SAT: Improving Semi-Supervised Text Classification with Simple Instance-Adaptive Self-Training

Hui Chen **Wei Han** **Soujanya Poria**
Singapore University of Technology and Design
{hui_chen, wei_han}@mymail.sutd.edu.sg
sporia@sutd.edu.sg

Abstract

Self-training methods have been explored in recent years and have exhibited great performance in improving semi-supervised learning. This work presents a **S**imple instance-**A**daptive self-**T**raining method (SAT) for semi-supervised text classification. SAT first generates two augmented views for each unlabeled data and then trains a meta-learner to automatically identify the relative strength of augmentations based on the similarity between the original view and the augmented views. The weakly-augmented view is fed to the model to produce a pseudo-label and the strongly-augmented view is used to train the model to predict the same pseudo-label. We conducted extensive experiments and analyses on three text classification datasets and found that with varying sizes of labeled training data, SAT consistently shows competitive performance compared to existing semi-supervised learning methods. Our code can be found at https://github.com/declare-lab/SAT.git.

1 Introduction

Pretrained language models have achieved extremely good performance in a wide range of natural language understanding tasks (Devlin et al., 2019). However, such performance often has a strong dependence on large-scale high-quality supervision. Since labeled linguistic data needs large amounts of time, money, and expertise to obtain, improving models' performance in few-shot scenarios (i.e., there are only a few training examples per class) has become a challenging research topic.

Semi-supervised learning in NLP has received increasing attention in improving performance in few-shot scenarios, where both labeled data and unlabeled data are utilized (Berthelot et al., 2019b; Sohn et al., 2020; Li et al., 2021). Recently, several self-training methods have been explored to obtain task-specific information in unlabeled data. UDA (Xie et al., 2020) applied data augmentations to unlabeled data and proposed an unsupervised consistency loss that minimizes the divergence between different unlabeled augmented views. To give self-training more supervision, MixText (Chen et al., 2020a; Berthelot et al., 2019b) employed Mixup (Zhang et al., 2018; Chen et al., 2022) to learn an intermediate representation of labeled and unlabeled data. Both UDA and MixText utilized consistency regularization and confirmed that such regularization exhibits outstanding performance in semi-supervised learning. To simplify the consistency regularization process, FixMatch (Sohn et al., 2020) classified two unlabeled augmented views into a weak view and a strong view, and then minimized the divergence between the probability distribution of the strong view and the pseudo label of the weak view. However, in NLP, it is hard to distinguish the relative strength of augmented text by observation, and randomly assigning an augmentation strength will limit the performance of FixMatch on text.

To tackle this problem in FixMatch, our paper introduces an instance-adaptive self-training method SAT, where we propose two criteria based on a classifier and a scorer to automatically identify the relative strength of augmentations on text. Our main contributions are:

- First, we apply popular data augmentation techniques to generate different views of unlabeled data and design two novel criteria to calculate the similarity between the original view and the augmented view of unlabeled data in FixMatch, boosting its performance on text.

- We then conduct empirical experiments and analyses on three few-shot text classification datasets. Experimental results confirm the efficacy of our SAT method.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 6141–6146
December 7-11, 2022

2 Method

2.1 Problem Setting

In this work, we learn a model to map an input $x \in \mathcal{X}$ onto a label $y \in \mathcal{Y}$ in text classification tasks. In semi-supervised learning, we use both labeled examples and unlabeled examples during training. Let $\mathcal{X} = \{(x_b, y_b) : b \in (1, ..., B)\}$ be a batch of B labeled examples, where x_b are the training examples and y_b are labels. Let $\mathcal{U} = \{u_b : b \in (1, ..., \mu B)\}$ be a batch of μB unlabeled examples, where μ is a hyperparameter which determines the relative sizes of $\mathcal{X}$ and $\mathcal{U}$.

