Sentence-aware Adversarial Meta-Learning for Few-Shot Text Classification

Suhe Wang , Xiaoyuan Liu , Bo Liu , Diwen Dong
College of Computer
National University of Defense Technology
Changsha, China
wangsuhe16,xyliu,KyLe.liu,ddw_bak@nudt.edu.cn

Abstract

Meta-learning has emerged as an effective approach for few-shot text classification. However, current studies fail to realize the importance of the semantic interaction between sentence features and neglect to enhance the generalization ability of the model to new tasks. In this paper, we integrate an adversarial network architecture into the meta-learning system and leverage cost-effective modules to build a novel few-shot classification framework named SaAML. Significantly, our approach can exploit the temporal convolutional network to encourage more discriminative representation learning and explore the attention mechanism to promote more comprehensive feature expression, thus resulting in better adaptation for new classes. Through a series of experiments on four benchmark datasets, we demonstrate that our new framework acquires considerable superiority over state-of-the-art methods in all datasets, increasing the performance of 1-shot classification and 5-shot classification by 7.15% and 2.89%, respectively.

1 Introduction

Deep learning usually relies on a vast amount of labeled examples to accomplish tasks. However, this requirement is problematic for few-shot text classification when only a few examples are available in novel classes, which leads to poor model generalization for new tasks. Motivated by the fact that humans can quickly recognize new knowledge after learning a few examples, few-shot learning is becoming a hot research topic.

Early studies exploit data augmentation and regularization procedures (Salamon and Bello, 2017) to deal with the overfitting due to data sparseness. More recent research mainly falls into two approaches: (1) transfer-learning based methods (Pan et al., 2019; Gupta et al., 2020), which transfer and propagate knowledge attained from the source domain to classify unseen examples in the target domain. (2) meta-learning based methods (Tong, 2019; Sun et al., 2019), which learn knowledge by repeating lots of meta-tasks in a training episode manner and leverage knowledge extracted to briskly predict new samples during meta-testing. Specifically, Bao et al. (Bao et al., 2019) incorporated distributional signatures of words into the meta-learning framework to make impressive performance. Han et al. (Han et al., 2021) first proposed to introduce an adversarial domain adaptation network to strengthen the generalization ability of the meta-learning system.

Despite the remarkable progress of few-shot classification approaches (Geng et al., 2019; Bao et al., 2019; Han et al., 2021), most existing meta-learning models still suffer tough challenges: example diversity. Even examples in the same class have various representations, bringing about the difficulty of extracting generic features based on a few training examples and the urgent demand for strong adaptability of the model to new tasks.

In this paper, we propose a straightforward but remarkably powerful framework to cope with the above challenges. We deploy an adversarial network architecture to train the meta-learning system and create a novel framework Sentence-aware Adversarial Meta-Learner (SaAML). Specifically, we employ the temporal convolutional network (TCN) (Bai et al., 2018) and the multilayer perceptron network (MLP) to build the generator and discriminator, respectively. The model can grasp sentences' inherent semantic information through the adversarial training of the generator and the discriminator. Then we artfully fuse the word embeddings and the features of the generator to construct high-quality discriminative sentence features. Moreover, we further build a feature enhancer (FE) to leverage the multi-head attention mechanism to fine-tune the features of support examples and query examples, aiming to create more compatible feature representations. Our research methodically develops how to

Proceedings of the 29th International Conference on Computational Linguistics, pages 4844–4852
October 12–17, 2022.

better exploit the adversarial network architecture and sentence-aware interaction knowledge to boost few-shot text classification performance. The main innovations of our research are as follows:

1) We analyze the limitation to the performance of current meta-learning approaches and establish a novel system SaAML based on the adversarial network architecture. And we combine it with various meta-classification techniques to handle the few-shot classification dilemma.

2) We build the meta-learning system with more cost-effective modules, .e.g., the temporal convolutional network (TCN), the semantic extractor (SE), and the feature enhancer (FE). Then we jointly train the whole model in an end-to-end fashion.

3) To evaluate the effectiveness and robustness of the proposed model, we conduct massive comparison experiments and ablation studies on four datasets. The results indicate that our new framework SaAML outperforms state-of-the-art approaches with nearly 5.02% performance improvement, and the proposed enhancement modules are more beneficial and flexible.

