SuperST: Superficial Self-Training for Few-Shot Text Classification

Ju-Hyoung Lee*, **Joonghyuk Hahn***, **Hyeon-Tae Seo, Jiho Park, Yo-Sub Han**
Department of Computer Science, Yonsei University, Seoul, Republic of Korea
dlwngud81922@gmail.com, greghahn@yonsei.ac.kr, dchs504@gmail.com,
jhpark2159@gmail.com, emmous@yonsei.ac.kr

Abstract

In few-shot text classification, self-training is a popular tool in semi-supervised learning (SSL). It relies on pseudo-labels to expand data, which has demonstrated success. However, these pseudo-labels contain potential noise and provoke a risk of underfitting the decision boundary. While the pseudo-labeled data can indeed be noisy, fully acquiring this flawed data can result in the accumulation of further noise and eventually impacting the model performance. Consequently, self-training presents a challenge: mitigating the accumulation of noise in the pseudo-labels. Confronting this challenge, we introduce superficial learning, inspired by pedagogy's focus on essential knowledge. Superficial learning in pedagogy is a learning scheme that only learns the material 'at some extent', not fully understanding the material. This approach is usually avoided in education but counter-intuitively in our context, we employ superficial learning to acquire only the necessary context from noisy data, effectively avoiding the noise. This concept serves as the foundation for SuperST, our self-training framework. SuperST applies superficial learning to the noisy data and fine-tuning to the less noisy data, creating an efficient learning cycle that prevents overfitting to the noise and spans the decision boundary effectively. Notably, SuperST improves the classifier accuracy for few-shot text classification by 18.5% at most and 8.0% in average, compared with the state-of-the-art SSL baselines. We substantiate our claim through empirical experiments and decision boundary analysis.

Keywords: Semi-supervised learning, self-training, superficial learning

1. Introduction

Self-training, a widely used semi-supervised-learning (SSL) technique, follows a two-step process: it initially trains a classifier using labeled data, then iteratively pseudo-labels unlabeled data based on the classifier's confidence scores and retrains the classifier with these pseudo-labeled samples. In the context of few-shot text classification, where labeled data is scarce, self-training has shown remarkable success (Van Engelen and Hoos, 2020; Chen et al., 2021; Cui et al., 2022). However, the approach has a fundamental challenge—it can accumulate errors during training due to the presence of noisy pseudo-labeled data (Rizve et al., 2021; Liao et al., 2022). The noise in pseudo-labeled data has a cascading effect on the classifier's performance, potentially leading the the generation of more noisy pseudo-labels. Ultimately, this cycle can degrade the overall performance of the self-training process.

Delving further into this issue, the primary concern is closely tied to the decision boundary, which delineates the boundaries between different classes (Yang et al., 2023). While it is relatively straightforward to classify data points well within the interior of each class's decision area, data located in proximity to the decision boundary poses a significant challenge for a classifier. The presence of noise in the pseudo-labels triggers this challenge, complicating the classifier to predict the data into the correct decision boundary. Further iterative process of the self-training accumulates the complication, impacting the model performance.

Simultaneously, we confront the need to incorporate these pseudo-labeled data to improve the model, particularly when only a limited amount of labeled data is at our disposal. It becomes evident that the current self-training approach has a fundamental flaw, characterized by the use of the same learning strategy for both the pristine labeled data, often referred as *golden*, and the pseudo-labeled data, which carry the potential noise. We advocate for the adoption of a distinct and appropriate learning strategy that specifically addresses the challenges associated with noisy data.

We introduce a specialized learning strategy termed *superficial learning* (Frăsineanu, 2013; Marton and Säljö, 1976), inspired by principles from pedagogy. Pedagogy typically emphasizes attaining a deep and comprehensive understanding of the subject matter. In contrast, superficial learning involves a more cursory approach, prioritizing essential knowledge over exhaustive comprehension. This approach is often discouraged in educational contexts, where instructors aim to foster a deeper understanding of the subject matter.

In our specific context, the data contains noise and potentially wrong information. Acquiring this flawed material can lead to the assimilation of erroneous knowledge, a scenario to avoid. However, due to the limited availability of labeled data, we are compelled to make use of this potentially flawed information. Instead of attempting to grasp the full context of this noisy data, we propose a strat-

*The first two authors contributed equally to this work.

LREC-COLING 2024, pages 15436–15447
20-25 May, 2024.

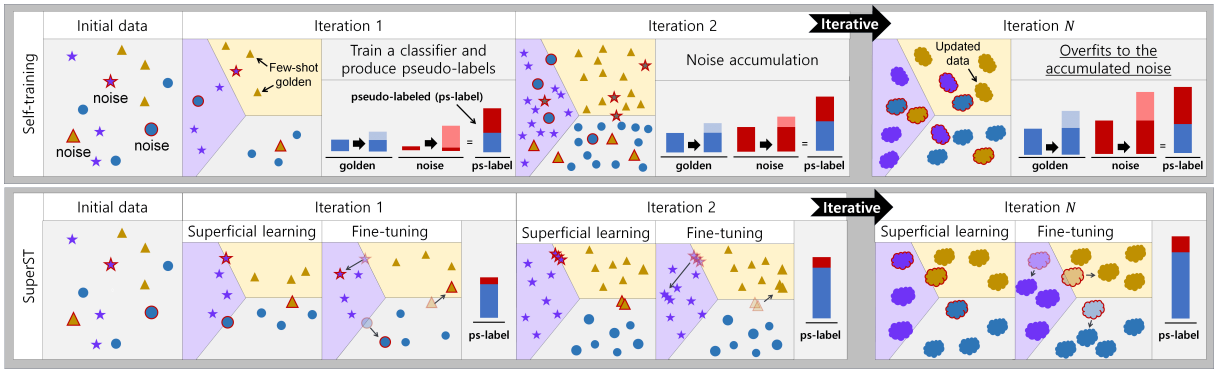

Figure 1: Demonstration of the self-training and the SuperST. While the self-training algorithm accumulates the noise, degrading the classifier in an iterative manner, SuperST confronts this problem by superficial learning. We illustrate the detailed procedure of superficial learning and fine-tuning in Figure 2.

egy that involves obtaining a partial understanding, capturing the necessary context 'to some degree'. In this way, we utilize superficial learning to train on the noisy data, avoiding a full comprehension of the incorrect information and concentrating on the required knowledge. For the pristine labeled data, we employ fine-tuning to ensure a comprehensive understanding of the correct information.

