# Semantic Space-Based Self-Training for Semi-Supervised Multi-label Text Classification

Zhewei XU † and Mizuho IWAIHARA ‡

Graduate School of Information, Production and Systems, Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, Fukuoka, 808-0135 Japan

E-mail: † xuzhewei@toki.waseda.jp, ‡ iwaihara@waseda.jp

**Abstract** Finetuning a pre-trained language model is recognized as an effective method for text classification. Since the learning rate in the finetuning stage is often small, we think that classification accuracy depends largely on the quality of the pre-trained model. Based on such an assumption, we design a self-training method that better adapt to the popular two stage training pattern for multi-label text classification under a semi-supervised scenario. We enhance the quality of the pre-trained model by continuously finetuning the semantic space toward increasing high-confidence predictions, intending to further promote the performance on target tasks. We also introduce a comprehensive confidence score and dynamic confidence threshold in selecting high-confidence predictions. Finally, stop criteria are newly designed to prevent the classifier from being poisoned by noise. Our experimental results confirm the effectiveness of our method and show a significant improvement over the baseline methods.

**Keyword** Semi-supervised learning, Multi-label classification, Self-training, Semantic space finetuning

## 1. Introduction

Text classification occupies an important role in natural language processing and information retrieval. It is widely used in the fields like document management, news categorization and sentiment analysis. However, lack of labeled samples is still a challenge that cannot be ignored.

Although the emergence of pre-trained language models caused a great sensation in the field of natural language processing, these pre-trained models can hardly achieve satisfactory results with insufficient labeled data.

To deal with the shortage of labeled data, researchers use not only labeled data, but also unlabeled data for model training, which is known as semi-supervised learning.

Self-training and co-training are two symbolic semi-supervised learning approaches. Both have been proved effective for semi-supervised learning through plenty of researches [5][16][12][13]. Generally, co-training involves two classifiers and it requires sufficient and independent assumptions to ensure that the two classifiers effectively promote each other. In comparison, self-training works without any assumption, making it versatile to various tasks.

Although self-training has achieved a great success, most of existing researches only apply self-training on multi-class classification, in which each sample falls into exactly one category. In reality, many instances like Wikipedia articles, patents and academic papers could be classified into multiple categories. While only a small number of papers apply self-training to these multi-label classifications [1][18].

In this paper, we adapt the self-training structure to multi-label scenario and propose a new approach named Semantic Space-Based Self-training for Semi-Supervised Multi-label Text Classification (S3-MTC). It is a fusion of self-training and popular two-stage (pre-training and finetuning) learning pattern, in which the potential of the pre-trained model is fully explored. Specifically, we finetune the semantic space for the pre-trained model and then, add a linear layer on the top to finetune the classifier. After that, confidence predictions are picked out for the semantic space finetuning in the next round. By this way, the semantic space of the pre-trained model is going to be more and more adaptive to the target dataset, so that the classification performance is continuously improved with each iteration. Besides, we propose a strategy for confidence evaluation and high-confidence prediction selection for minimizing the introduction of noise. Also, stop criteria are designed to enable the system to stop training in time when it senses too much noise, preventing the training process from falling into a vicious circle.

The rest of the paper is organized as follows. Section 2 reviews the related work in the fields of self-training and semi-supervised multi-label learning. Section 3 formulates the target task and introduces the motivation about the proposed approach. Section 4 explains each process of the proposed method in detail. Section 5 introduces the

experiments and results. Finally, Section 6 summarizes our work and draws conclusion.

## 2. Related work

**Self-training.** The original idea of self-training can be traced back to 1965 [7]. Since then, it is constantly being extended and improved by researchers. Until now, excellent works based on self-training are still emerging. One of the notable works is from Lee [4], in which he brought pseudo labels to self-training and achieved great improvement. Meng et al. [17] integrated self-training to their model to deal with text classification problems and outperformed their baselines significantly. Xie et al. [12] proposed a noisy student method based on self-training for image classification and achieved a new state-of-the-art on ImageNet. Li et al. [15] adapted self-training for few-shot scenario and enabled the model to complete self-training without intervention.