2 Related Work

Few-shot text classification is a crucial application scenario of few-shot learning in natural language processing (NLP), which has obtained increasing attention and research. Transfer learning (Gupta et al., 2020), as a feasible approach to tackle few-shot text classification, intends to make the knowledge learned from the source domain more compatible with the target domain and reduce the shift between different domains. For example, Tzeng et al. (Tzeng et al., 2017) leverages the adversarial domain adaptation framework to bridge the domain gaps without example constraints. Moreover, with the rapid development of pre-trained language models, it also exhibits excellent performance by fine-tuning the representations of training examples, such as BERT (Devlin et al., 2018) and GPT-3 (Brown et al., 2020).

As the dominant program in few-shot text classification, meta-learning (Schmidhuber, 1987) systems develop rapidly and achieve great success. Meta-learning approaches are mainly divided into two categories: (1) Optimization-based methods, which resort to "learning to fine-tune" strategy to train the model. For example, (Finn et al., 2017; Lee et al., 2019; Rajeswaran et al., 2019) develop an optimization procedure of model parameter initialization to quickly obtain outstanding performance after a small amount of gradient update steps based on few-shot training examples.(2) Metric-based methods, which complete the classification task through a specific distance metric, .e.g., Matching Network (Vinyals et al., 2016) is calculated by the cosine similarity, while the Euclidean distance is the measure standard of Prototypical Network (Snell et al., 2017). Relational Network (Sung et al., 2018) and GNN Network (Yang et al., 2020) utilize convolutional neural networks and graph neural networks to learn metric functions dynamically. Recently, Bao et al. (Bao et al., 2019) argues that statistical information of sentences plays a vital role in classification tasks. Sun et al. (Sun et al., 2021) subtly combines data augmentation with meta-learning to generate more diverse samples, preventing model overfitting. Han et al. (Han et al., 2021) is the first to explore the adversarial domain adaptation network for the performance improvement of meta-learning framework.

3 Preliminary

The few-shot text classification is usually viewed as a N-way K-shot task. Firstly, the input of the model is a set with multiple labeled examples, including support set C_{train} and query set C_{test}, where C_{train} are disjoint from C_{test}. Then the meta-learner conducts the episode-based strategy (Vinyals et al., 2016) to train a classifier on C_{train}. Finally, the meta-learner accomplishes the goal to predict new examples of C_{test} with the classifier.

For the episode-based strategy, we randomly select N classes (N-way) from C_{train}, and then separately sample K examples (K-shot) as the support set S and P examples as the query set Q from each of these chosen classes, which can be denoted as:

$$\begin{aligned} S &= \{(X_i, Y_i)\}_{i=1}^{N\times K} \\ Q &= \{(X_j, Y_j)\}_{j=1}^{N\times P} \end{aligned} \quad (1)$$

where X is the input text sentence and Y is the corresponding label. It is worth noting that the same example sampling manner is implemented to build the support set S and query set Q during meta-training and meta-testing. We leverage the macro-averaged accuracy across all testing episodes to evaluate the performance of the meta-learner.

