: Generalized experimental results on benchmarks and real-world e-commerce dataset, where I and II refer to two versions of partitions respectively.

	Topic		Emotion		Situation		E-commerce	
	I	II	I	II	I	II	I	II
Word2vec	38.16	49.08	18.42	12.17	59.02	37.89	59.52	70.17
Label similarity	39.36	45.70	27.43	17.81	67.73	39.96	61.90	72.73
FC	20.93	29.29	33.76	12.98	38.47	34.15	34.10	30.57
RNN+FC	31.09	28.63	33.05	19.47	32.98	25.61	32.44	26.52
BERT	67.73	60.20	29.31	11.96	75.08	51.48	70.77	79.74
BERT+self-training	73.24	**67.97**	33.71	**20.76**	76.03	53.18	73.95	82.74
BERT+RL	**74.46**	66.70	**37.33**	20.57	**77.23**	**53.63**	**75.58**	**83.97**

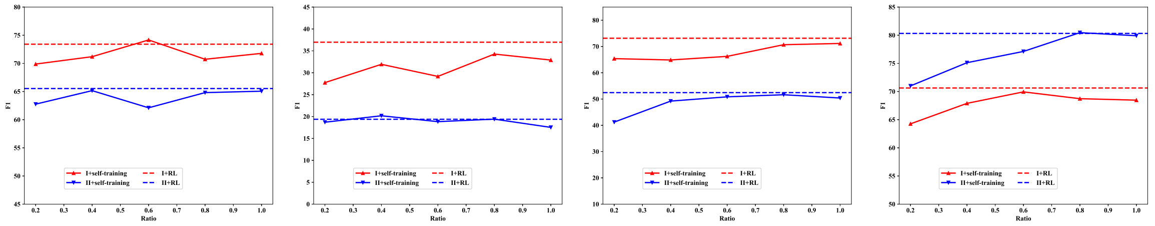
Table 4: Non-generalized experimental results on benchmarks and real-world e-commerce dataset, where I and II refer to two versions of partitions respectively.

4.4 Results

Table 3 shows the experimental results on benchmarks and real-world e-commerce dataset in generalized setting. For baseline methods, Word2vec and Label similarity are unsupervised approaches, which cannot get desirable results as the effectiveness of these methods heavily rely on the similarity of text and label. Therefore, it may not perform well on dataset like emotion detection. Label similarity performs slightly better than Word2vec, which proves that max aggregation of n-grams is better than mean aggregation in Word2vec method. As for the supervised FC and RNN+FC method, FC gets slightly better results than RNN+FC in most datasets. As the number of categories and the scale of training dataset are small, RNN+FC may overfit on seen class data and cannot generalize well on unseen class data.

For variants of our method, we can observe that the full model BERT+RL outperforms all other baselines. On average, BERT+RL achieves an improvement of 15.4% over BERT. To be specific, the base matching model BERT performs better than previous baselines, which shows good generalization results benefiting from pre-training on large-scale corpus. For BERT+self-training, the integration of unlabeled data augments the base matching model and shows superior performance than BERT. Last but not least, our full model BERT+RL shows substantial improvement over BERT+self-training in most datasets. Under the condition that the number of selected instances remains the same, reinforced selection strategy can still yield better performance than the simply confidence-based strategy, which proves the effectiveness of our RL policy.

For non-generalized ZSL setting, we can get similar results as presented in Table 4. On average, BERT+RL achieves an improvement of 5.4% over BERT. However, we notice that the improvement is more significant in generalized ZSL compared to non-generalized ZSL. The reason is that model trained on seen class data tends to bias towards seen classes, resulting in poor performance in generalized setting (Song et al., 2018). Our approach, however, could relieve the bias in favour of seen classes by incorporating pseudo-labeled unseen class data.



(a) Topic (b) Emotion (c) Situation (d) E-commerce

Figure 4: Performance with regards to selected instance ratio ϵ. One can see the RL data selection strategy does not rely on manually-set ratio and can yield consistently better performance than the competitors in most cases.