**Semi-supervised multi-label classification.** There have been several trials for multi-label classification under semi-supervised scenario. Chen et al. [6] and Wu et al. [9] dealt with this task by solving a sylvester equation and by imposing maximum-margin regularization over unlabeled data respectively. Both of their methods achieved good performance through experiments. Kong et al. [14] held the point that each multi-label instance has a label concept composition, based on this idea they transformed the classification problem into an optimization problem of estimating the label composition for each unlabeled instance. Zhan et al. [13] applied co-training on semi-supervised multi-label classification and validated that their model performs favorably against state-of-the-art multi-label learning approaches.

## 3. Preliminaries

### 3.1 Problem formulation

Given a collection of $n$ documents $X = \{x_1, x_2, \dots, x_l, x_{l+1}, \dots, x_n\}$ and a collection of $m$ categories $C = \{c_1, c_2, \dots, c_m\}$, we assume that each document $x$ is associated with a vector of labels $\boldsymbol{y} = [y^1, y^2, \dots, y^m], y^i \in \{0,1\}$, where $y^i = 1$ indicates the document belongs to $c_i$, otherwise $y^i = 0$. In the given document collection $X$, the labels of the first $l$ documents $x_1, x_2, \dots, x_l$ are known to us. We denote this labeled document set as $X_{label}$. The labels of the remaining $l - n$ documents $x_{l+1}, x_{l+2}, \dots, x_n$ are unknown to us. We denote this unlabeled document set as $X_{unlabel}$. Our target task is to predict the labels of the unlabeled document set $X_{unlabel}$ by utilizing all the information provided by the document collection $X$.

### 3.2 Motivation

Although pre-trained language models like BERT [8] have achieved great successes on various natural language processing tasks, when we focus on a single task, pre-trained language models that contain general text information have still not met our increasing expectations.

Researchers began to explore how to make pre-trained models better adapt to target tasks to maximize their capabilities. Sun et al. [3] tried further pre-training BERT with the original two tasks: Next sentence prediction and mask prediction, and drew the conclusion that further pre-training on the task-specific dataset effectively boosts the performance on target classification tasks. While in [11], the embedding extracted from the SBERT model, which is a pre-trained model that further finetunes BERT with the aim to produce semantically meaningful sentence embeddings, are proved outperforming embedding from other approaches, including BERT embedding, on multiple classification tasks.

Essentially speaking, the above works finetuned the semantic space of the general pre-trained language model and indeed improved the effect on the target classification tasks. It can be inferred that the more sufficient the semantic space is finetuned, the better the performance on target tasks will be.

We expect that as more high-confidence predictions are discovered and entered the self-training structure, more task-specific samples can be used for semantic space finetuning, contributing to further classifier refinement.

## 4. Proposed approach

Figure 1 shows the overview of our proposed Semantic Space based Self-training for semi-supervised Multi-label Text Classification (S3-MTC). We firstly utilize labeled documents to finetune the semantic space of the pre-trained model in Step 1, after which the system enters the cycle of semantic-space based self-training. In Step 2, we establish the classifier and finetune it on the same dataset with semantic space finetuning. Then we infer prediction scores for all the unlabeled documents, select high-confidence predictions and assign pseudo labels as shown in Steps 3 to 5. In Step 6, we combine the labeled set and the confidence set together to finetune the semantic space of the pre-trained model again. This cycle

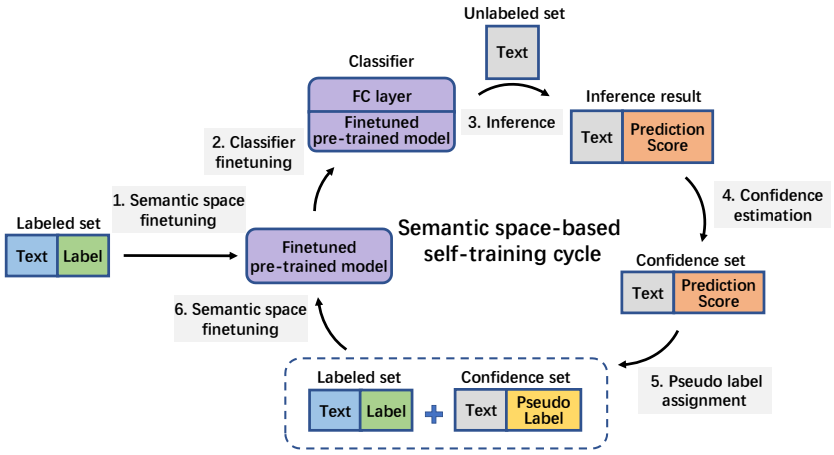
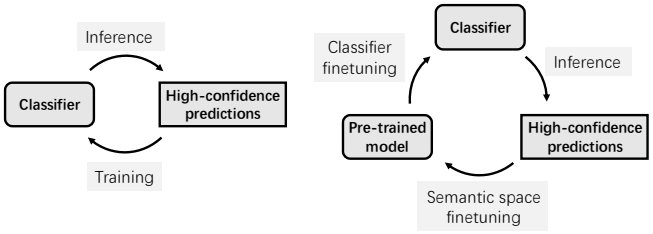