2.2 SAT

The entire process of SAT is illustrated in Algorithm 1. Similar to common semi-supervised learning methods, our approach consists of a supervised part and an unsupervised part. Our supervised part minimizes the cross-entropy loss between the labeled data and their targets. Our unsupervised part first generates two unlabeled augmented views, then applies an augmentation choice network to determine the relative augmentation strength, and finally calculates a consistency loss between the probability distribution of the strongly-augmented view and the pseudo label of the weakly-augmented view. Since the relative augmentation strength in our SAT method has no direct correlation to the augmentation techniques, our semi-supervised learning process can be more adaptive to the training data, compared to FixMatch.

The augmentation choice network is trained by the labeled data and we design two criteria to train it where (1) one is based on a **classifier** and (2) the other is based on a **scorer**. Line 2 to Line 7 in Algorithm 1 shows how we train the augmentation choice network. For each labeled data, we first calculate the similarity between the original data and its augmented variants, respectively, and then rank the augmented samples according to the similarity scores. In our classifier-based criterion, we employ a **cross-entropy loss** to measure the distance, while in our scorer-based criterion, we calculate the **cosine similarity**. Afterward, we define the one with a higher similarity score as the weakly-augmented sample and use it to train the augmentation choice network. For our classifier-based method, we apply a **cross-entropy loss** as the training objective. For our scorer-based method, we use a **contrastive loss** (Chen et al., 2020b) to update the network. Finally, the trained augmentation choice network is used to automatically identify the augmentation strength in unlabeled data.

Algorithm 1: SAT: Simple Instance-Adaptive Self-Training

Input: $\mathcal{D}^{train} = \{\mathcal{X}, \mathcal{U}\}$ where $\mathcal{X} = \{(x_b, y_b) : b \in (1, ..., B)\}$ and $\mathcal{U} = \{u_b : b \in (1, ..., \mu B)\}$; augmentation methods α_1, α_2; main network $f(;\theta)$ with parameters θ and its probability distribution p; augmentation choice network $G(;\theta_G)$ with parameters θ_G; criteria $\mathcal{C}$, Γ; cross-entropy loss H; unlabeled loss weight λ_u; confidence threshold τ; learning rates β, η

Output: Updated network weights θ

`// Calculate supervised loss`

1 $l_s = \frac{1}{B} \sum_{b=1}^{B} H(y_b, p(y|x_b))$

2 **for** $(x_b, y_b) \in \mathcal{X}$ **do**

3 $\quad i_1^b, i_2^b = \mathcal{C}(\alpha_1(x_b), x_b, y_b), \mathcal{C}(\alpha_2(x_b), x_b, y_b)$

4 $\quad i_w^b, i_s^b = \text{Descending}(i_1^b, i_2^b)$

5 **end**

`// Update the augmentation choice network`

6 $l_{aug\_choice} = \frac{1}{B} \sum_{b=1}^{B} \Gamma(x_b, \alpha_1(x_b), \alpha_2(x_b), i_w^b)$

7 $\theta_G = \theta_G - \beta \nabla l_{aug\_choice}$

8 **for** *each* $u_b \in \mathcal{U}$ **do**

9 $\quad \hat{i}_w^b, \hat{i}_s^b = G(u_b, \alpha_1(u_b), \alpha_2(u_b); \theta_G)$

10 **end**

`// Calculate unsupervised loss`

11 $l_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbf{1}\{\max(p(y|\alpha_{\hat{i}_w^b}(u_b))) > \tau\} H(\text{argmax}(p(y|\alpha_{\hat{i}_w^b}(u_b))), p(y|\alpha_{\hat{i}_s^b}(u_b)))$

`// Total loss: add up supervised loss and unsupervised loss`

12 $l_{total} = l_s + \lambda_u l_u$

`// Update the main network`

13 $\theta = \theta - \eta \nabla l_{total}$

3 Experimental Setup

We conducted empirical experiments to compare our approach with a couple of existing semi-supervised learning methods on a variety of text classification benchmark datasets.