Label	BERT	BERT+RL
Joy	1. Good morning joyful people. Choose happiness to have a great day today. 2. I was filled with joy when I heard I had been selected to come here at Kamuzu College of Nursing.	1. And they all rejoiced, and embraced him and kissed him without stopping. 2. When I got a record as a gift from a friend.
Sadness	1. I'm sick and sad, missing out on Martini Lounge tonight. 2. Crossing the bridge, leaving ocean city I'm sad.	1. When I learned that two of my friends had a serious car accident. 2. Oh my god! Got in a car accident! Pray for him!
Whitening	1. Mizon Good Night White Sleeping Mask. 2. Intimate Skin Whitening Cream For Face.	1. VieBeauti Dark Spot Corrector Remover. 2. Intimate Skin lightening Cream.
Nordic style	1.Aah Nordic modern cloth sofa size living room. 2. Nordic Side Table, Modern Decoration.	1. Fabric sofa, simple and modern apartment living room. 2. Modern simple style living room chandelier.

Table 5: Qualitative comparison between BERT and BERT+RL. Left: texts predicted with high confidence; Right: texts being misclassified by BERT while being correctly labeled by BERT+RL.

4.5 Impact of Selection Ratio

When selecting the same number of instances per iteration, previous experimental results show our reinforced selection strategy can yield better performance than the greedy strategy. We define ϵ as the ratio of selected instances size to all unlabeled instances size. In this section, we vary the selection ratio ϵ among $\{0.2, 0.4, 0.6, 0.8, 1.0\}$ for self-training method. For each iteration, we select top $\frac{\epsilon}{N_1}M$ instances and add them into training set. Figure 4 shows the performances with different selection ratios in generalized ZSL setting. Clearly, the performance of self-training method varies with different ratio of instances selected. The optimal ratio of selection instances also varies with different datasets. However, our reinforced data selection strategy does not rely on manually-set ratio and can yield consistently better performance than the self-training method in most cases.

4.6 Case Study

In Table 5, we listed some examples to further reveal the differences between BERT and BERT+RL method. In the left part of the table, texts predicted by BERT with highest confidence are listed. We can easily find that these texts share a simple matching pattern that label words appear in the text, which is highlighted with red color. These simple patterns are exactly class-invariant patterns we defined previously, which can be shared among classes. In the right part of the table, we select the texts which are misclassified by BERT but are predicted correctly by BERT+RL. We can observe that those texts are harder to be distinguished since these matching patterns are more class-dependent, which cannot be directly transferred from other classes. There is no doubt that model trained on other classes would fail in such cases. For our method, we first tackle the easy instances, then add these instances into training set iteratively. With the integration of instances with easy pattern, the model can learn harder pattern gradually. In this way, our method can learn to transfer between classes even with low similarity.

5 Conclusion

In this paper, we propose a reinforced self-training framework for zero-shot text classification. To realize the transferring between classes with low similarity, our method essentially turns a zero-shot learning problem into a semi-supervised learning problem. In this way, our approach could leverage unlabeled data and alleviate the domain shift between seen classes and unseen classes. Beyond that, we use reinforcement learning to learn data selection policy automatically, thus obviating the need to manual adjustment. Experimental results on both benchmarks and real-world e-commerce dataset demonstrate the effectiveness of the integration of unlabeled data and the reinforced data selection policy.

Acknowledgments

This work is funded by NSFC U19B2027/91-846204/61473260, national key research program 2018YFB1402800, and supported by Alibaba-ZJU Frontier Technology Research Center.

References

Chenhua Chen, Yue Zhang, and Yuze Gao. 2018. Learning how to self-learn: Enhancing self-training using neural reinforcement learning. In *2018 International Conference on Asian Language Processing (IALP)*, pages 25–30. IEEE.