Our approach strikes a balance between essential knowledge acquisition and deep comprehension, effectively handling the nuances of noisy and golden data. We adopt superficial learning by adjusting the low learning rate and small epoch number to learn the pseudo-labeled data effectively, addressing noise and bridging the correct decision boundary. In summary, we propose a novel framework named SuperST[1], which stands for **Super**ficial **S**elf-**T**raining. SuperST leverages superficial learning for the noisy data and fine-tuning for the golden data. It is a straightforward yet highly effective approach that adjusts the appropriate learning strategy on the dependence of the noise level in the data.

SuperST's performance evaluation encompasses four widely used benchmark datasets, enabling comparisons with the previous SSL approaches. We conduct empirical analyses to assess the effectiveness of superficial learning. The experimental results demonstrate the superior performance of our approach over previous state-of-the-art methods on each dataset. Our primary contributions are as follows:

- We demonstrate that superficial learning effectively helps the model to predict the right decision boundary of noisy data through experiments and visualization.
- SuperST is a simple and robust self-training framework that requires no external resources and improves accuracy to a maximum extent of 26% from the initially trained classifier.
- We demonstrate that SuperST achieves the state-of-the-art performance for few-shot text classification spreading the gap of accuracy by at most 18.5% and 8% in average compared with the previous SSL baselines.

[1] https://github.com/HiitLee/SuperST

2. Related Work

Including self-training, SSL is widely used to improve the performance of models in various NLP tasks. It is especially effective in few-shot learning tasks as it utilizes both limited labeled data and plentiful unlabeled data in training. In this section, we present some of the successful SSL approaches for the few-shot text classification: bootstrapping learning, adversarial learning and consistency learning.

2.1. Bootstrapping Learning

Bootstrapping is a widely-used method that trains an initial classifier with labeled data and exploits the trained classifier to pseudo-label the unlabeled data and re-train a new classifier. Self-training is a traditional learning mechanism that uses high confidence prediction of the classifier to bootstrap the classifier (Yarowsky, 1995). One major drawback of bootstrapping is that it depends on the performance of the initially trained classifier. If the initially trained classifier has very low performance, then the new training sets become unreliable, and it can lead to accumulated classification errors along the training process. Several attempts have been made to alleviate the problem (Blum and Mitchell, 1998; Xiaojin, 2008; Søgaard, 2010; Jo and Cinarel, 2019). However, since these methods still depend on the performance of the initially trained classifier, it can easily still overfit to the limited labeled data.

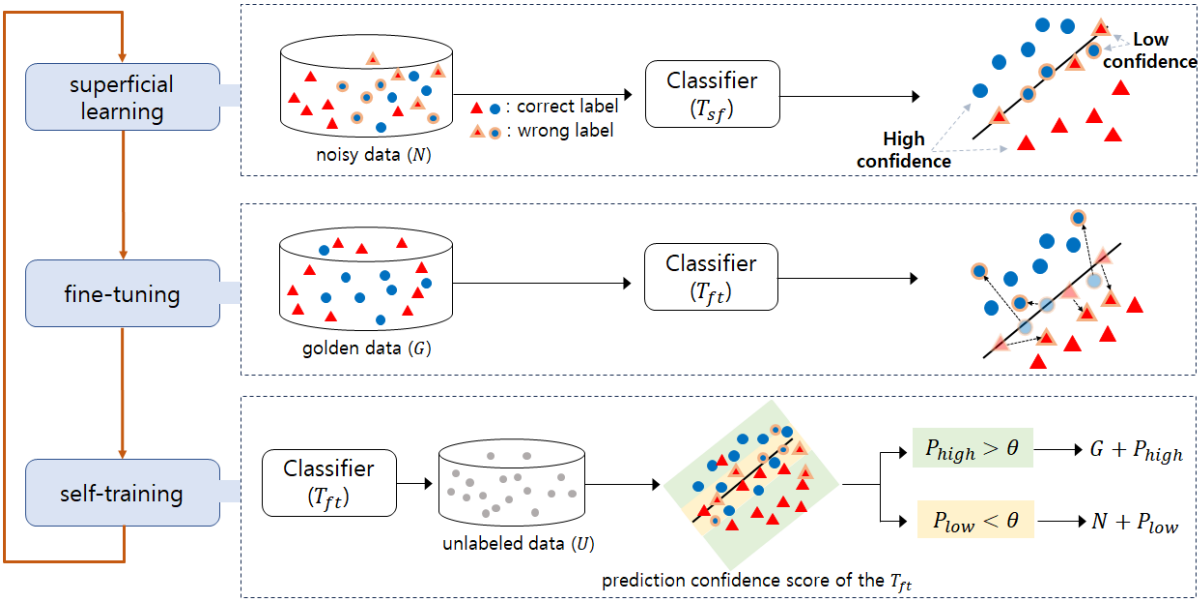

Figure 2: Our proposed method is SuperST which employs lexical-based pseudo-labeling and superficial learning to bootstrap the self-training. We set `1e-6` learning rate and 1 epoch for superficial learning and `1e-5` learning rate and 5 epochs for fine-tuning. We set 0.9 as the threshold to verify the high confidence.