**Fig. 1. The process of the proposed S3-MTC**

iteratively runs until stop criteria are satisfied.

The difference from traditional self-training is shown in Figure 2 that self-training uses high-confidence predictions to train the classifier directly, while our method uses high confidence predictions for finetuning the semantic space, which is in turn used for cosine similarity-based search of samples for classifier finetuning. As a distinction, we name our method semantic space-based self-training.




(a)Traditional self-training (b)Semantic space-based self-training

**Fig. 2. Comparison between traditional self-training and semantic space based self-training**

## 4.1 Semantic space finetuning

Following SBERT [12], we use siamese network structure to finetune the semantic space of BERT on the target task dataset. The target is to make semantically similar documents have closer intermediate representations in the semantic space.

We extract text pairs from $X_{label} \cup X_{conf}$, to construct text pair set $S$ for semantic space finetuning. However, if pairs are formed of every two texts, $S$ will expands exponentially as the text count increases. To avoid such intractability, we randomly take $N_p$ positive samples and $N_n$ negative samples to construct text pair set $S_x$ for each $x \in (X_{label} \cup X_{conf})$:

$$S_x = \{(x, x^{p_1}), \dots, (x, x^{p_{N_p}}), (x, x^{n_1}), \dots, (x, x^{n_{N_n}})\}$$
$$x, x^{p_i}, x^{n_j} \in (X_{label} \cup X_{conf}), i = 1,2, \dots, N_p, j = 1,2, \dots, N_n \quad (1)$$

Here, positive sample $x^{p_i}$ has at least one positive label consistent with $x$, while negtive sample $x^{n_i}$ has no consistent positive label with $x$.

The text pair set $S$ is a combination of all the constructed text pairs:

$$S = \bigcup_{x \in (X_{label} \cup X_{conf})} S_x \quad (2)$$

For a pair of text $(x_v, x_w)$, we intend to train the semantic space so that the similarity of two text representations $\boldsymbol{v}$ and $\boldsymbol{w}$ is positively correlated with that of label vectors $\boldsymbol{y_v}$ and $\boldsymbol{y_w}$:

$$\min \ |\cos(\boldsymbol{v}, \boldsymbol{w}) - \cos(\boldsymbol{y_v}, \boldsymbol{y_w})| \quad (3)$$

So that the loss function of the semantic space finetuning is:

$$\text{Loss} = \sum_{(\boldsymbol{v},\boldsymbol{w}) \in S} |\cos(\boldsymbol{v}, \boldsymbol{w}) - \cos(\boldsymbol{y_v}, \boldsymbol{y_w})| \quad (4)$$

## 4.2 Classifier finetuning

As shown in Figure 3, we construct the multi-label classifier by adding a fully connected layer on top of the pre-trained model finetuned by the semantic space. Then, sigmoid function is used to produce the results $\boldsymbol{ps_u} \in \boldsymbol{R}^m$:

$$\boldsymbol{ps_u} = \boldsymbol{\sigma}(\boldsymbol{M}\boldsymbol{u}^T) \quad (5)$$

Here, $\boldsymbol{u} \in \boldsymbol{R}^d$ is the representation of document $x_u$, $\boldsymbol{\sigma}()$ is an element-wise sigmoid function of dimension $m$, $\boldsymbol{M} \in \boldsymbol{R}^{m \times d}$ is a trainable matrix intended to capture the label correlations.