Methods	AG News ($c = 4$)		Yahoo! ($c = 10$)		IMDB ($c = 2$)			
	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)	Average	Δ
BERT (Devlin et al., 2019)	$69.18_{3.7}$	$68.27_{3.5}$	$58.11_{1.6}$	$57.38_{1.9}$	$63.16_{1.4}$	$62.93_{1.6}$	63.17	-
UDA (Xie et al., 2020)	$76.69_{3.2}$	$76.51_{3.0}$	$59.32_{2.0}$	$58.47_{2.3}$	$64.88_{1.7}$	$64.57_{1.5}$	66.74	+3.57
MixText (Chen et al., 2020a)	$78.07_{2.8}$	$77.23_{3.5}$	$59.93_{1.9}$	$59.24_{1.8}$	$65.22_{1.1}$	$65.78_{1.2}$	67.58	+4.41
FixMatch (Sohn et al., 2020)	$80.22_{2.4}$	$79.98_{2.1}$	$60.17_{1.7}$	$59.86_{1.5}$	$64.52_{1.6}$	$64.31_{1.4}$	68.18	+5.01
SAT: classifier-based (Ours)	$\mathbf{86.38_{2.0}}$	$\mathbf{86.29_{2.3}}$	$\mathbf{61.51_{1.8}}$	$\mathbf{61.09_{1.6}}$	$65.43_{1.2}$	$64.28_{1.4}$	70.83	+7.66
SAT: scorer-based (Ours)	$85.43_{1.2}$	$85.30_{1.5}$	$61.33_{1.5}$	$60.96_{1.4}$	$\mathbf{68.96_{1.7}}$	$\mathbf{68.92_{1.6}}$	**71.82**	**+8.65**

Table 1: Accuracy (%) and Macro F1 (%) on three diverse text classification tasks for BERT, UDA, MixText, FixMatch, and our SAT method. c: number of classes; Δ: improvement compared with BERT.

3.1 Datasets and Metrics

We considered three diverse few-shot text classification scenarios in our experiments: AG News which categorizes more than 1 million news articles into 4 categories — World, Sports, Business, and Sci/Tech (Zhang et al., 2015), and Yahoo! Answers which classifies question-answer pairs into 10 clusters where all question-answer pairs in a cluster ask about the same thing (Zhang et al., 2015), and IMDB which predicts sentiment of movie reviews to be positive or negative (Maas et al., 2011).

We used the original test set as our test set and randomly sampled from the training set to construct the training unlabeled set and development set. To balance the class distribution in the experiments, we randomly sampled N_c samples per class to be used for training. In AG News and IMDB, $N_c = 10$. In the Yahoo! dataset, we set N_c as 20 to ensure consistent results. For all experiments, we used accuracy (%) and macro F1 score (%) as the evaluation metrics.

3.2 Baselines

To test the effectiveness of our approach, we compared it with several popular self-training methods: UDA (Xie et al., 2020), MixText (Chen et al., 2020a), FixMatch (Sohn et al., 2020). To ensure fair comparisons, we used the same augmentation techniques[1], i.e., back-translation (Sennrich et al., 2016b) and synonym replacement (Wei and Zou, 2019), in all baselines. For back-translation augmentation, we used German as the middle language. For synonym replacement augmentation, the substitution percentage is 30%. Also, we used the same BERT-based-uncased model, unlabeled data size, and batch size in all methods.

[1]The implementation is based on https://github.com/makcedward/nlpaug.

4 Results

4.1 Main Results

This section compares our SAT method with BERT, UDA, MixText, and FixMatch on three text classification datasets. Our main results are shown in Table 1. Results are averaged over five different runs.

We observed that BERT achieves a mean score across our three datasets of 63.17%, and UDA improves performance noticeably by +3.57%. MixText and FixMatch show better performance in improving the BERT baseline, where the mean score increases are +4.41% and +5.01%, respectively. Our classifier-based SAT method, which applies a cross-entropy loss to measure the similarity between the original data and its augmented variants, achieves a mean score of 70.83%, outperforming FixMatch by +2.65%. The scorer-based SAT, which employs cosine distance to measure the similarity further improves about 1% over the classifier-based SAT.

