SALNet: Semi-Supervised Few-Shot Text Classification with Attention-based Lexicon Construction

Ju-Hyoung Lee[1], **Sang-Ki Ko**[2], **Yo-Sub Han**[1]
[1]Yonsei University, Seoul, Republic of Korea
[2]Kangwon National University, Kangwon, Republic of Korea
juhyounglee@yonsei.ac.kr, sangkiko@kangwon.ac.kr, emmous@yonsei.ac.kr

Abstract

We propose a semi-supervised bootstrap learning framework for few-shot text classification. From a small number of the initial data, our framework obtains a larger set of reliable training data by using the attention weights from an LSTM-based trained classifier. We first train an LSTM-based text classifier from a given labeled dataset using the attention mechanism. Then, we collect a set of words for each class called a lexicon, which is supposed to be a representative set of words for each class based on the attention weights calculated for the classification task. We bootstrap the classifier using the new data that are labeled by the combination of the classifier and the constructed lexicons to improve the prediction accuracy. As a result, our approach outperforms the previous state-of-the-art methods including semi-supervised learning algorithms and pretraining algorithms for few-shot text classification task on four publicly available benchmark datasets. Moreover, we empirically confirm that the constructed lexicons are reliable enough and substantially improve the performance of the original classifier.

Introduction

Recently, text classifiers using deep learning show great success in various NLP tasks due to lots of labeled training data. However, one critical limit is that these effective classifiers are hard to make when there is not enough labeled data. Often it is difficult and expensive to obtain a reasonable number of labeled data since it requires many well-trained human annotators. On the other hand, a traditional approach for text classification is to use a domain lexicon for classification since we can correctly decide a class using the lexicon and the size of the training data is irrelevant. The efforts to address the data sparsity problem are to make lexicons of each domain and use it for text classification (Lu and Tsou 2010; Lei et al. 2011; Hailong, Wenyan, and Bo 2014; Bandhakavi et al. 2017; Lee et al. 2018). On the other hand, it is not easy to automatically make a high-quality lexicon for a new domain (and. Binbin Chen and Bernstein 2016; Feng et al. 2018). Semi-Supervised Learning (SSL) is an approach to use both a small number of labeled data and a large number of unlabeled data in training. The traditional SSL methods based on neural network (Yarowsky 1995; Blum and Mitchell 1998; Li and Liu 2003; Zhu, Ghahramani, and Lafferty 2003; Rosenberg, Hebert, and Schneiderman 2005; Wang and Zhang 2007; Yaslan and Cataltepe 2010; Jo and Cinarel 2019) use only high confidence predictions of classifiers or utilize the agreement among the different classifiers, for pseudo-labeling. These methods can lead to accumulated classification errors along the training process. Therefore, it remains challenging to train classifiers for text classification under semi-supervision, because it still requires hundreds of thousands of labeled training data to achieve satisfactory performance.

We combine the deep learning classifier with the lexicon and tackle the labeled-data sparsity problem. With respect to the attention mechanism, important words often have high attention weights for text classification (Cho et al. 2014; Sutskever, Vinyals, and Le 2014; Bahdanau, Cho, and Bengio 2015; Luong, Pham, and Manning 2015; Xu et al. 2015). We visualize the attention layer and empirically confirm that the attention mechanism indeed assigns higher weights to words that can represent the class of input data. We also observe that even though the performance of the initially trained classifier is poor, the classifier assigns higher weights to the important words of the data predicted with high confidence. For instance, the following is a visualization example of the attention mechanism on one of the IMDB review dataset. We collect these high weight words from the attention mechanism and make a set of words (called *lexicon*) for our semi-supervised classification.

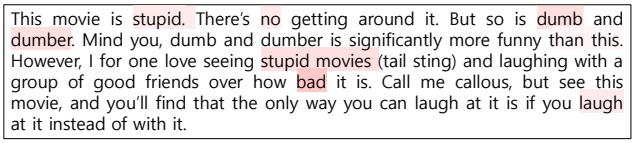

Figure 1: Visualization of the attention mechanism using a LSTM-based classifier trained with few labeled data. Darker colors indicate higher attention weights.

We propose a semi-supervised few-shot text classification with attention-baed lexicon construction when there is only a small number of labeled data. Our approach is closely related to neural bootstrapping methods. The bootstrapping methods for SSL use only predictions of neural-based classi-

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

fiers for labeling data. One major drawback of the bootstrapping is that if the initially trained classifiers have very low performance, then the new training set becomes unreliable because it may contain lots of incorrectly-labeled data. We overcome the problem by utilizing a set of words, which can explicitly predict the unlabeled data using pattern matching.