The multi-label classifier is finetuned with the average binary cross entropy loss function:

$$Loss = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{l} \left(-y_j^i \log ps_j^i + \left(1 - y_j^i\right) \log\left(1 - ps_j^i\right)\right) \quad (6)$$

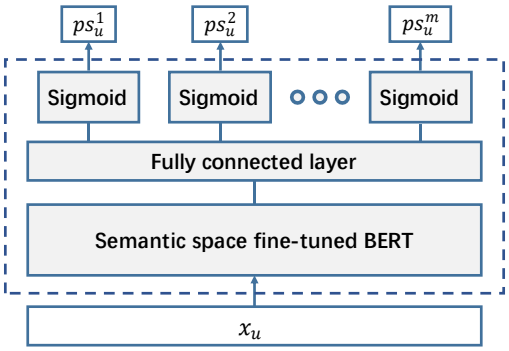



**Fig. 3. The structure of the multi-label classifier**

### 4.3 Comprehensive confidence estimation

For each unlabeled document, we calculate *confidence score* $CS^i$ on each category to estimate whether a prediction result is trustable or not.

We expect that the confidence score reflects the information not only from the classifier but also from neighbors in the semantic space. So we introduce $CS_{ps}^i$ and $CS_{nbr}^i$ to estimate confidences considering these two factors.

For an unlabeled sample $x_u$, $CS_{ps}^i$ represents how confident the classifier is on the prediction result of the $i$-th category, which is defined as:

$$CS_{ps}^i = ps_u^i - 0.5 \tag{7}$$

$CS_{nbr}^i$ is based on the assumption that semantically similar text pairs should have similar labels. So the prediction scores of $k$-nearest neighbors $N_u = KNN(x_u)$ are able to reflect the confidence of the unlabeled sample $x_u$. We define $CS_{nbr}^i$ as follows:

$$CS_{nbr}^i = \frac{1}{|N_u|}\sum_{v \in N_u}\left(ps_u^i - 0.5\right) \times \cos(\boldsymbol{u}, \boldsymbol{v}) \tag{8}$$

which is the weighted average of neighbors' prediction scores biased by the cosine distance with $x_u$.

We produce the confidence score on the $i$-th category by:

$$CS^i = \begin{cases} \max\{CS_{ps}^i, CS_{nbr}^i\}, & CS_{ps}^i > 0 \ and \ CS_{nbr}^i > 0 \\ \min\{CS_{ps}^i, CS_{nbr}^i\}, & CS_{ps}^i < 0 \ and \ CS_{nbr}^i < 0 \\ 0, & others \end{cases} \tag{9}$$

The higher $|CS^i|$, the more reliable the result is. The confidence score of each sample should be a comprehensive consideration for each category. Therefore, we define confidence score $CS$ as:

$$CS = \frac{1}{m}\sum_{i=1}^{m}|CS^i| \tag{10}$$

### 4.4 High-confidence prediction selection

We denote by $\varepsilon_{conf}$ the confidence threshold, such that all the unlabeled samples satisfying $CS > \varepsilon_{conf}$ are regarded as high-confidence, constituting the confidence set $X_{conf}$.

However, the performance of the classifier changes as the self-training proceeds. A fixed $\varepsilon_{conf}$ cannot always precisely select satisfactory high-confidence predictions. Therefore, we propose dynamic confidence threshold that changes in each iteration for high-confidence prediction selection.

Since the confidence score $CS$ ranges in [0,0.5], we firstly divide the range into 50 cells at 0.01 intervals. We denote the interval as $I_{[B,B+0.01)}$ with the confidence boundary $B$. We calculate $CS$ for each $x \in X_{label}$ and organize all the training data into different groups by confidence intervals. Then, we count the number of labeled samples $\#I_{[B,B+0.01)}$ and calculate hamming loss $HL_{[B,B+0.01)}$ in each interval. After that, $\varepsilon_{conf}$ is determined according to Algorithm 1.