From these results, we obtained the following findings. Firstly, strategically selecting the strongly and weakly augmented samples in self-training can effectively boost performance. Compared to FixMatch, we improved the performance by adding a lightweight meta-learner to automatically identify augmentation strengths, without sacrificing much training time. Secondly, when tackling datasets with few training examples, using cosine distance to measure the similarity between two examples and using contrastive loss to train the meta-learner shows better and more robust performance.

4.2 Ablation: Size of Labeled Data

This ablation investigates how our semi-supervised learning method performs for different sizes of labeled data. Fig. 1 compares the average scores of accuracy and macro F1 of FixMatch and our meth-

ods. First, we observed that our methods can consistently outperform FixMatch with varying data sizes. This indicates that strategically selecting the strongly and weakly augmented samples contributes to the final performance in self-training. Second, when N_c increases from 3 to 10, the scores of the three methods increase accordingly. When N_c becomes 20, the performance of FixMatch and the classifier-based SAT drops, which is consistent with prior findings on the diminished effect of data augmentation for larger datasets (Xie et al., 2020; Andreas, 2020). However, the scorer-based SAT does not show an obvious performance decrease, showing that in few-shot datasets, the scorer-based method is more robust than the classifier-based method.

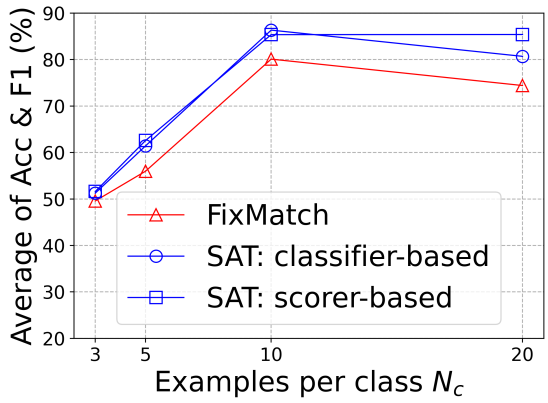

Figure 1: Average scores of accuracy and macro F1 from FixMatch and our method for different sizes of labeled data on the AG News dataset.

4.3 Ablation: Various Augmentation Techniques

To evaluate the effect of various augmentation techniques in our method, we performed experiments using different text augmentation technique combinations in our method. These augmentation techniques are widely-used: **(1) Synonym Replacement (SR)** substitutes words with WordNet synonyms (Wei and Zou, 2019); **(2) Pervasive Dropout (PD)** applies a word-level dropout with a probability of 0.1 on text (Sennrich et al., 2016a); **(3) Random Insertion (RI)** randomly inserts words in a sentence (Wei and Zou, 2019); **(4) Back-translation (BT)** translates text into another language and then back into the original language (Sennrich et al., 2016b).

Fig. 2 compares average scores of accuracy and macro F1 from different augmentation techniques in our method on the AG News dataset. The combination of back-translation and synonym replacement improves performance the best, perhaps because they maintain a good balance between injecting proper perturbation noise and preserving the original meaning of the text.

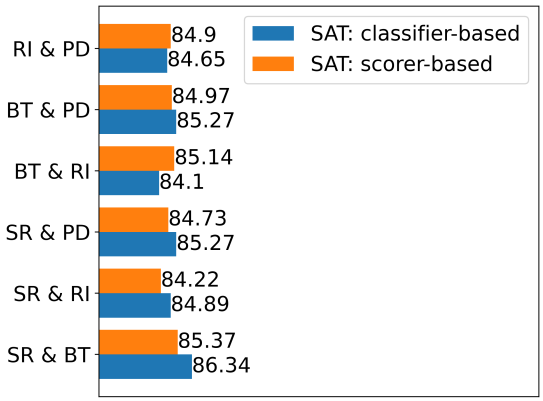

Figure 2: Average scores of accuracy and macro F1 from different augmentation technique combinations in our method on the AG News dataset, where $N_c = 10$.