We evaluate our approach using four publicly available benchmark datasets and compare the performance with the previous state-of-the-art methods including the other semi-supervised learning algorithms (Yarowsky 1995; Jo and Cinarel 2019) and pretraining algorithms (Gururangan et al. 2019; Devlin et al. 2019). The experimental results demonstrate that our approach, **S**emi-supervised with **A**ttention-based **L**exicon construction **Net**work (SALNet), outperforms the previous state-of-the-art methods on four benchmark datasets.

Our main contributions are as follows:

- We propose a semi-supervised bootstrap learning framework that utilizes lexicons constructed by attention mechanism, and our approach has improved accuracy of at least 1% to 8% from initially trained classifiers.
- We verify the effectiveness of the constructed lexicons by improving the accuracy of at least 1% to 9% from the original classifier.
- We demonstrate experimentally that our approach is an effective approach when there is an extremely small labeled dataset.

Background

Here we provide some background knowledge on neural network models discussed in our paper.

Attention mechanism. In the sequence-to-sequence translation model, Bahdanau et al. (2015) hypothesize that the fixed-length context vector c is a bottleneck since the length of the input sequence can vary. They proposed the attention mechanism that computes the context vector by looking at relevant parts from the hidden states of the encoder. Indeed, the attention mechanism has proven surprisingly useful in many tasks in natural language processing. They defined each conditional probability at time i depending on a dynamically computed context vector c_i as follows:

$$p(y_i|y_1, y_2, \ldots, y_{i-1}, \mathbf{x}) = \mathrm{softmax}(g(\hat{s_i})),$$

where $\hat{s_i}$ is the hidden state of the decoder RNN at time i computed by $\hat{s_i} = R(y_{i-1}, \hat{s_{i-1}}, c_i)$.

The context vector c_i is computed as a weighted sum of the hidden states from encoder: $c_i = \sum_{j=1}^{n} \alpha_{ij} h_j$, where

$$\alpha_{ij} = \frac{\exp(\mathrm{score}(s_{i-1}, h_j))}{\sum_{k=1}^{n} \exp(\mathrm{score}(s_{k-1}, h_j))}.$$

Here the function 'score' is called an *alignment function* that computes how well the two hidden states from the encoder and the decoder, respectively, match. For example, $\mathrm{score}(s_i, h_j)$, where s_i is the hidden state of the encoder at time i and h_j is the hidden state of the decoder at time j implies the probability of aligning the part of the input sentence around position i and the part of the output sentence around position j.

Transformer. Vaswani et al. (2017) propose the Transformer, a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output. The transformer consists of stacked multi-head attention and parameterized linear transformation layers for both the encoder and decoder. At each layer, the multi-head attention employs h attention heads and performs the self-attention mechanism to capture various context of the input sentence.

Methods

The proposed method involves the following steps:

1. Create a base classifier from a given labeled data, which is very few. Importantly, we train the base classifier to overfit the training set. The classifier must include an attention mechanism to collect crucial words for each classification.
2. Re-run the base classifier that has an attention mechanism on the unlabeled dataset U.
3. Obtain a set of crucial words for predicting U, which we call a lexicon of our method; in other words, we use U and the classifier attention weights to create the lexicon for sets of crucial words.
4. Predict the labels of the data in U using the trained classifier and the lexicons.
5. Add the new labeled data to the training set, and train the classifiers again, starting with the first step.
6. Repeat the process above until pseudo-labeled data is no longer added to training set. Using the development set at each epoch, we do early-stopping during all the training.

Figure 2 is an overview of our method. We use an attention-based LSTM (Wang et al. 2016) for constructing a set of crucial words for text classification. The attention-based LSTM extracts a set of relevant words based on their attention weights for each class. Ruder and Plank (2018) showed that the relative order of confidence is more robust than absolute confidence and thus, we select the top n unlabeled data with the highest confidence from the classifier in each class. Then, we select m words that have the highest attention weights from each of the selected n data, and make n sets that consist of m words. We regard the collected word set as a lexicon for the corresponding class. Table 1 shows an example of a set of crucial words in each lexicon of the AG News dataset obtained from an initially trained classifier.