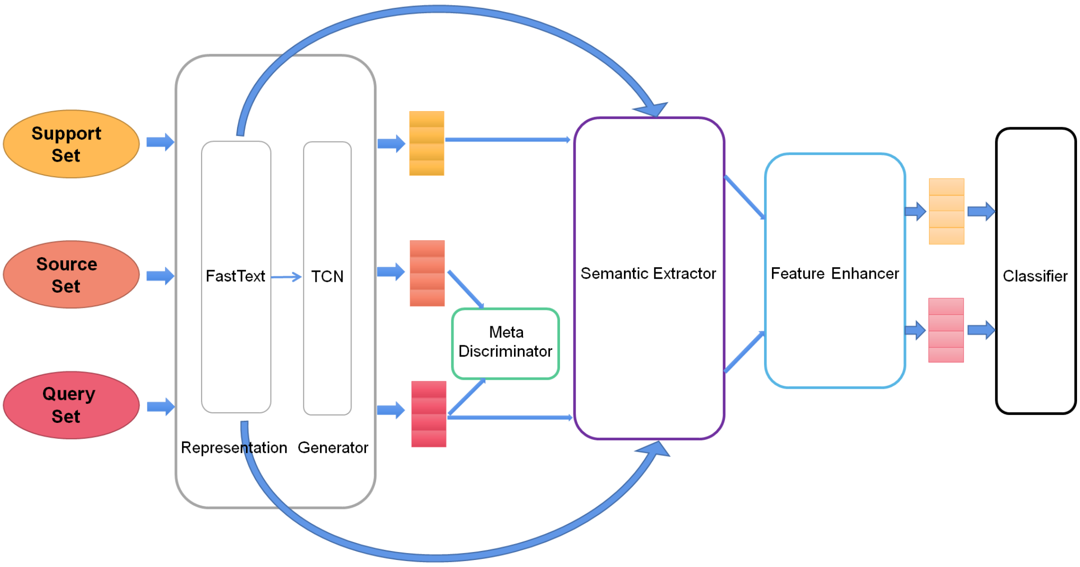


Figure 1: The overall framework of SaAML

4 Our Approach

Our proposed SaAML is broadly built to pursue the excellent performance of few-shot text classification. The overall architecture of SaAML is shown in Figure 1. First, we leverage the representation generator with TCN as the core block to capture comprehensive semantic information of sentences and exploit the meta discriminator to strengthen the learning ability of the model. Then, we contrive a feature enhancer to further fine-tune the example vectors with sentence-aware knowledge for more consistent representations. Finally, we jointly train all modules of SaAML with different schemes in an end-to-end manner.

4.1 Representation Generator

The representation generator consists of a word embedding encoder and a temporal convolutional network (TCN) (Bai et al., 2018), as shown in Figure 1.

The word embedding encoder converts each word into the embedding vector. We construct the embedding vector $\boldsymbol{v}$ with d dimensions via fastText (Joulin et al., 2016).

The goal of the temporal convolutional network (TCN) (Bai et al., 2018) is to acquire more transferable feature information. TCN architecture is very simple yet effective, covering some of the best techniques of current convolutional networks, such as 1D full convolution, causal convolution, and dilated convolution, as shown in Figure 2. To be specific, we exploit the 1D full convolution to make the model's output with the same length as the input. We adopt the causal convolution to ensure that the current result is only convolved with inputs from now to earlier, with no "leakage" of knowledge from the future to the past. Significantly, we also utilize the dilated convolution to expand the receptive field of the network and encourage the result of richer semantic information.

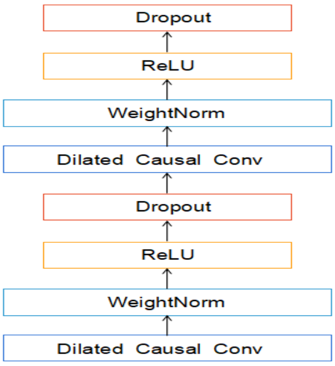
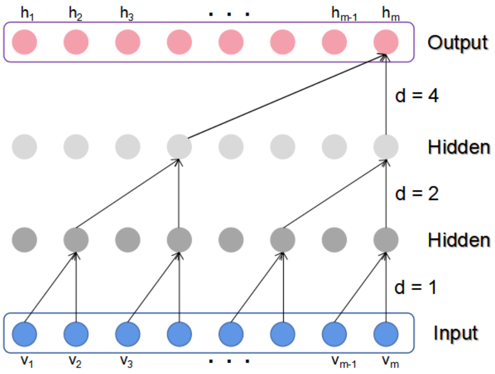


Figure 2: The convolutional layer structure of TCN.

Given that the input is a sequence of word vectors $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdot\cdot\cdot, \boldsymbol{v}_m]$, where m is the number of words in the sentence. The output of TCN is a matrix $\boldsymbol{h}_{d\times m} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \cdot\cdot\cdot, \boldsymbol{h}_m]$ with contextual embeddings, as shown in Figure 3.The matrix $\boldsymbol{h}_{d\times m}$ is then converted into the sentence vector $\boldsymbol{h}^g$ by a linear layer with $softmax$ function.