The usage of lexicon and rules also shows improvements in few-shot learning. Lee et al. (2021) propose SALNet that utilizes both a lexicon for each label and the prediction confidence score of the classifier to obtain correctly-labeled data in self-training. Hahn et al. (2021) uses rules of grammar in the pseudo-label procedure to retrieve more robust pseudo-labeles. They extend their studies to data augmentation and propose GDA that leverages slot information for context-aware augmentation (Hahn et al., 2023). Kim et al. (2022) present ALP that generates augmented samples with diverse syntactic structures with plausible grammar. They bootstrap the classifier utilizing the classifier's confidence score trained with augmented data. These methods demonstrate their effectiveness in the few-shot text classification. However, these methods still are vulnerable to overfitting problems, since the performance highly relies on limited data.

2.2. Adversarial Learning

Adversarial training is a regularization technique that enhances model robustness against input perturbations. Recently, adversarial learning has been proposed for SSL. Miyato et al. (2016) applies adversarial training to adapt perturbations to the word embeddings rather than to the original input. The adversarial training improves not only classification performance but also the quality of word embeddings. Gururangan et al. (2019) propose a lightweight pretraining framework that combines a variational autoencoder to document modeling. They optimize the initial parameters of the classifier by pretraining a unigram document model as a variational autoencoder on unlabeled data. They offer a competitive lightweight alternative for pretraining from unlabeled data in the low-resource setting. However, the performances of these methods are low when labeled data is extremely limited.

2.3. Consistency Learning

Consistency learning adds noise to unlabeled data and utilizes the distribution of classifier prediction as labels to train the classifier. Data augmentation is a technique that increases the amount of labeled data without resorting to additional labeling costs. Easy Data Augmentation (EDA) (2019) is the simplest method that utilizes synonym replacement, random insertion, swap and deletions for text data augmentation. Recent SSL approaches utilize consistency training on a large number of unlabeled data and augmented data to constrain model predictions to be invariant to noise. Unsupervised Data Augmentation (UDA) (2020a) employs data augmentation to generate diverse and realistic noise and enforces the model to be consistent with respect to these noises. Chen et al. (2020) propose Mixtext that creates a large number of augmented training samples by interpolating text in hidden space. They exploit several SSL techniques to further utilize unlabeled data including self target-prediction entropy minimization and consistency regularization after back translations. However, these methods rely on the performance of data augmentation methods.

u	... i ... how god **awful** ... **crappy** ... a weapon is **hilarious** ... this movie ... **crappiness**.
k in u	positive:{hilarious} negative:{awful crappy crappiness}
l occurrence	positive: 1 negative: 3
class c of u	negative

Figure 3: An illustration of the pseudo-labeling process of unlabeled data based on keywords.

3. Proposed Methods

SuperST is a self-training framework that combines both superficial learning and fine-tuning strategies based on the level of noise in the pseudo-labeled data. on the dependence of noise that the pseudo-labeled data contains. We judge the noise of pseudo-labels by the classifier's confidence score. Using a pre-defined threshold score, we divide the pseudo-labeled data into two groups: those with low confidence scores below the threshold and those with high confidence scores above the threshold. We define *noisy data* as pseudo-labeled data with a low confidence score and *golden data* as labeled data and pseudo-labeled data with a high confidence score. The framework and learning processes of SuperST are represented in Figure 2 and Algorithm 1, respectively.

3.1. Initialization of SuperST

We produce the initial pseudo-labels using the keyword extraction algorithm. The amount of initial labeled data is very limited and if we use a classifier trained with this initial few-shot data for pseudo-labeling, pseudo-labels are likely to overfit the initial data. In order to mitigate this dependency on the classifier, we use the following keyword extraction algorithm.

The framework first produces pseudo-labeled data based on keywords extracted using rule- and model-based algorithms. From the unlabeled data, we extract the words that are representatives of the unlabeled data. The rule-based algorithm computes the distribution of the words and utilizes words that frequently occur as the keywords. The model-based algorithm computes a cosine similarity between the word embeddings and the document embeddings of the model, and extracts words with the highest similarity as keywords.

We find which class these keywords are located in and label the keywords with their corresponding classes. The labeled keywords are treated as lexicons and we pseudo-label the unlabeled data with such lexicons by counting the number of matched keywords for each label. We assign the label with the largest number of matched keywords. The following illustrates how this scheme works in more detail.

In our experiments, we have N unlabeled data $U = \{u_1, u_2, \ldots, u_N\}$, M labeled data $D = \{d_1, d_2, \ldots, d_M\}$ and a set of classes $C = \{c_1, c_2, \ldots, c_L\}$ where L denotes the number of class labels. From U, we collect a set K of keywords utilizing the rule- and model-based algorithms. We present the full details of these algorithms in Appendix A.

For each keyword $k \in K$, if k is in a labeled data $d \in D$, then we assign the class of d to k. Let C_k be the set of classes for k:

$$C_k = \{l : \text{ the label of } d \text{ that contains } k\}^2.$$

After assigning classes to all the keywords in K, we then pseudo-label the unlabeled data in U. For each $u \in U$, we check which keyword k is located in u and as multple keywords can occur in u, we assign a class that occurs the most to u. Figure 3 shows an example of a pseudo-labeled sentence.

The number of pseudo-labeled data can vary for each class and to prevent the imbalance in the number of data for each class, we select the same number of pseudo-labeled data for each class. Let this initial set of pseudo-labeled data be P. Our initialization also involves initializing the set of noisy and golden data. The initial set N of noisy data is P and G of golden data is D.

3.2. Superficial Learning

Illustrated in Section 1, superficial learning is a strategy to effectively train the noisy data, identifying the required knowledge instead of deeply learning the context of data that is potentially wrong. We implement the strategy by training the classifier T with N by 1 epoch with a low learning rate instead of employing the same hyperparameters used in T. We denote the trained T with superficial learning as T_{sf}.