Minmin Chen, Kilian Q Weinberger, and John Blitzer. 2011. Co-training for domain adaptation. In *Advances in neural information processing systems*, pages 2456–2464.

Yann N Dauphin, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2013. Zero-shot learning for semantic utterance classification. *arXiv preprint arXiv:1401.0509*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Brett Drury, Luis Torgo, and Jose Joao Almeida. 2011. Guided self training for sentiment classification. In *Proceedings of Workshop on Robust Unsupervised and Semisupervised Methods in Natural Language Processing*, pages 9–16.

Yang Fan, Fei Tian, Tao Qin, Jiang Bian, and Tie-Yan Liu. 2017. Learning what data to learn. *arXiv preprint arXiv:1702.08635*.

Meng Fang, Yuan Li, and Trevor Cohn. 2017. Learning how to active learn: A deep reinforcement learning approach. *arXiv preprint arXiv:1708.02383*.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. 2012. Attribute learning for understanding unstructured social activity. In *European Conference on Computer Vision*, pages 530–543. Springer.

Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. 2015. Transductive multi-view zero-shot learning. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2332–2345.

Zhongqiang Huang and Mary Harper. 2009. Self-training pcfg grammars with latent annotations across languages. In *Proceedings of the 2009 conference on empirical methods in natural language processing: Volume 2-Volume 2*, pages 832–841. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Zornitsa Kozareva, Boyan Bonev, and Andres Montoyo. 2005. Self-training and co-training applied to spanish named entity recognition. In *Mexican International conference on Artificial Intelligence*, pages 770–779. Springer.

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE.

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2013. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465.

Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. 2008. Zero-data learning of new tasks. In *AAAI*, volume 1, page 3.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical reasoning on chinese morphological and semantic relations. *arXiv preprint arXiv:1805.06504*.

Xin Li, Yuhong Guo, and Dale Schuurmans. 2015. Semi-supervised zero-shot classification with label representation learning. In *Proceedings of the IEEE international conference on computer vision*, pages 4211–4219.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics*, pages 152–159. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2008. When is self-training effective for parsing? In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 561–568. Association for Computational Linguistics.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016. All-in text: Learning document, label, and word representations jointly. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. 2013. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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

Pengda Qin, Weiran Xu, and William Yang Wang. 2018. Robust distant supervision relation extraction via deep reinforcement learning. *arXiv preprint arXiv:1805.09927*.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2018, page 3132. NIH Public Access.

Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. 2013. Transfer learning in a transductive setting. In *Advances in neural information processing systems*, pages 46–54.

Marcus Rohrbach, Michael Stark, and Bernt Schiele. 2011. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR 2011*, pages 1641–1648. IEEE.

Kenji Sagae. 2010. Self-training without reranking for parser domain adaptation and its impact on semantic role labeling. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 37–44. Association for Computational Linguistics.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943.

Jie Song, Chengchao Shen, Yezhou Yang, Yang Liu, and Mingli Song. 2018. Transductive unbiased embedding for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1024–1033.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063.

Isaac Triguero, Salvador García, and Francisco Herrera. 2015. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information systems*, 42(2):245–284.

Vincent Van Asch and Walter Daelemans. 2016. Predicting the effectiveness of self-training: Application to sentiment classification. *arXiv preprint arXiv:1601.03288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Sappadla Prateek Veeranna, Jinseok Nam, Eneldo Loza Mencıa, and Johannes Fürnkranz. Using semantic similarity for multi-label zero-shot classification of text documents.

Wei Wang, Vincent W Zheng, Han Yu, and Chunyan Miao. 2019. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):13.

Jiawei Wu, Lei Li, and William Yang Wang. 2018. Reinforced co-training. *arXiv preprint arXiv:1804.06035*.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. 2018. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. *arXiv preprint arXiv:1707.06690*.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*.

Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. 2019. Integrating semantic knowledge to tackle zero-shot text classification. *arXiv preprint arXiv:1903.12626*.