**Algorithm 1**: $\varepsilon_{conf}$ searching

**Process**:

$\varepsilon_{conf} \leftarrow 0.49$
$B_{count\_max} \leftarrow \arg\max_B \#I_{[B,B+0.01)}$
while $\varepsilon_{conf} > B_{count\_max}$:
  if $HL_{[B-0.01,B)} < \varepsilon_{HL}$:
    $\varepsilon_{conf} \leftarrow \varepsilon_{conf} - 0.01$
  end if
end while
return $\varepsilon_{conf}$

### 4.5 Pseudo label assignment

For a confidence score $CS^i \in [-0.5,0.5]$ on the $i$-th category, closing to 0 shows its ambiguous while closing to 0.5 or -0.5 means it is more confidence about the result. Therefore, we design a mapping function to map $CS^i$ to soft pseudo label $y'^i$:

$$y'^i = \begin{cases} \max\{0, \alpha \times {CS^i}^3 + 0.5\}, & CS^i < 0 \\ \min\{1, \alpha \times {CS^i}^3 + 0.5\}, & CS^i \geq 0 \end{cases} \tag{11}$$

### 4.6 Stop criteria

In the process of selecting high-confidence predictions and assigning pseudo labels, noise can be easily introduced, leading self-training into a vicious circle. To prevent this situation, we must stop self-training before its expected performance deteriorates. To realize this, we need to find signals indicating whether the self-training is

running well or not.

For an ideal self-training process, $X_{conf}$ is expected to continue to expand so that the classifier is able to continuously benefit from it. Besides, since $X_{conf}$ is used in training, the confidence set of the $T$-th iteration $X_{conf}^{T}$ should include most of the samples in $X_{conf}^{T-1}$. Furthermore, the overlapped samples should have consistent predicted labels, or we judge that excessive noise is introduced which negatively affects the model performance.

Based on the above analysis, we introduce *confidence consistency* of the $T$-th iteration $C_{conf}^{T}$, which considers the changes of the confidence set $X_{conf}$, to indicate whether the classifier performance is deteriorating.

$$C_{conf}^{T} = \frac{\#(X_{conf}^{T-1} \cap X_{conf}^{T} \textit{ with same predicted label})}{\#X_{conf}^{T-1}} \quad (12)$$

Here, '#' refers to the sample count. The numerator is the count of the overlapping samples in $X_{conf}^{T-1}$ and $X_{conf}^{T}$ having same predicted labels. The denominator is the count of the confidence set in the last iteration.

$C_{conf}^{T}$ represents the consistency between the confidence set of the current iteration and the previous iteration. When $C_{conf}^{T} < \varepsilon_{cons}$, we judge that too much noise is introduced and stop the self-training process.

In addition to confidence consistency $C_{conf}^{T}$, the situation that the confidence set includes most of the unlabeled samples can also be regarded as a signal that the classifier is strong enough and the process can be stopped. So, we define *confidence proportion* $P_{conf}^{T}$:

$$P_{conf}^{T} = \frac{\#X_{conf}^{T}}{\#X_{unlabel}} \quad (13)$$

When $P_{conf}^{T} > \varepsilon_{prop}$ is satisfied, we stop training.

If both above two conditions are not satisfied, self-training stops when it meets the maximum iteration $T_{max}$.

Our proposed approach is summarized in Algorithm 2.

**Algorithm 2**: Semantic Space-based Self-training for semi-supervised Multi-label Text Classification (S3-MTC)

**Input**: $X_{label} = \{x_1, x_2, \dots, x_l\}$, $Y_{label} = \{\boldsymbol{y_1}, \boldsymbol{y_2}, \dots, \boldsymbol{y_l}\}$, $X_{unlabel} = \{x_{l+1}, x_{l+2}, \dots, x_n\}$

**Output**: $Y_{unlabel} = \{\boldsymbol{y_{l+1}}, \boldsymbol{y_{l+2}}, \dots, \boldsymbol{y_n}\}$

**Process**:

```
finetune semantic space using X_label
for iteration in 1, 2, ..., T_max do:
    finetuned classifier using X_label
    infer X_unlabel by eq(5)
    confidence prediction estimation by eq(7-10)
    select high-confidence predictions by CS > ε_conf
    pseudo label assignment by eq(11)
    finetune semantic space using X_label ∪ X_conf
    if C_conf < ε_cons or P_conf > ε_prop:
        break the loop
end for
infer X_unlabel and get Y_unlabel
```

# 5. Experiments

## 5.1 Dataset

We use Reuters-21578 dataset [19] to evaluate the performance of our proposed approach. Top 10 topics ("acq", "corn", "crude", "earn", "grain", "interest", "money-fx", "ship", "trade", "wheat") are picked out from ModApte split [2] of the Reuters-21578 text collection and the first 8000 samples are retained to be our experiment dataset.