3.3. Fine-Tuning

Fine-tuning is a strategy to train the golden data. We implement the strategy by training the classifier T_{sf} with G using the same hyperparameters from T.

We denote the fine-tuned classifier as T_{ft} and use T_{ft} for pseudo-labeling; for each data $u \in U$, we compute a confidence score $s_u(c)$ of T_{ft} for u's

[2]A keyword can be located in more than one data, resulting in multiple classes

Algorithm 1 A procedure of SuperST, `SuperST(U,D)`. The inputs U, D and T are the unlabeled data set, labeled data set and the baseline classifier, respectively.

function KEY-EXTRACTION(U, D)
  return a set K of keywords from U where K is extracted by rule- and model-based extraction algorithm
end function
function KEY-LABEL(U, D)
  $K \leftarrow$ KEY-EXTRACTION(U, D)
  for keyword k in K **do**
    $C_k \leftarrow \emptyset$
    for labeled data d in D **do**
      Find d that contains k
      $C_k \leftarrow$ add the label of d
    end for
  end for
  return a labeled keyword set K_L
end function
procedure SUPERST(U,D)
  $N \leftarrow \emptyset$ ▷ initialize noisy data set
  $G \leftarrow D$ ▷ initialize golden data set
  $K_L \leftarrow$ KEY-LABEL(U, D)
  for unlabeled data u in U **do**
    u_l: pseudo-labeled u by K_L
    $N \leftarrow$ add u_l ▷ initial pseudo-label
  end for
  for i in `iterationNum` **do**
    train T with N by superficial learning
    train T with G by fine-tuning
    for unlabeled data u in U **do**
      for class c in C **do**
        $s_u(c)$: T's confidence score of u for c
      end for
      $t_u \leftarrow \arg\max_{c \in C}(s_u(c))$
      if $s_u(t_u) \geq \theta$ **then**
        u_l: pseudo-labeled u by the label t_u
        $G \leftarrow$ add u_l
      else
        u_l: pseudo-labeled u by the label t_u
        $N \leftarrow$ add u_l
      end if
    end for
  end for
end procedure

prediction of each class $c \in C$. Let $t_u \in C$ denote the target class of u, where

$$t_u = \arg\max_{c \in C}(s_u(c)).$$

We pseudo-label u with t_u and depending on the s_u score, we add the data to N or G. We use a pre-defined threshold θ and generate the following sets, P_{high} and P_{low}:

$$P_{high} = \{u \mid u \in U \text{ such that } s_u(c) \geq \theta\}, \quad (1)$$
$$P_{low} = \{u \mid u \in U \text{ such that } s_u(c) < \theta\}. \quad (2)$$

3.4. Self Training

SuperST repeats the iterations of superficial learning and fine-tuning illustrated in Section 3.2 and 3.3. More specifically, the classifier trains N by superficial learning and trains G by fine-tuning. The trained classifier produces pseudo-labels which we partition into P_{high} and P_{low}. We add P_{low} to N, a set of noisy data and P_{high} to G, a set of golden data. We then repeat this iteration in a self-training manner.

$$\text{Iteration} : T \xrightarrow[N]{superficial} T_{sf} \xrightarrow[G]{fine-tune} T_{ft}, \quad (3)$$
$$\text{Update } N : N + P_{low}, \quad (4)$$
$$\text{Update } G : G + P_{high}. \quad (5)$$

Depending on circumstances, for each class, the number of pseudo-labeled data differs, and without balancing, the imparity potentially induces a label bias. We balance U and G respectively, by choosing the same number of pseudo-labeled data for each class.

3.5. Details on Implementation

We use BERT (Devlin et al., 2018) as our baseline classifier T (Chen et al., 2020). Then, we distinguish superficial learning and fine-tuning by differentiating the epoch number and the learning rate; For superficial learning and fine-turning, the epoch number is 1 and 5, and the learning rate is `1e-6` and `1e-5`, respectively. Finally, our threshold θ for the confidence score is 0.9.

4. Experimental Setup

We run our experiments on NVIDIA DGX A100. In all experiments, we set the max sentence length to 256. Following Sections 4.1 and 4.2 explain the details on datasets and baselines, respectively.

4.1. Datasets

The following Table 1 illustrates the details of datasets used for experiments in Section 5.

Dataset	Classes	Val	Un	Test
IMDB	2	2,000	12,500	12,500
AGNews	4	2,000	30,000	1,900
Yahoo	10	5,000	20,000	6,000
DBpedia	14	2,000	10,000	5,000

Table 1: Data distribution of four benchmark datasets. The table presents the number of data per class (Un: Unlabeled data).

Dataset	Labeled Data		Models								Ours
	#train	#val	V	E	B	T	U	M	S	A	SuperST(B)
IMDB (**IM**)	5	5	51.2	53.6	53.3	60.2	58.7	54.1	66.5	67.1	**79.5** (18.5%↑)
	10	10	58.4	59.3	60.2	61.0	62.6	58.5	67.4	71.3	**81.7** (14.6%↑)
	200	2000	82.2	82.4	86.9	87.4	89.1	89.4	88.2	79.6	**89.6** (12.6%↑)
	2500	2000	85.8	91.2	89.8	90.3	90.8	91.3	90.4	83.6	**91.4** (9.3%↑)
AGNews (**AG**)	5	5	64.5	66.6	71.1	66.0	54.2	82.0	82.5	82.3	**84.6** (2.6%↑)
	10	10	72.4	72.6	75.2	75.7	76.3	85.6	86.1	85.2	**86.9** (2.0%↑)
	200	2000	83.9	84.5	87.5	88.1	88.3	89.2	89.3	89.3	**89.7** (0.4%↑)
	2500	2000	86.2	91.3	90.8	91.0	91.2	91.5	91.8	91.4	**92.1** (0.8%↑)
Yahoo! Answer (**Ya**)	5	5	31.5	32.4	43.8	51.2	50.5	60.1	56.2	55.2	**64.3** (16.5%↑)
	10	10	41.2	42.7	45.4	56.0	52.4	65.7	62.4	64.2	**66.1** (3.0%↑)
	200	2000	59.9	68.3	69.3	69.8	70.2	71.3	70.2	67.8	**71.8** (5.9%↑)
	2500	2000	70.2	73.4	73.2	73.5	73.6	74.1	73.1	71.2	**74.4** (4.5%↑)
DBpedia (**DB**)	5	5	71.2	68.0	72.8	94.0	92.2	96.1	93.4	94.9	**97.5** (2.5%↑)
	10	10	82.9	84.5	82.8	95.3	95.5	97.4	97.4	96.7	**98.5** (1.9%↑)
	200	2000	87.5	98.4	98.5	98.7	98.8	98.9	98.9	98.9	**99.0** (0.1%↑)
	2500	2000	90.1	99.0	99.0	99.0	99.1	99.2	99.1	99.1	**99.2** (0.1%↑)