Furthermore, the representation generator also competes against the meta discriminator as much as possible, so that the discriminator can not determine whether the samples are from the source



h_1 h_2 h_3 · · · h_{m-1} h_m
Output
d = 4
Hidden
d = 2
Hidden
d = 1
Input
v_1 v_2 v_3 · · · v_{m-1} v_m

Figure 3: The TCN with dilation factors d = 1, 2, 4 and filter kernel k = 2.

domain or the target domain. Only in this way can more comprehensive features be created for better classification performance, which is the core motivation in building the representation generator and meta discriminator.

4.2 Meta Discriminator

We consider the examples in the support set and query set as the target domain, and the remaining examples as the source domain. Definitely, we sample a subset of the same size as the query set from the source domain as the source set. The discriminator comprises a three-layer feed-forward neural network and a $softmax$ function. The calculation process is as follows:

$$\hat{Y} = softmax(MLP(\boldsymbol{h}^g)) \tag{2}$$

where $\boldsymbol{h}^g$ denotes the sentence vector of each example from the representation generator. $\hat{Y} \in \{0, 1\}$ represents whether the example is from the query set or the source set.

4.3 Semantic Extractor

The semantic extractor fuses the word embedding from fastText and the representation vector from the generator to create the more comprehensive feature representation $\boldsymbol{w}$ for each example, which can be expressed as:

$$\boldsymbol{w} = \boldsymbol{V}_{d\times m} \cdot \boldsymbol{h}^g \tag{3}$$

where $\boldsymbol{V}_{d\times m} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdot\cdot\cdot, \boldsymbol{v}_m]$. $\boldsymbol{v}_m$ is the word embedding vector and m is the number of words in the sentence. $\boldsymbol{h}^g$ denotes the sentence vector generated by the representation generator.

4.4 Feature Enhancer

We construct the feature enhancer with the multi-head attention mechanism (Vaswani et al., 2017) and the MLP layer. The multi-head attention module consists of multiple self-attention units. Each self-attention unit has a powerful capability to catch valuable feature information about the input, and the knowledge learned by various self-attention units from their respective perspectives can be incorporated in a concatenation manner, as shown in Figure 4.

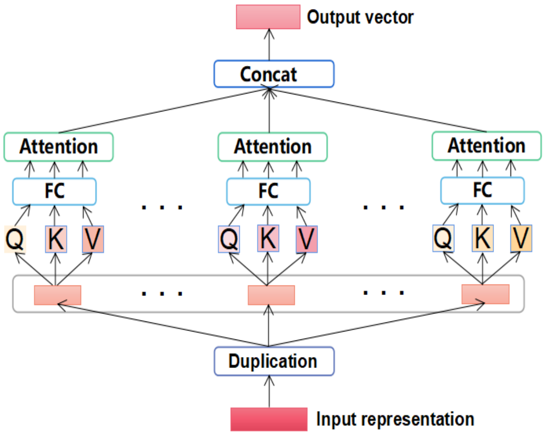


Figure 4: Details of the multi-head attention mechanism.

Thus, we employ the multi-head attention mechanism to effectively grasp the feature interaction between the representation sequences $\boldsymbol{w}^s$ of support examples and the representation sequences $\boldsymbol{w}^q$ of query examples. We then leverage the MLP layer with $GELU(\cdot)$ activation function (Hendrycks et al., 2020) to produce the final support vector sequences $\boldsymbol{z}^s$ and query vector sequences $\boldsymbol{z}^q$. The whole process above can be described as:

$$E_{sq} = [\boldsymbol{w}^s, \boldsymbol{w}^q] \tag{4}$$

$$H_{sq} = MHAttention(\,E_{sq}\,) \tag{5}$$

$$\boldsymbol{z}^s, \boldsymbol{z}^q = GELU(MLP(H_{sq})) \tag{6}$$

where $\boldsymbol{w}^s = [\boldsymbol{w}_1^s, \boldsymbol{w}_2^s, \cdots, \boldsymbol{w}_k^s]$ and $\boldsymbol{w}^q = [\boldsymbol{w}_1^q, \boldsymbol{w}_2^q, \boldsymbol{w}_3^q, \cdot\cdot\cdot, \boldsymbol{w}_p^q]$. The k and p represent the number of examples in the support set and query set, respectively. E_{sq} is the concatenation result of $\boldsymbol{w}^s$ and $\boldsymbol{w}^q$. $MHAttention(\cdot)$ is actually the main component of $Transformer$ model (Vaswani et al., 2017) and is used alone in our work to generate the self-attention output H_{sq}. The final support vector sequences $\boldsymbol{z}^s = [\boldsymbol{z}_1^s, \boldsymbol{z}_2^s, \cdot\cdot\cdot, \boldsymbol{z}_k^s]$ and query vector sequences $\boldsymbol{z}^q = [\boldsymbol{z}_1^q, \boldsymbol{z}_2^q, \cdot\cdot\cdot, \boldsymbol{z}_p^q]$.