## 5.2 Baselines

The performance of our proposed S3-MTC is compared against two semi-supervised learning algorithms: TRAM [14] and COINS [12].

TRAM formulates the multi-label classification as an optimization problem of estimating label concept compositions. Then TRAM derives a closed-form solution to this optimization problem and an effective algorithm is used to assign labels to the unlabeled instances.

COINS adapts co-training strategy to multi-label learning. Two classification models are generated by dichotomizing the feature space with diversity maximization, and then pairwise ranking predictions on unlabeled data is iteratively communicated for model refinement.

We also select four BERT-based models for comparison. The differences among these models are shown in Table 1. Here, the label "finetuning" means we finetune the base model on the labeled data, but use no unlabeled data. "Further pre-training" means we finetune the pre-trained model on the target dataset with the original task of the pre-trained model. For SBERT, further pre-training refers to comparing cosine similarity between documents and their labels on the target dataset. We denote further pre-trained SBERT as FP-SBERT in the following part.

## 5.3 Evaluation metrics

To evaluate from multiple aspects, we consider not only the overall accuracy but also the individual accuracy of each category. Therefore, we adopt macro-F1, micro-F1 and hamming loss to evaluate the model performance.

**Table 1 Differences among BERT-based models**

| Method | Base model | Further pre-training | Self-training | Finetune classifier in self-training | Finetune semantic space in self-training |
|---|---|---|---|---|---|
| Finetuning | BERT | × | × | × | × |
| Finetuning | SBERT | × | × | × | × |
| Finetuning | SBERT | ○ | × | × | × |
| Self-training | SBERT | ○ | ○ | ○ | × |
| S3-MTC (Ours) | SBERT | ○ | ○ | ○ | ○ |

Macro-F1 and micro-F1 are calculated based on true positives, false negatives, and false positives, while hamming loss globally evaluates the proportion of miss-classified sample-label pairs, such that a relevant label is missed or an irrelevant label is predicted.

## 5.4 Implementation details

**Composition of labeled and unlabeled document sets.** We randomly choose $s$ positive samples on each category to compose the labeled document set $X_{label}$. Suppose there are $m$ labels, then the labeled document set $X_{label}$ contains a total of $s \times m$ samples. Each category has at least $s$ positive samples in $X_{label}$. The remaining documents are naturally composed to be the unlabeled document set $X_{unlabel}$. For Reuters dataset, we vary the parameter s in the range $s = \{8, 16, 24, 40, 80\}$, so that the labeled samples account for exactly 1%, 2%, 3%, 5%, 10% of all the samples. The labeled set is divided into the training set and the validation set by the ratio of 7:1.

**Hyperparameters.** The learning rate of the semantic space finetuning and the classifier finetuning are both 2e-5. For text pair construction, we set $N_n = N_p = 50$. Semantic space is finetuned for one epoch in each iteration. Thresholds $\varepsilon_{HL}$, $\varepsilon_{cons}$, $\varepsilon_{prop}$ are 0.01, 0.9, 0.9. The maximum iteration $T_{max}$ is set to 10. The pre-trained BERT and SBERT models we used are “bert-base-uncased” and “bert-base-nli-stsb-mean-tokens”. We run the code five times and calculate the evaluation metrics for averaging. All the experiments are executed on a NVIDIA RTX 3090 GPU.

## 5.5 Results

Tables 2, 3 and 4 show macro-F1, micro-F1 and hamming loss scores among all the methods under different conditions. Higher F1 score represents better performance, while hamming loss is the opposite.

According to the results, our proposed S3-MTC outperforms the other baselines under most of the conditions.

However, when the labeled data is extremely small, the performance of BERT-based models is significantly lower than traditional machine learning methods. While with the increase of the labeled samples, BERT-based models gradually exert their advantages and achieve satisfactory results. It reveals the weakness of these BERT-based models that cannot deal with very few samples well. It can be foreseen that the initial classifier with poor performance can hardly generate enough useful information for further training, making it difficult to reach to the expected result with small labeled samples.