Table 2: Results (test accuracy(%)) comparison of SSL models. All results report the average scores over five random samplings. Models are as follows: **V**AMPIRE, **E**LECTRA, **B**ERT, **T**Mix, **U**DA, **M**ixText , **S**ALNet, **A**LP, **SuperST**(**B**ERT).

We use four text classification benchmark datasets to evaluate the performance; IMDB review (Maas et al., 2011), AG News (Zhang et al., 2015), Yahoo! Answers (Chang et al., 2008), DBpedia (Mendes et al., 2012). We use the original test set and randomly sample 5−2500 sentences per class of the original set as the labeled set and 5−2000 sentences per class as a validation set. We use the validation set to determine the best model at each epoch. We also randomly sample 5000−30000 sentences per class and remove the labels to use as an unlabeled dataset. All data have a balanced class distribution.

4.2. Baseline Models

In order to evaluate the effectiveness of SuperST, we compare our approach with state-of-the-art SSL methods: BERT (Devlin et al., 2018), VAMPIRE (Gururangan et al., 2019), ELECTRA (Clark et al., 2020), TMix (Chen et al., 2020), UDA (Xie et al., 2020a), SALNet (Lee et al., 2021) and ALP (Kim et al., 2022). For fair comparisons, we adopt the same hyperparameters used in their respective papers and randomly sample the data for our experiments.

5. Results and Analysis

Table 2 demonstrates that SuperST achieves state-of-the-art performance in few-shot text classification. The effectiveness of applying superficial learning to the self-training mechanism is evident from the table; however, further examination of the results is essential. We have conducted experiments on widely recognized benchmark datasets to ensure the generalizability of our proposal. Furthermore, we present extensive experiments and analyses in the validity of superficial learning and SuperST, providing comprehensive insights into SuperST.

5.1. Comparison with SSL Baselines

SSL baselines, MixText, UDA and ALP (Chen et al., 2020; Xie et al., 2020b; Kim et al., 2022) utilize additional resources such as back translations (Sennrich et al., 2015) and WordNet (Miller, 1995). SALNet (Lee et al., 2021) provides a new approach by bootstrapping the classifier with lexicons extracted from the labeled data. These approaches focus on providing novel pseudo-labeling policies for the enhancement. On the other hand, SuperST provides a simple yet effective strategy solving the fundamental problem of noise in pseudo-labels.

We conducted experiments for 1) extreme few-shots (5, 10) and 2) relatively limited shots (200, 2500). SuperST outperforms all the baselines in all settings, illustrated in Table 2. The performance improvements vary within each setting, on the amount of labeled data used for training and validation data. SuperST aims to maximize the use of noisy data, and our approach demonstrates particular effectiveness in extremely few-shot settings, where noise more prone to occur. We can observe this claim in Table 2. The performance of SuperST for the 5- and 10-shots achieves at most 18.5% and 14.6% increase from the state-of-