We count the number of matching words from unlabeled data with respect to the lexicon of each class. Then, we regard the number of matching words as a prediction confidence of the corresponding class and assign the class of the highest confidence as a predicted label. If there is a tie for the number of matching words between two classes, then we ignore both classes and do not predict the label since there is a

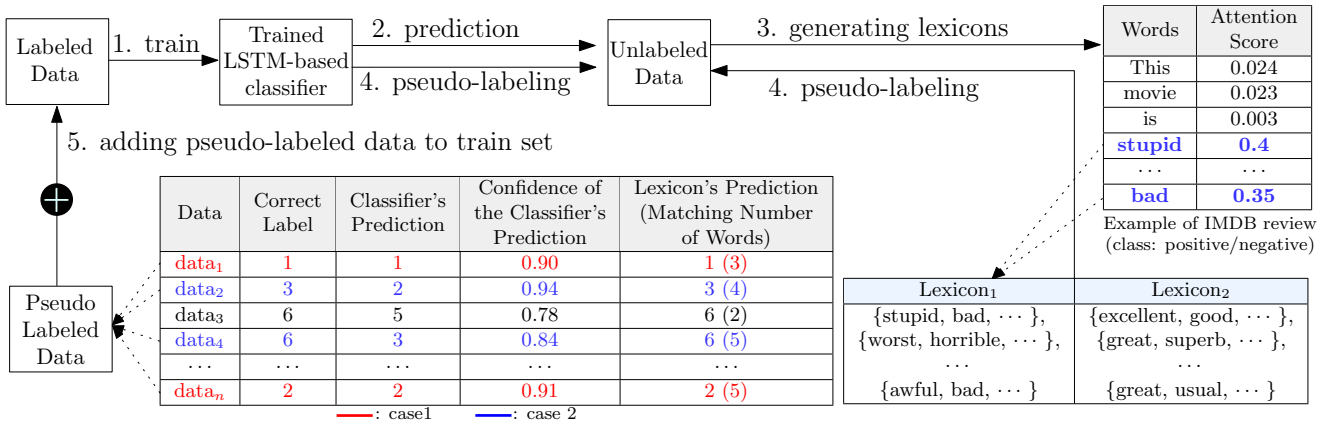

Figure 2: Our proposed method, using both attention-based classifier and lexicons. We set the t_1= 3 and t_2 = 4 in our method and set 0.9 as the threshold to verify the high confidence.

Class	Each lexicon that consists of n word sets (m words per set)
World	{Iraqi, hostage, Allawi, iyad, minister, prime}, {officials, ⋯ }, ⋯ ,{egypts, ⋯ }
Sports	{yankees, championship, sox, league, game, series}, {beating, ⋯ }, ⋯ ,{outfielder, ⋯ }
Business	{mae, fannie, mortgage, finance, company, accounting}, {stocks, ⋯ }, ⋯ ,{profits, ⋯ }
Science/Tech	{concern, software, threats, infrastructure, cyber, viruses}, {system, ⋯ }, ⋯ ,{network, ⋯ }

Table 1: An example of four lexicons from the AG News dataset with four classes.

possibility of an incorrect prediction. We have two cases for pseudo-labeling, as shown in Figure 2.

- Case 1: If the classifier predicts the unlabeled data with high confidence and the lexicon has at least t_1 matching words, then we label the data according to the prediction of the classifier. In Case 1, we use the lexicon for selecting the correct label among the predictions of the classifiers.
- Case 2: If the lexicon has at least t_2 matching words, then we label the data according to the lexicon prediction. In other words, in Case 2, we use the lexicon to decide the label of unlabeled data that the classifier incorrectly predicts with low confidence.

If we use more than two classifiers in SALNet, we can add the pseudo-labeled data by repeating the process of Case 1 and Case 2 for the additional classifiers. Once we obtain a new dataset after pseudo-labeling, the new dataset may have a different number of data in each class. This imbalance may make a classifier overfit to larger classes. We avoid this problem by selecting the same number of data from each class, which is the number of data in the smallest class, for the next training step.

Experimental Setup

We describe the experimental setup for evaluation.

Datasets

We use four benchmark datasets to evaluate the performance of our proposed method across different domains; IMDB review (Maas et al. 2011), AG News (Zhang, Zhao, and LeCun 2015), Yahoo! Answers (Chang et al. 2008), DBpedia (Mendes, Jakob, and Bizer 2012). We take only 1% of the original training data as our labeled data with random sampling. In the new labeled dataset, we use 85% of its data as a training set, and 15% of its data as a development set. We remove the labels of the remaining 99% data. All data have a balanced class distribution. We use the development set to determine early-stopping at each epoch. Table 2 presents the data distribution.

Hyperparameters

We use pretrained GloVe (Pennington, Socher, and Manning 2014) as word embedding for all experiments. GloVe is trained on a dataset of 42 billion tokens with a vocabulary of 1.9 million words and has 300 dimension embedding vectors. Since attention-based LSTM (Wang et al. 2016) and TextCNN (Kim 2014) are simple and have a high performance, we select the two basic models as classifiers (Jo and Cinarel 2019). The TextCNN consists of filter windows of size 3, 4, 5 with 100 feature maps each of which is followed by ReLU activation and max-pooling. The attention-based LSTM consists of 300 hidden sizes. We train all classifiers with a batch size of 128, and optimize them using the Adam optimizer (Kingma and Ba 2015) with 0.001 and 0.005 learning rates.