Focusing on the BERT-based models, directly finetuning pre-trained model shows poor performance on classification. The reason is that there is a gap between the pre-training corpus and our target corpus, which is insurmountable by simply finetuning with a small learning rate. In comparison, further pre-training before finetuning significantly boosts the performance. It proves that adjusting the semantic space to fill up the gap is an effective way to improve performance.

There is another leap when information contained in unlabeled samples are leveraged in the training process. Both two self-training-based methods have improved the classifier's prediction ability. In comparison, our semantic space-based self-training surpasses traditional self-training under all the conditions, proving the effectiveness of the semantic space finetuning in the self-training process.

**Table 2 Macro-F1 under different conditions**

| Labeled sample proportion | 1% | 2% | 3% | 5% | 10% |
|---|---|---|---|---|---|
| TRAM | 0.570 | 0.698 | 0.702 | 0.722 | 0.735 |
| COINS | 0.071 | 0.120 | 0.135 | 0.142 | 0.144 |
| Finetuning (BERT) | 0.271 | 0.562 | 0.583 | 0.658 | 0.699 |
| Finetuning (SBERT) | 0.311 | 0.541 | 0.619 | 0.642 | 0.691 |
| Finetuning (FP-SBERT) | 0.605 | 0.782 | 0.782 | 0.792 | 0.824 |
| Self-training | 0.638 | 0.781 | 0.828 | 0.829 | 0.840 |
| S3-MTC (ours) | **0.640** | **0.784** | **0.833** | **0.850** | **0.855** |

**Table 3 Micro-F1 under different conditions**

| Labeled sample proportion | 1% | 2% | 3% | 5% | 10% |
|---|---|---|---|---|---|
| TRAM | **0.776** | **0.827** | 0.835 | 0.843 | 0.853 |
| COINS | 0.072 | 0.133 | 0.153 | 0.161 | 0.168 |
| Finetuning (BERT) | 0.173 | 0.311 | 0.309 | 0.642 | 0.637 |
| Finetuning (SBERT) | 0.184 | 0.291 | 0.329 | 0.594 | 0.622 |
| Finetuning (FP-SBERT) | 0.619 | 0.767 | 0.777 | 0.786 | 0.838 |
| Self-training | 0.661 | 0.754 | 0.844 | 0.848 | 0.876 |
| S3-MTC (ours) | 0.670 | 0.779 | **0.855** | **0.869** | **0.895** |

**Table 4 Hamming loss under different conditions**

| Labeled sample proportion | 1% | 2% | 3% | 5% | 10% |
|---|---|---|---|---|---|
| TRAM | **0.047** | **0.037** | 0.035 | 0.033 | 0.031 |
| COINS | 0.077 | 0.049 | 0.036 | 0.033 | 0.026 |
| Finetuning (BERT) | 0.104 | 0.096 | 0.093 | 0.067 | 0.067 |
| Finetuning (SBERT) | 0.102 | 0.099 | 0.091 | 0.069 | 0.068 |
| Finetuning (FP-SBERT) | 0.065 | 0.044 | 0.042 | 0.040 | 0.032 |
| Self-training | 0.064 | 0.046 | 0.031 | 0.031 | 0.026 |
| S3-MTC (ours) | 0.060 | 0.042 | **0.029** | **0.028** | **0.022** |

### 5.6 Exploration on semantic space

To further confirm the impact of semantic space on classifier finetuning, we use same samples to finetune classifiers from different semantic spaces for comparison.

The original SBERT model and the semantic space finetuned SBERT model, which possess different semantic spaces, are used for experimentation. To visualize the semantic space, we first generate document embeddings by SBERT model, and reduce the dimension to two by t-SNE[10] for display.

We finetune classifiers on the same 400 labeled samples from different pre-trained language models having different semantic spaces to compare the classification performance when only the semantic space changes. Results are shown in Figure 4.

Each color represents a distinct category within a semantic space. As the boundaries between categories in the semantic space become clearer, the performance of the trained classifier is improved, which further confirms the importance of semantic space finetuning.