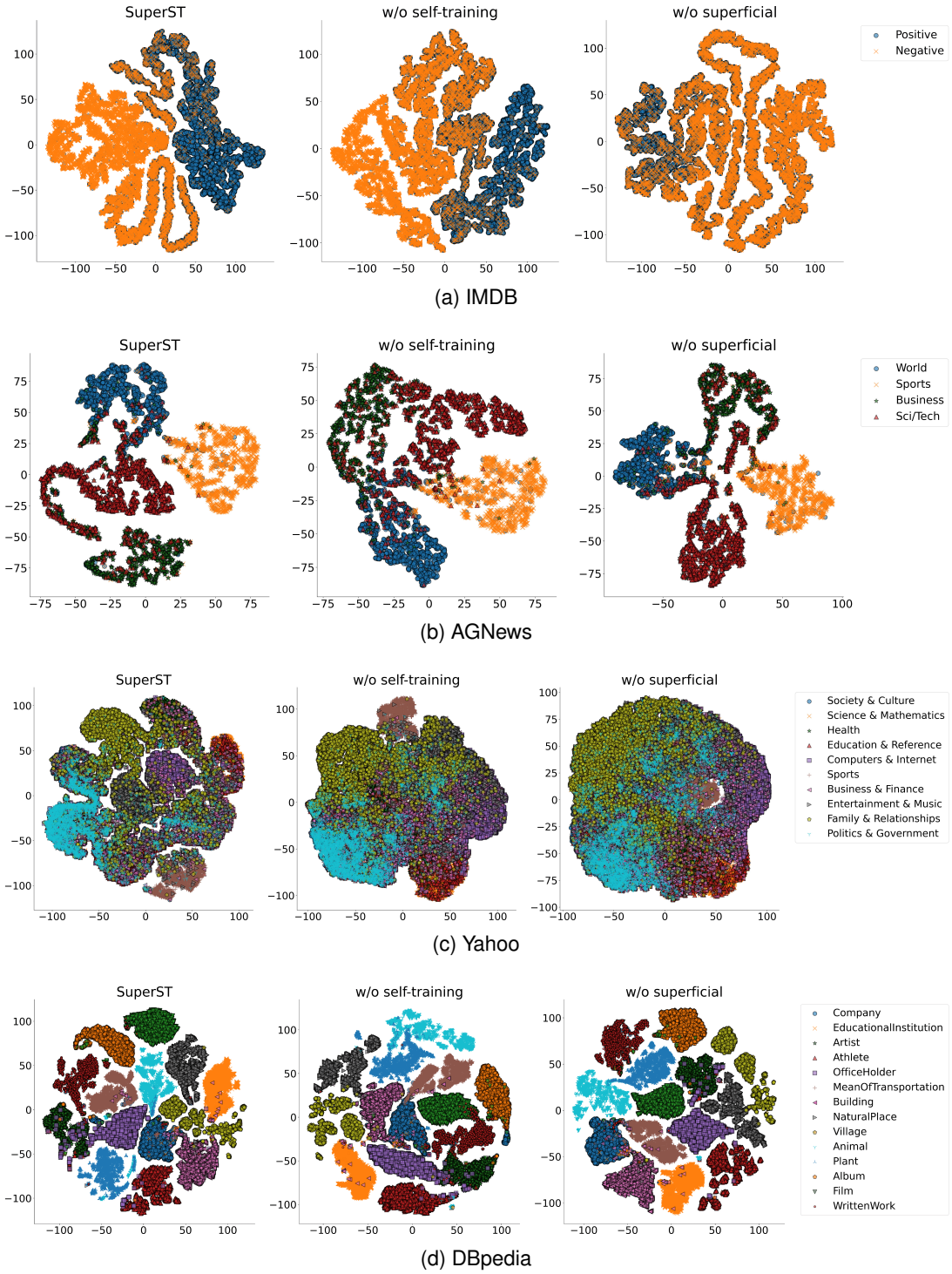

Figure 4: A t-SNE visualization of 1) SuperST, 2) SuperST without self-training, and 3) SuperST without superficial learning for all datasets.

the-art models, respectively. On the other hand, improvements of 200- and 2500-shot experiments are relatively small compared with the extreme few-shot settings. Such results empirically prove that SuperST, resolving the fundamental problem of noisy pseudo-labels, is the most effective for extreme few-shot setting experiments.

An intriguing issue is that, compared with other baselines, SuperST shows a relatively small performance gap regarding the number of initial labeled data. We demonstrate the SSL baseline performance of 5-, 10-, 200- and 2500-shot learning and while performances of baselines vary on a large scale, SuperST varies within roughly 10% comparing the scores of 5-shot and 2500-shot settings. For instance, on the IMDB, ALP performs 67.1% for

5-shot and 83.6% for 2500-shot learning, showing a 24.6% performance gap. The baselines highly depend on the size of initial data and the performance of extreme few-shot (5, 10) learning does not achieve satisfactory performance compared with relatively limited shot (200, 2500) learning. On the other hand, the gap between the 5-shot and 2500-shot performance of SuperST for the IMDB is 15.0%. Considering all the datasets, the performance gap of SuperST is 10.3% in average. Throughout observations on the results, SuperST effectively addresses the challenges associated with the dependence on initial data size and the deficiency of extreme few-shot learning.

5.2. Effectiveness of Superficial Learning

It is straightforward that the fundamental element of SuperST is superficial learning. From the introduction, we claim that current self-training faces a challenge: pseudo-labels often contain noise and potentially incorrect information that rapidly accumulate in iterations of self-training, eventually degrading the performance. Our claim continues on declaring that a specialized learning strategy is mandatory to effectively learn the noisy data. Based on this claim, in this section, we empirically prove that superficial learning is a simple yet effective solution for solving the challenge. In order to demonstrate the strength of superficial learning, we investigate further on whether superficial learning is indeed useful for catching the required information and mitigating the noise. We first conduct experiments examining the performance of 1) SuperST, 2) SuperST without self-training and 3) SuperST without superficial learning.

Model	IM	AG	Ya	DB
SuperST	**79.5**	**84.6**	**64.3**	**97.5**
w/o self-training	77.2	79.1	54.7	96.1
w/o superficial	63.5	80.6	45.2	96.4

Table 3: A 5-shot performance table of 1) SuperST, 2) SuperST without self-training, and 3) SuperST without superficial learning.

From Table 3, we can see that both self-training and superficial learning are the core components of SuperST. The importance of superficial learning is especially emphasized for the IMDB and Yahoo datasets. The performance of SuperST with and without superficial learning for each dataset varies by 25.2% and 42.3%, respectively. We also present the t-SNE visualization of all datasets in Figure 4. The figure illustrates the decision boundary prediction of 1) SuperST, 2) SuperST without self-training and 3) SuperST without superficial learning. The SuperST without self-training and SuperST without superficial learning both do not provide any clear decision boundary between classes, `positive` and `negative`. The decision boundary is clear in the visualization of SuperST's prediction and this result strengthens our claim on the effectiveness of superficial learning.

For the AGNews and DBpedia datasets on the other hand, the performance gap is more pronounced when considering self-training, as evident in Table 3. We examine the baseline scores of AGNews and DBpedia to investigate this phenomenon. As Table 2 illustrates, the baseline performance for AGNews and DBpedia is significantly higher compared to that of IMDB and Yahoo. This indicates that the initial classifier performs relatively well, and as a result, self-training even without superficial learning leads to some performance enhancements. The performance gaps of SuperST and SuperST without self-training for AGNews and DBpedia are 7.0% and 1.5%, respectively. These gaps are relatively smaller compared with IMDB and Yahoo, and the differences in decision boundary predictions, as seen in Figure 4(b) and (d), are not substantially large. However, it is apparent that SuperST, applying superficial learning to the self-training mechanism, achieves a meaningful enhancement. We can empirically conclude that superficial learning is a promising learning strategy for self-training.