Our proposed method, SALNet, uses size 50 of lexicons and three (=t_1) and four (=t_2) matching words for predicting a classes of unlabeled data. With respect to size of lexicon, our empirical study shows that when the data size is small or the number of classes is large, it is better to have a small lexicon for each class. In regard to matching words, we empiri-

Dataset	Label Type	Classes	Max Length	Train	Dev	Unlabeled	Test
IMDB review	Review Sentiment	2	300	212	38	24,750	25,000
AG News	News Topic	4	150	1,020	180	118,800	7,600
Yahoo! Answer	QA Topic	10	100	1,700	300	198,000	46,400
DBpedia	Wikipedia Topic	14	200	1,190	210	138,600	70,000

Table 2: Data distribution of four benchmark datasets.

cally observe that the lexicon ambiguously predicts the class of data with one or two matching words from each class. For instance, our approach with one matching word incorrectly predicts the following example from the Yelp review dataset as positive, because it contains a positive word "good".

> "Even though Midler and Alvarado give good performances this film really drags and I was bored silly by the end"

Therefore, we assume that it would be better to have three or more matching words for correctly identifying a class of unlabeled data instead of one or two matching words with respect to lexicons.

Baselines

We compare our method with a traditional SSL and three state-of-the-art baselines to verify the effectiveness of our approach. We set 0.9 as the threshold to verify the high confidence for all bootstrapping methods.

- **Self-training (Yarowsky 1995):** Self-training is a one of the simplest approach for SSL. Since the source code is not available in public, we implement this method using their pseudo algorithm.
- **Delta-training (Jo and Cinarel 2019):** Since the code was not published, we implement it ourselves based on the pseudo algorithm presented in their paper. Since all baselines including our approach use high confidence as a threshold, we use the high confidence instead of a model ensemble in delta-training for the fairness of experiments.
- **VAMPIRE (Gururangan et al. 2019):** Variational Methods for Pretraining In Resource-limited Environments (VAMPIRE) pretrained a unigram document model as a variational autoencoder (VAE) on unlabeled data and used its internal states as features in a downstream classifier. We use the same hyperparameters used in their paper for the experiments.
- **BERT (Devlin et al. 2019):** We use the pretrained BERT-based-uncased-model and fine-tuned it for the text classification.

Experimental Results and Analysis

We evaluate our approach and baselines to test the effectiveness of our approach for few-shot text classification task.

Main Results

Table 3 demonstrates that our approach outperforms all the baselines, especially when the performance of initially trained classifiers is less than 80%. The semi-supervised bootstrapping approaches such as self-training and delta-training, which uses only high confidence predictions of classifiers, show low performance when the performance of initially trained classifiers is lower than 75%. The results show the drawback of the bootstrapping that it is only effective when the initially trained classifiers perform well. On the other hand, SALNet shows relatively robust performance, compared to other bootstrapping algorithms. The performance slightly improves on Yahoo! Answer, since there is a small number of data that the classifier predicted with high confidence. On DBpedia, since the performance of the initially trained classifier is already over 94%, the improvement of performance is very small.

We observe that fine-tuning the pretrained BERT achieves the best performance on Yahoo! Answer and DBpedia. Recall that BERT uses word-piece tokenization and generates a contextualized vector for each word while GloVe encodes a word into a fixed-sized vector representation. Therefore, each pretrained model may perform differently depending on the characteristics of dataset. Importantly, our approach with low resources can outperform pretrained BERT in certain domains, compared to pretrained BERT that requires a significant amount of computational resources and a large-scale dataset. Moreover, SALNet with BERT as a classifier outperforms the original BERT and the self-training with BERT on four benchmark datasets. The BERT in SALNet repeats the fine-tuning process whenever additional pseudo-labeled data is obtained.

Experiments for few-shot learning. We conduct experiments to demonstrate the effectiveness of SALNet when there is an extremely small number of labeled data. We randomly select only 0.2% of the original training set and assign 15% of the labeled training set to the development set. As mentioned earlier, bootstrapping methods except for SALNet lead to accumulated classification errors along the semi-supervised learning process and eventually result in low improvement of performance. On the other hand, SALNet shows a stable improvement of performance for few shot text classification since SALNet employs reliable lexicons, as shown in Table 4. We observe that BERT exhibits the best performance on two datasets (Yahoo! Answer and DBpedia), compared to SALNet using the baseline classifiers such as attention-based LSTM and TextCNN. However, SALNet outperforms the original BERT and the self-training with BERT when we use BERT in SALNet.