The feature enhancer is very beneficial to boost the feature correlation between support examples and query examples and decrease the interference of noisy representations. With limited knowledge, the model can create the class-level representations that are more compatible with query examples and the query representations that are more consistent with support examples, and acquire fine generalization ability across various classification tasks rapidly.

4.5 Classifier

We adopt the prototypical network (Snell et al., 2017) with Euclidean distance as the classifier in our framework, which enables the model to easily solve the learning problem and efficiently accelerate training convergence.

Therefore, the classifier generates the class-level vector $\boldsymbol{C}_u$ for each class u based on support vectors $\boldsymbol{z}_i^s$ and measures the similarity probability, $\boldsymbol{D}_j$, between the query vector $\boldsymbol{z}_j^q$ and the class vector $\boldsymbol{C}_u$ through the Euclidean distance function $R(\cdot,\cdot)$.

$$\boldsymbol{C}_u = \frac{1}{K} \sum_{(\boldsymbol{z}_i^s, Y_i) \in S_u} \boldsymbol{z}_i^s \tag{7}$$

$$\boldsymbol{D}_j = \frac{e^{R(\boldsymbol{C}_u, \boldsymbol{z}_j^q)}}{\sum_{n=1}^{N} e^{R(\boldsymbol{C}_n, \boldsymbol{z}_j^q)}} \tag{8}$$

where $S_u \subset S$ is the subset corresponding to class u with the same label Y_i in the support set S. And K represents the number of examples in each class.

4.6 Loss Function

The modules of our framework SaAML are trained using different strategies, .e.g, we train the classifier from scratch for each episode, while the representation generator is optimized across all training episodes. For each training episode, we first employ the source set and query set to update the parameters of the meta discriminator. Next, we update the parameters of the representation generator and classifier over the support set and query set. The details of the loss function are introduced below.

The classifier loss consists of cross-entropy loss and difference loss. The difference loss aims to maximize the distance between different class vectors, making each class as directionally different as possible, which is defined as:

$$L_{DL} = \lambda \sum_{i \neq j} \parallel \boldsymbol{C}_i^T \boldsymbol{C}_j \parallel_F^2 \tag{9}$$

where $\| \cdot \|_F$ is the Frobenius norm, and λ is the hyperparameter and can be set to 10^{-3}.

Thus, the classifier loss can be represented as:

$$L^C = -\frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} Y_{n,k} \, log \, P_{n,k} + L_{DL} \tag{10}$$

where $P_{n,k}$ indicates the probability that the n-th query example is predicted to be the k-th label. $Y_{n,k}$ denotes the ground-truth label. The K and N are the number of categories and examples, respectively.

It is standard practice to apply the cross-entropy loss as the loss function for the discriminator.

$$\begin{aligned} L^D = -\frac{1}{2n} \sum_{k=1}^{n} [\, (Y_k^d \, log \, \hat{Y}_k \\ + (1 - Y_k^d) \, log \, (1 - \hat{Y}_k))] \end{aligned} \tag{11}$$

where Y_k^d and $\hat{Y}_k$ denotes the real class of the example and the prediction result of the discriminator, respectively. The n is the number of examples in the source set or query set.

As for the loss function of the generator, it can be regarded as the combination of the classifier loss function and the discriminator loss function, which are used to gain the final classification results and confuse the discriminator, respectively.

$$L^G = L^C - L^D \tag{12}$$

5 Experiment

We investigate the performance of our proposed SaAML against six existing well-established baselines through extensive experiments on four benchmark datasets. Furthermore, we also develop a series of ablation studies to further illustrate the effectiveness and robustness of SaAML.

5.1 Datasets

There are four few-shot text classification datasets in our work, as expressed in Table 1.