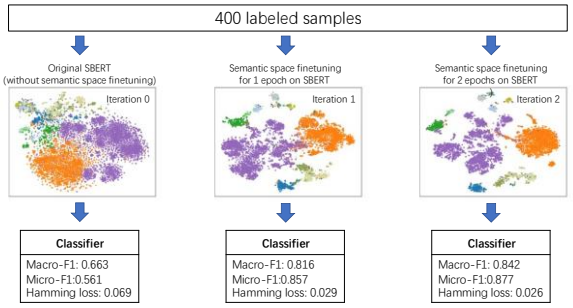



**Fig. 4. Classifier performance corresponding to different semantic spaces**

## 6. Conclusion

In this paper, we proposed a method named Semantic Space Self-training for Multi-label Text Classification (S3-MTC), which fuses self-training with prevailing two-stage learning pattern under the multi-label scenario. In the semantic space-based self-training cycle, we calculate confidence score for each unlabeled sample based on the prediction scores of the sample and its neighbors, and select high-confidence predictions according to a dynamic confidence threshold, determined with respect to the classification performance on the labeled samples. Confidence consistency is then introduced to judge whether the cycle should continue or terminate.

Through the experiments, our proposed method S3-MTC outperforms other methods under most of the conditions, which confirms its superiority. The importance of semantic space finetuning is also proved by the experimental results and the following analysis.

However, S3-MTC still has difficulty in handling extreme conditions like giving less than 10 samples for each label. Data augmentation will be one way to improve the robustness of the initial model under extreme conditions, thereby improving the overall effect of the self-learning. Besides, introducing the semantic information of the label itself may also help to solve such problems.

## References

[1] Lima, A.C. E.S., Castro, L. N.: A multi-label, semi-supervised classification approach applied to personality prediction in social media. Neural Networks 58: 122-130 (2014).

[2] Apté, C., Damerau, F., Weiss, S. M.: Towards language independent automated learning of text categorization models. In: SIGIR (1994).

[3] Sun, C., Qiu, X., Xu, Y., Huang, X.: How to finetune bert for text classification?. In: CCL (2019).

[4] Lee, D.: Pseudo-label: The simple and efficient

semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML (2013).

[5] Ma, F., Meng, D., Xie, Q., Li, Z., Dong, X.: Self-paced co-training. In: ICML (2017).

[6] Chen, G., Song, Y., Wang, F., Zhang, C.: Semi-supervised multi-label learning by solving a sylvester equation. In: SDM (2008).

[7] Scudder, H. J.: Probability of error of some adaptive pattern-recognition machines. IEEE Transactions on Information Theory 11.3: 363-371 (1965).

[8] Devlin, J., Chang, M., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019).

[9] Wu, L., Zhang, M.: Multi-label classification with unlabeled data: An inductive approach. In: ACML (2013).

[10] Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.11 (2008).

[11] Reimers, Nils, and Iryna Gurevych.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: EMNLP-IJCNLP (2019).

[12] Xie, Q., Luong, M., Hovy, E., Le, Q. V.: Self-training with noisy student improves imagenet classification. In: CVPR (2020).

[13] Zhan, W., Zhang, M.: Inductive semi-supervised multi-label learning with co-training. In: ACM SIGKDD (2017).

[14] Kong, X., Michael K., Zhou Z.: Transductive multilabel learning via label set propagation. IEEE Transactions on Knowledge and Data Engineering 25.3: 704-719 (2011).

[15] Li, X., Sun, Q., Liu, Y., Zheng, S., Zhou, Q., Chua, T., Schiele, B.: Learning to self-train for semi-supervised few-shot classification. In: NIPS (2019).

[16] Meng, Y., Shen, J., Zhang, C., Han, J.: Weakly-supervised neural text classification. In: CIKM (2018).

[17] Meng, Y., Shen, J., Zhang, C., Han, J.: Weakly-supervised hierarchical text classification. In: AAAI (2019).

[18] Wei, Z., Wang H., Zhao, R.: Semi-supervised multi-label image classification based on nearest neighbor editing. Neurocomputing 119: 462-468 (2013).

[19] University of California, Irvine. Reuters-21578 Text Categorization Collection. https://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html (1999).