Empirically, we confirm the relative robustness of our approach when utilizing 0.2% to 0.9% of the original training

Method	IMDB review	AG News	Yahoo! Answer	DBpedia
Baseline (attention-based LSTM)	74.35 (3.38)	85.78 (1.18)	49.77 (1.15)	94.41 (0.58)
Baseline (TextCNN)	75.39 (1.20)	88.08 (0.49)	49.66 (0.36)	96.10 (0.10)
Self-training (TextCNN)	76.43 (2.42)	88.71 (0.35)	51.35 (0.25)	97.05 (0.10)
Self-training (attention-based LSTM)	74.97 (4.05)	87.01 (1.19)	51.80 (1.50)	95.80 (0.47)
Self-training (BERT)	82.53 (4.20)	89.45 (0.21)	56.71 (3.06)	98.38 (0.18)
Delta-training	77.68 (1.49)	85.31 (0.56)	50.60 (0.46)	96.77 (0.14)
BERT (fine-tuning with 1% of the total labeled data)	79.74 (3.91)	88.76 (0.18)	57.58 (0.65)	98.01 (0.17)
VAMPIRE	64.64 (7.60)	85.88 (0.42)	50.57 (1.27)	92.29 (1.65)
SALNet (attention-based LSTM)	79.00 (2.06)	88.22 (0.28)	51.97 (1.14)	96.62 (0.28)
SALNet (attention-based LSTM + TextCNN)	80.33 (1.76)	89.23 (0.22)	53.48 (0.81)	97.48 (0.13)
SALNet (attention-based LSTM + BERT)	**84.87** (1.40)	**90.35** (0.26)	**59.08** (0.76)	**98.66** (0.24)

Table 3: Performance (test accuracy (%)) comparison with baselines. Each result is an average over five random samplings with standard deviation in parentheses, and the highest mean result shown in bold.

Method	IMDB review	AG News	Yahoo! Answer	DBpedia
Training set (0.2% of total labeled data)	42	204	340	238
Dev set	8	36	60	42
Baseline (attention-based LSTM)	66.16 (1.18)	80.26 (3.81)	38.95 (0.87)	82.45 (3.14)
Baseline (CNN)	63.46 (3.16)	85.67 (0.78)	44.18 (0.67)	91.77 (0.44)
Self-training (TextCNN)	56.98 (3.87)	86.25 (0.96)	45.57 (1.57)	93.79 (0.50)
Self-training (attention-based LSTM)	59.65 (6.46)	81.09 (2.54)	41.20 (2.79)	84.30 (3.19)
Self-training (BERT)	55.25 (4.07)	87.68 (0.64)	46.47 (6.28)	97.96 (0.39)
Delta-training	55.79 (4.01)	86.18 (0.83)	46.44 (0.59)	92.85 (0.58)
BERT (fine-tuning with 0.2% of the total labeled data)	59.11 (2.54)	86.51 (0.75)	47.82 (1.36)	97.55 (0.51)
VAMPIRE	53.58 (7.38)	76.64 (1.68)	41.66 (2.13)	84.47 (1.86)
SALNet (attention-based LSTM)	69.94 (1.91)	85.59 (1.18)	43.08 (2.27)	92.76 (1.10)
SALNet (attention-based LSTM + TextCNN)	71.34 (3.22)	87.68 (0.57)	46.59 (1.26)	95.38 (0.71)
SALNet (attention-based LSTM + BERT)	**75.77** (1.08)	**88.59** (0.41)	**53.65** (0.95)	**98.23** (0.12)

Table 4: Experimental results using an extremely small number of labeled data. Each result is an average over five random samplings with standard deviation in parentheses.

set. On the other hand, if we take less than 0.2% of the original training set, the size of data is too small to contain words that represent each class. Therefore, SALNet constructs the unreliable lexicons, and it can lead to accumulated errors along the training process.

Analysis

Attention mechanism. Figure 3 visualizes the attention layer of the initially trained attention-based classifier for identifying lexicon in our model: the darker the color, the higher the score. We notice that the initially trained classifier assigns a higher score to important words of the data predicted with high confidence; in other words, the attention mechanism successfully identifies relevant words for text classification using attention scores. This is why the attention mechanism is crucial in our method for effective classification.

Figure 3: Visualization of the attention score of data predicted by the initially trained classifier with high confidence.