5.3. Ablation Study

Superficial learning targets obtaining a partial understanding of noisy labels 'to some extent' and we utilize the learning rate and the epoch number to implement superficial learning. We have conducted extensive experiments varying the parameters and empirically concluded the appropriate parameters of superficial learning; 1 epoch with a `1e-6` learning rate is enough for superficial learning. Table 4 shows the performance of SuperST with varying epoch numbers on the initial superficial learning and Table 5 represents the performance with a varying learning rate of superficial learning.

Epochs	IM	AG	Ya	DB
<1	53.8	85.3	32.2	95.0
1	79.5	**84.6**	**64.3**	**97.5**
2	**79.7**	83.7	63.5	97.2
5	76.2	74.5	58.3	96.8
10	77.8	64.7	53.9	95.8
20	67.0	55.9	40.4	94.8

Table 4: 5-shot performance (accuracy(%)) with different epoch numbers for superficial learning.

We observe that when we set relatively large epoch numbers and learning rates, the classifier

Learning Rate	IM	AG	Ya	DB
1e-8	53.4	70.3	45.7	96.9
1e-7	52.4	**85.2**	63.5	97.3
1e-6	**79.5**	84.6	**64.3**	**97.5**
1e-5	68.7	54.5	41.5	92.8
1e-4	50.0	53.0	10.2	90.4

Table 5: 5-shot performance (accuracy(%)) with different learning rate for superficial learning.

does not perform well. On the other hand, SuperST does not also perform well with `1e-7` and `1e-8` learning rates which are too low. The classifier learns little knowledge from the pseudo-labeled data in these cases. We empirically confirm that the classifier achieves the best performance when trained with `1e-6` learning rate for 1 epoch.

While the two parameters are the core of superficial learning, we further investigate the parameter, confidence score of the classifier for pseudo-labeling. Illustrated in Algorithm 1, the threshold value θ represents a borderline of golden data for pseudo-labels. It is important to adjust the right value for the threshold as the bigger confidence value assigns more accurate pseudo-labels but in a small amount and the smaller confidence value assigns many pseudo-labels but with relatively low accuracy. Table 6 shows the performance varying the confidence threshold. We can see that the value `0.9` is the optimal value.

Confidence	IM	AG	Ya	DB
0.9	**79.5**	**84.6**	**64.3**	**97.5**
0.8	79.2	83.5	65.1	97.0
0.7	74.1	82.3	65.1	97.5
0.6	74.3	81.3	62.3	96.9

Table 6: 5-shot performance (accuracy(%)) of models with different confidence score threshold.

We have analyzed on the performance of SuperST and the validity of each component, demonstrating their effectiveness in enabling the classifier to learn the necessary information.

5.4. Keyword Extraction Analysis

We examine several strategies for extracting essential information based on keyword lexicons. We have proven that the pseudo-label by lexical knowledge is effective and we are now curious whether our suggested method of utilizing TF-IDF is valid. We employ TF-IDF as our baseline method for constructing lexicons as it is a widely used traditional method for extracting lexical knowledge (Guo and Roth, 2021; Xiong et al., 2021). We inspect three methods for constructing lexicons: 1) random, 2) TF-IDF, 3) KeyBERT[3] and 4) KeyBERT+TF-IDF (KBTI). It is not surprising that random labeling performs the worst. Interestingly, Table 7 indicates that TF-IDF is a sufficiently competitive technique for lexicon constructions. While TF-IDF achieves the best performance, the relatively low performance of KBTI is odd since the lexicons should contain more knowledge than the lexicons constructed applying each method respectively. Through this analysis, we conclude that KBTI produces lexicons containing more noises that deteriorate the performance.

Model	IM	AG	Ya	DB
Random	55.1	81.8	41.2	94.3
TF-IDF	**79.5**	**84.6**	**64.3**	97.5
KeyBERT	77.4	83.9	61.9	**97.9**
KBTI	73.7	81.3	60.2	97.0

Table 7: 5-shot performance (accuracy(%)) of different keyword matching methods.

6. Conclusion

SuperST shows the effectiveness of superficial learning utilized in self-training for few-shot text classification. We empirically proved our claim that superficial learning effectively resolves the problem of noisy pseudo-labels. While it is surprisingly simple, it achieves an outstanding performance. Our extensive analyses of each component in SuperST and visualization of decision boundary predictions verify that it is enough for the classifier to train noisy data by superficial learning; 1 epoch with a `1e-6` learning rate. We have maximized the utilization of noisy data by SuperST which is particularly effective in extreme few-shot settings. For future work, we plan to apply SuperST on more various classification tasks. Also, we plan to design a more sophisticated algorithm for extracting lexical knowledge and explore the effectiveness of superficial learning in other NLP tasks.

Acknowledgements

This research was supported by the NRF grant (RS-2023-00208094) and the AI Graduate School Program (No. 2020-0-01361) funded by the Korean government (MSIT). Han is a corresponding author.

[3] https://github.com/MaartenGr/KeyBERT

Bibliographical References

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *Proceedings of the International Conference on Artificial Intelligence, AAAI*, pages 830–835.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. *arXiv preprint arXiv:2004.12239*.

Yiming Chen, Yan Zhang, Chen Zhang, Grandee Lee, Ran Cheng, and Haizhou Li. 2021. Revisiting self-training for few-shot learning of language model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 9125–9135.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Hongyan Cui, Gangkun Wang, Yuanxin Li, and Roy E. Welsch. 2022. Self-training method based on GCN for semi-supervised short text classification. *Inf. Sci.*, 611:18–29.