5 Related Work

Our work combines data augmentation and consistency regularization to improve semi-supervised learning and is inspired by prior work in these areas. Recently, several papers have proposed reinforcement learning policies (Cubuk et al., 2019; Ho et al., 2019) and curriculum learning strategies (Wei et al., 2021; Zhang et al., 2021) to automatically augment data. Also, a couple of consistency regularization methods are introduced to simplify the semi-supervised learning process (Berthelot et al., 2019a; Sohn et al., 2020) as well as to boost performance in domain adaptation scenarios (Berthelot et al., 2021) to improve semi-supervised learning. As far as we know, our work is the first to apply a meta-learner to automatically determine the augmentation strength in consistency regularization in semi-supervised text classification.

6 Conclusion

In closing, this paper has proposed an instance-adaptive self-training method SAT to boost performance in semi-supervised text classification. Inspired by FixMatch, SAT combines data augmentation and consistency regularization and designs a novel meta-learner to automatically determine the relative strength of augmentations. Empirical

experiments and ablation studies confirm SAT is simple yet effective in improving semi-supervised learning.

Limitations

Our proposed method has two limitations. First, in our experiments, we found the semi-supervised process is easy to be influenced by unlabeled data size. During training, we adjusted the size of unlabeled data in each batch by adjusting the μ value, as mentioned in Section 2.1. It will be a future direction that we design some strategies to automatically learn the μ value. Second, this work only considered the situation where there are only two augmentations in consistency regularization. As our SAT method can automatically rank the augmentation strengths, our future work is to extend SAT to regularize more than two augmentations.

Ethical Considerations

Augmentation techniques discussed in Section 4.3 should be used with care since they might generate data that do not align with the original meaning.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. This research is supported by the Ministry of Education, Singapore, under its AcRF Tier-2 grant (Project no. T2MOE2008, and Grantor reference no. MOET2EP20220-0017). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not reflect the views of the Ministry of Education, Singapore.

References

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566.

David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. 2019a. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019b. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32.

David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alexey Kurakin. 2021. Adamatch: A unified approach to semi-supervised learning and domain adaptation. In *International Conference on Learning Representations*.

Hui Chen, Wei Han, Diyi Yang, and Soujanya Poria. 2022. Doublemix: Simple interpolation-based data augmentation for text classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4622–4632.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020a. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. 2019. Population based augmentation: Efficient learning of augmentation policy schedules. In *International Conference on Machine Learning*, pages 2731–2741. PMLR.

Changchun Li, Ximing Li, and Jihong Ouyang. 2021. Semi-supervised text classification with balanced deep representation distributions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5044–5053.

Andrew 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 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608.

Jason Wei, Chengyu Huang, Soroush Vosoughi, Yu Cheng, and Shiqi Xu. 2021. Few-shot text classification with triplet networks, data augmentation, and curriculum learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5493–5500.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388.

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

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

A Dataset Statistics

The dataset statistics and split information are presented in Table 2.

Datasets	# Unlabeled	# Dev	# Test
AG News	5000	2000	1900
Yahoo!	5000	2000	6000
IMDB	5000	1000	12500

Table 2: Dataset statistics and data splits. The number of unlabeled data, dev data and test data in the table means the number of data per class.

B Implementation Details

We performed a grid search for hyperparameters: $\eta_{main} \in \{5e-5, 1e-3\}, \eta_{bert} \in \{1e-5, 5e-5\}$, β is fixed at $1e^{-4}$, $\mu \in \{3, 4, 5, 8, 10, 20\}$, $\tau \in \{0.90, 0.95, 0.99\}$, and batch size is fixed at 32. We tuned our model on a single NVIDIA RTX 8000 GPU. We ran each experiment for 50 epochs with a patience of 15 or 10 for early stopping.