Effectiveness of the constructed lexicons. In order to label the unlabeled data for semi-supervised learning, we pre-

Lex. Size		IMDB	AG News	Yahoo	DBpedia
100	r.	13.98	16.00	10.22	37.72
	ac.	86.17	92.93	81.54	94.06
150	r.	16.97	18.97	12.67	41.94
	ac.	83.61	92.81	81.80	97.50
200	r.	19.86	20.90	14.92	44.08
	ac.	81.47	92.86	80.19	94.54

Table 5: The ratio of data predicted by lexicons (r.) and test accuracy of the lexicons (ac.).

BERT + Lex.	IMDB	AG News	Yahoo	DBpedia
BERT	80.25	87.64	54.49	98.25
BERT + 100	89.55	90.51	61.45	98.71
BERT + 150	87.79	90.43	59.93	98.30
BERT + 200	88.99	89.71	54.83	98.67

Table 6: Performance (test accuracy (%)) on four benchmark datasets using added train set that obtained from lexicons. Each result is an average over five random samplings.

dict the class of data when the data contains a word that is contained in the lexicon for the class. Table 5 shows the ratio of data predicted by lexicons and the accuracy of the lexicons. We also conduct experiments to verify the effectiveness of the constructed lexicons. Since BERT shows the best performance among the baselines, we use BERT to demonstrate the effectiveness of lexicons. We obtain a new dataset predicted by the lexicons, and select the same number of data from each class to avoid overfitting to a larger class. Then, we update the training dataset with the new dataset and train BERT with the updated dataset. Table 6 demonstrates that the constructed lexicons effectively improve the performance of BERT. We remark that this experiment uses BERT to verify the effectiveness of the constructed lexicons, while the previous experiment uses BERT for bootstrapping learning of SALNet.

Ablation study. We perform ablation studies to show the effectiveness of each component in SALNet. We exclude the lexicon in SALNet to verify the effectiveness of the lexicon. Table 7 shows that the combination of the lexicons and the attention-based LSTM outperforms the model only with attention-based LSTM on four datasets. The effectiveness is especially prominent when labeled data is extremely limited. Since SALNet constructs the lexicons using the data predicted by the classifier with high confidence and the constructed lexicons explicitly predict the unlabeled data using pattern matching, the lexicons improve the reliability of pseudo-labeling. We conduct further experiments to explore the performance according to the number of classifiers in SALNet, and the results are shown in Table 7. Since we

Method		IMDB	AG News	Yahoo	DBpedia
LSTM	A	75.43	87.70	50.98	96.25
	B	64.73	82.50	37.92	85.25
SALNet (#)	A	79.00	88.22	51.97	96.62
	B	69.94	85.59	43.08	92.76
LSTM+ CNN	A	77.85	88.78	53.16	96.99
	B	68.31	85.12	44.33	90.06
SALNet (*)	A	80.33	89.23	53.48	97.48
	B	71.34	87.68	46.59	95.38

Table 7: Ablation experiments using 1% (A), 0.2% (B) of the original training set. Each result is an average over five random samplings. #: attention-based LSTM, *: attention-based LSTM + CNN

can obtain an increased training set that consists of many predicted labels with high confidence if we use two classifiers rather than one for pseudo-labeling, then using two classifiers outperforms only using one classifier in SALNet. Also, there is a difference in the performance of two classifiers since the initial performance of TextCNN outperforms attention-based LSTM on four datasets.

Analysis of failure cases. We manually categorize the resulting errors into two types: type-1, involving short sentences of fewer than ten words, and type-2, which involve ambiguous crucial words from each lexicon. Table 8 shows a few examples of types and errors. From type-1, we can see that the short-sentences examples do not contain crucial words for text classification. In this case, the lexicon contains words that do not belong to the core of the class. For the type-1 errors, we plan to exclude data of less than ten words in the training set. Type-2 errors are caused by ambiguous words of each class. For example, if there are classes such as “artist” and “album”, the lexicon of each class can include “singer”, “sing”, “fan” and “guitar”. For type-2 errors, we plan to count the number of the ambiguous words in each class and assign a weight to keywords according to the confidence value of the classifier.

Related Work

Most of the prior works on SSL for text classification has been covered the type of bootstrapping, adversarial training as well as the type of pretraining. In this section, we introduce several studies for semi-supervised learning discussed in our paper.

Bootstrapping Algorithms

The bootstrapping methods such as self-training (Yarowsky 1995), co-training (Blum and Mitchell 1998), tri-training (Zhu 2005) leverage the predictions of the neural network model on unlabeled data to obtain additional information that can be used during training.