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*.

Ecaterina Sarah Frăsineanu. 2013. Approach to learning process: superficial learning and deep learning at students. *Procedia-Social and Behavioral Sciences*.

Ruohao Guo and Dan Roth. 2021. Constrained labeled data generation for low-resource named entity recognition. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4519–4533.

Suchin Gururangan, Tam Dang, Dallas Card, and Noah A. Smith. 2019. Variational pretraining for semi-supervised text classification. In *Proceedings of the International Conference of the Association for Computational Linguistics, ACL*, pages 5880–5894.

Joonghyuk Hahn, Hyunjoon Cheon, Kyuyeol Han, Cheongjae Lee, Junseok Kim, and Yo-Sub Han. 2021. Self-training using rules of grammar for few-shot NLU. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 4576–4581.

Joonghyuk Hahn, Hyunjoon Cheon, Elizabeth Orwig, Su-Hyeon Kim, Sang-Ki Ko, and Yo-Sub Han. 2023. GDA: grammar-based data augmentation for text classification using slot information. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 7291–7306.

Hwiyeol Jo and Ceyda Cinarel. 2019. Delta-training: Simple semi-supervised text classification using pretrained word embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP*, pages 3456–3461.

Hazel H. Kim, Daecheol Woo, Seong Joon Oh, Jeong-Won Cha, and Yo-Sub Han. 2022. ALP: data augmentation using lexicalized pcfgs for few-shot text classification. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, pages 10894–10902.

Ju-Hyoung Lee, Sang-Ki Ko, and Yo-Sub Han. 2021. Salnet: Semi-supervised few-shot text classification with attention-based lexicon construction. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, pages 13189–13197.

Christopher Liao, Theodoros Tsiligkaridis, and Brian Kulis. 2022. Pick up the PACE: fast and simple domain adaptation via ensemble pseudo-labeling. *arXiv preprint*, 2205.13508.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the International Conference on Association for Computational Linguistics, ACL*, pages 142–150.

Ference Marton and Roger Säljö. 1976. On qualitative differences in learning: I—outcome and process. *British journal of educational psychology*, 46(1):4–11.

Pablo N. Mendes, Max Jakob, and Christian Bizer. 2012. Dbpedia: A multilingual cross-domain knowledge base. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC*, pages 1813–1817.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S. Rawat, and Mubarak Shah. 2021. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *9th International Conference on Learning Representations, ICLR*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Anders Søgaard. 2010. Simple semi-supervised training of part-of-speech taggers. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 205–208.

Jesper E Van Engelen and Holger H Hoos. 2020. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Zhu Xiaojin. 2008. Semi-supervised learning literature survey. *Computer Sciences TR*, 1530.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020a. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. 2020b. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*.

Yijin Xiong, Yukun Feng, Hao Wu, Hidetaka Kamigaito, and Manabu Okumura. 2021. Fusing label embedding into BERT: an efficient improvement for text classification. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1743–1750.

Weiyi Yang, Richong Zhang, Junfan Chen, Lihong Wang, and Jaein Kim. 2023. Prototype-guided pseudo labeling for semi-supervised text classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL*, pages 16369–16382.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the International Conference on Neural Information Processing Systems, NeurIPS*, pages 649–657.

A. Statistical-based Keyword Extractions

We employ statistical- and model-based approaches for keyword extractions. We simply use language models as a model-based approach for keyword extractions. We illustrate one of the baseline statistical-based keyword extraction, the algorithm for extracting keywords from the unlabeled dataset X_U based on the statistical approach.

We calculate the term frequency $tf(w, X_U)$ of the data in X_U that contains a word w with the following equation:

$$tf(w, X_U) = \frac{\sum_{x_u \in X_U} |x_u|_w}{\sum_{x_u \in X_U} |x_u|},$$

where we denote the number of words in x_u as $|x_u|$, the number of w in x_u as $|x_u|_w$. We calculate the inverse document frequency $idf(w, X_U)$ of w in X_U with the following equation:

$$idf(w, X_U) = \log\left(\frac{|X_U|}{|X_U|_w + 1}\right),$$

where we denote the number of data in X_U as $|X_U|$, the number of data that have w as $|X_U|_w$.

Using tf and idf, we compute a score $TF\text{-}IDF(w)$ of w to determine whether w is a keyword:

$$TF\text{-}IDF(w, X_U) = tf(w, X_U) \times idf(w, X_U).$$

We choose the top 3 words with a high $TF\text{-}IDF$ score and construct a set $K_\emptyset$ of keywords.

B. Baseline Models

We illustrate the details of all the baselines we employed for our work.

- VAMPIRE: Gururangan et al. (2019) propose a lightweight pretraining framework. They utilize a variational autoencoder to pretrain a unigram document model.
- ELECTRA: ELECTRA (Clark et al., 2020) is a pretrained language model that trains the generator and the discriminator. We used the pretrained electra-base-discriminator model for text classification.
- BERT: BERT (Devlin et al., 2018) is a pretrained language model that trains the model using a masking language modeling (MLM) objective. We used the pretrained BERT-based-uncased model for text classification.
- TMix: TMix (Chen et al., 2020) is an interpolation-based augmentation method that creates a large number of augmented training samples by interpolating text in hidden space.
- UDA: Xie et al. (2020a) present a new data augmentation method that generates diverse and realistic noise. The method enforces the model to be consistent with respect to these noises.
- MixText: MixText (Chen et al., 2020) is a method that introduces a new data augmentation method to solve the overfitting problem.
- SALNet: SALNet (Lee et al., 2021) is a bootstrap learning framework for few-shot text classification. They bootstrap the classifier using a combination of the trained classifier and the constructed lexicons.
- ALP: Kim et al. (2022) present the ALP for few-shot text classification, which generates augmented samples with diverse syntactic structures with plausible grammar.