Type	Example	Predicted	Ground Truth	Matching words
Type-1	where can sony ericsson?	N	E	None
	Antoinette Spaak Antoinette M.	N	O	
Type-2	Ronald Isley Ronald Isley is an American recording artist songwriter record producer and occasional actor.Isley is better known as the lead singer and founding member of the family music group the Brothers.	Ar	Al	artists, songwriter, record, producer
	doswell is acity in virginia? Yep, it's a city in Virginia. It is a small, unincorporated town that's doesn't have too much,but it is located close to Richmond. You can find out all about King's Dominion on their website about to host King's Fest, a large Christian music concert featuring artists like the David Crowder Band.	M	C	concert, band, music, artists

Table 8: Examples of errors by the constructed lexicons. N: None, E: Electronic, O: Office, Ar: Artist, Al: Album, M: Music, C: Culture, S: Society, T: Transportation, At: Athlete

One major weakness of the bootstrapping is that if the initially trained models have low performance, then it can lead to accumulated errors along the process. Several attempts (Abney 2008; Sogaard 2010) have been made to overcome the limitations. Recently, Jo et al. (2019) propose a variation of self-training framework for semi-supervised text classification. The method stems from the hypothesis that a classifier with pretrained word embedding always outperforms the same classifier with randomly initialized word embedding. Our approach and delta-training are based on a self-training framework. Therefore, we use delta-training and self-training as baselines to verify the excellence of our framework.

Adversarial Training Algorithms

The adversarial training is a technique of improving model performance by augmenting adversarial examples in the training process. Miyato et al. (2017) proposed a virtual adversarial training approach to smooth the output distributions of the neural networks on straight-forward classification tasks. They extended adversarial and virtual adversarial training to the text domain by applying perturbations to the word embeddings. Gururangan et al. (2019) introduced a lightweight pretraining framework for effective text classification when data and computing resources are limited. They pretrained a uni-gram document model as a variational autoencoder (VAE) on unlabeled data and used it for the downstream classifier. They demonstrated the effectiveness of the model for limited resource settings, without the need for computationally demanding. The adversarial training is not related to our framework. However, the adversarial training has the same goal of solving the data-sparsity problem for text classification. Therefore, we use VAMPIRE, a state-of-the-art model of the adversarial training for the experimental comparison.

Pretraining Algorithms

The pretraining algorithms transfer knowledge from rich-resource pretraining task to the low downstream tasks. Language Models (LMs) such as OpenAI Transformer (Vaswani et al. 2017) and BERT (Devlin et al. 2019) have achieved state-of-the-art performance on many classification tasks. The methods show excellent performance even with small number of labeled data. However, LMs require significant computational resources to train at a high scale. Unlike the two previous algorithms, pretraining algorithms are more related to unsupervised learning than semi-supervised learning. The pretraining model applies to other semi-supervised methods to improve performance. One of the latest adversarial learning, VAMPIRE, utilizes the pretraining model and has improved performance. Since the pretraining model can also be applied to the bootstrapping algorithms, we use the representative LMs, BERT, as a baseline, and show the improvement of performance when BERT is integrated into our framework, SALNet.

Conclusions and Future Work

We propose a simple, yet effective semi-supervised bootstrap learning framework for few-shot text classification, which takes full advantage of both only a small number of labeled data and a large number of unlabeled data. Our framework generates a lexicon using the attention mechanism, and we use the constructed lexicons for pseudo-labeling. The lexicon can select the correct label among the predictions of the classifier and correctly predict the unlabeled data that a model incorrectly predicts. Extensive experimental results demonstrate that our method achieves superior performance to the previous state-of-the-art methods.

We plan to develop methods that generate lexicons using the pretrained language model such as XLNet (Yang et al. 2019) and ALBERT (Lan et al. 2020). We also plan to study assigning weights to crucial words to enhance lexicon performance.

Acknowledgments

This research was supported by the NRF grant funded by MIST (NRF-2020R1A4A3079947) and the AI Graduate

School Program (2020-0-01361). Han is a corresponding author.

References

Abney, S. 2008. Semisupervised Learning for Computational Linguistics. *Computational linguistics* 34: 449–452.

and. Binbin Chen, E. F.; and Bernstein, M. S. 2016. Empath: Understanding Topic Signals in Large-Scale Text. In *Proceedings of the International Conference on Human Factors in Computing Sysmtes, CHI 2016*, 4647–4657.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate 3–12.

Bandhakavi, A.; Wiratunga, N.; Padmanabhan, D.; and Massie, S. 2017. Lexicon based feature extraction for emotion text classification. *Pattern Recognition Letters* 93: 133–142.

Blum, A.; and Mitchell, T. M. 1998. Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the International Conference on Computational Learning Theory, COLT 1998*, 92–100.

Chang, M.; Ratinov, L.; Roth, D.; and Srikumar, V. 2008. Importance of Semantic Representation: Dataless Classification. In *Proceedings of the International Conference on Artificial Intelligence, AAAI 2008*, 830–835.

Cho, K.; van Merrienboer, B.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, 1724–1734.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the International Conference of the North American Chapter of the Association for Computational Linguistics, NAACL 2019*, 4171–4186.

Feng, J.; Gong, C.; Li, X.; and Lau, R. Y. K. 2018. Automatic Approach of Sentiment Lexicon Generation for Mobile Shopping Reviews. *Wireless Communication Mobile Computing* 2018: 1–13.

Gururangan, S.; Dang, T.; Card, D.; and Smith, N. A. 2019. Variational Pretraining for Semi-supervised Text Classification. In *Proceedings of the International Conference of the Association for Computational Linguistics, ACL 2019*, 5880–5894.

Hailong, Z.; Wenyan, G.; and Bo, J. 2014. Machine Learning and Lexicon Based Methods for Sentiment Classification: A Survey. In *Proceedings of the International Conference on Web Information System and Application, WISA 2014*, 262–265.

Jo, H.; and Cinarel, C. 2019. Delta-training: Simple Semi-Supervised Text Classification using Pretrained Word Embeddings. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing, EMNLP 2019*, 3456–3461.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, 1746–1751.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization 22–30.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*.

Lee, H.; Lee, H.; Park, J.; and Han, Y. 2018. An abusive text detection system based on enhanced abusive and non-abusive word lists. *Decision Support Systems* 113: 22–31.

Lei, Z.; Riddhiman, G.; Mohamed, D.; Meichun, H.; and Bing, L. 2011. Combining Lexicon-based and Learning-based methods for twitter sentiment analysis. In *HP Laboratories, Technical Report*, 89–89.

Li, X.; and Liu, B. 2003. Learning to Classify Texts Using Positive and Unlabeled Data. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI 2003*, 587–594.

Lu, B.; and Tsou, B. K. 2010. Combining a large sentiment lexicon and machine learning for subjectivity classification. In *Proceedings of the International Conference on Machine Learning and Cybernetics, ICMLC 2010*, 3311–3316.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, 1412–1421.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the International Conference on Association for Computational Linguistics, ACL 2011*, 142–150.

Mendes, P. N.; Jakob, M.; and Bizer, C. 2012. DBpedia: A Multilingual Cross-domain Knowledge Base. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2012*, 1813–1817.

Miyato, T.; Dai, A. M.; and Goodfellow, I. J. 2017. Adversarial Training Methods for Semi-Supervised Text Classification. In *Proceedings of the International Conference on Learning Representations, ICLR 2017*, 13–18.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, 1532–1543.

Rosenberg, C.; Hebert, M.; and Schneiderman, H. 2005. Semi-Supervised Self-Training of Object Detection Models. In *Proceedings of the International Conference on Motion and Video Computing, MOTION 2005*, 29–36.

Ruder, S.; and Plank, B. 2018. Strong Baselines for Neural Semi-Supervised Learning under Domain Shift. In *Proceedings of the International Conference on Association for Computational Linguistics, ACL 2018*, 1044–1054.

Sogaard, A. 2010. Simple Semi-Supervised Training of Part-Of-Speech Taggers. In *Proceedings of the International Conference on Association for Computational Linguistics, ACL 2010*, 205–208.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, 3104–3112.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 6000–6010.

Wang, F.; and Zhang, C. 2007. Robust self-tuning semi-supervised learning. *Neurocomputing* 70: 2931–2939.

Wang, Y.; Huang, M.; Zhu, X.; and Zhao, L. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*, 606–615.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A. C.; Salakhutdinov, R.; Zemel, R. S.; and Bengio, Y. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the International Conference on Machine Learning, ICML 2015*, 2048–2057.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J. G.; Salakhutdinov, R.; and Le, Q. V. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Proceedings of the International Conference on Neural Information Processing Systems, NIPS 2019*, 5754–5764.

Yarowsky, D. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the International Conference on the Association for Computational Linguistics, ACL 1995*, 189–196.

Yaslan, Y.; and Cataltepe, Z. 2010. Co-training with relevant random subspaces. *Neurocomputing* 73: 1652–1661.

Zhang, X.; Zhao, J. J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. In *Proceedings of the International Conference on Neural Information Processing Systems, NIPS 2015*, 649–657.

Zhu, X. 2005. Semi-Supervised Learning Literature Survey. *Technical Report 1530* .

Zhu, X.; Ghahramani, Z.; and Lafferty, J. D. 2003. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In *Proceedings of the International Conference on Machine Learning, ICML 2003*, 912–919.