Table 1: Details of the four benchmark datasets.

Dataset	Avg. text length	samples per class	train/val/test classes
Amazon	140	1000	10 / 5 / 9
20 Newsgroups	340	941	8 / 5 / 7
HuffPost	11	900	20 / 5 / 16
Banking77	16	170	30 / 15 / 32

Amazon is a collection of 24 product categories (He and McAuley, 2016) with 142.8 million product reviews. Following the same approach

Table 2: Mean classification accuracy on Amazon, HuffPost, 20 Newsgroups, and Banking77 datasets.

	Amazon		HuffPost		20 Newsgroups		Banking77			
Method	5-way		5-way		5-way		10-way		15-way	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MAML	0.3965	0.4713	0.3572	0.4931	0.3384	0.4372	0.4691	0.6659	0.3672	0.5659
PN	0.3760	0.5214	0.3573	0.4474	0.3780	0.4535	0.4156	0.6975	0.3501	0.6393
IN	0.3491	0.4132	0.3874	0.4912	0.2876	0.3337	0.5291	0.6884	0.4553	0.6179
RRML	0.5022	0.7275	0.3610	0.4966	0.3760	0.5724	0.5256	0.8148	0.4694	0.7734
DS-RRML	0.6260	0.8112	0.4305	0.6350	0.5212	0.6830	0.6034	0.8373	0.5432	0.7900
MLADA	0.6842	0.8600	0.4502	0.6491	0.5961	0.7780	0.6055	0.8089	0.5513	0.7470
SaAML	**0.7147**	**0.8637**	**0.5126**	**0.6944**	**0.7079**	**0.8430**	**0.6860**	**0.8483**	**0.6235**	**0.8096**

(Bao et al., 2019), we select 1000 reviews from each category to establish the subset.

20 Newsgroups consists of about 20000 news sentences evenly partitioned on 20 different topics, which is from the news discussion boards (Lang, 1995).

HuffPost is extracted from the HuffPost articles between the year 2012 and 2018 (Misra, 2018), with news headlines in 41 categories.

Banking77 provides 77 classes of 13083 fine-grained intents from the banking domain, proposed by Casanueva et al. (Casanueva et al., 2020).

5.2 Baselines

Six existing few-shot learning baselines are adopted to compare with our proposed SaAML, which are briefly introduced as follows:

Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017) is an optimization-based approach to explicitly train model parameters such that the model can produce great generalization on new tasks after a few gradient update steps.

Prototypical Network (PN) (Snell et al., 2017) is a metric-based method that employs the feature average of support examples as the class vector (prototype).

Induction Network (IN) (Geng et al., 2019) constructs the class vector through the dynamic routing algorithm based on capsule network and leverages the relation module (Sung et al., 2018) to learn the measure function.

Ridge Regression Meta-Learner (RRML) (Bertinetto et al., 2018) exploits the ridge regression to obtain the class vector and develops proper regularization to reduce model overfitting and speed up model convergence.

Distributional Signature (DS) (Bao et al., 2019) considers that the distribution signature is very essential to catch more comprehensive feature representations. Therefore, this meta-learning framework achieves outstanding performance when combined with distribution signatures, where DS+RRML is the best method.

Meta-Learning Adversarial Domain Adaptation Network (MLADA) (Han et al., 2021) explores the adversarial network architecture to extract sentence features, improving the generalization and performance of meta-learning systems in various scenarios.

In our study, we adopt the pre-trained fastText (Joulin et al., 2016) as the embedding encoder in all methods. TCN includes a total of four layers. For each layer, the number of hidden units is 300 and kernel size is 2. The number of hidden units of MLP in the discriminator is 256 and 128, respectively. In the feature enhancer, the head number and the feature dimension of the attention mechanism are fixed as 6 and 300. For a fair comparison, 100, 100, and 1000 task episodes are randomly sampled individually in each training, validation, and testing epoch. Furthermore, we optimize model parameters through the *Adam* algorithm (Kingma and Ba, 2014) with a learning rate of $1e-5$ and use early stopping scheme when the performance on the validation set fails to increase within 20 epochs. We conduct all experiments on the NVIDIA Geforce GTX 3090 GPUs server.

5.3 Experimental Results

We can obtain several valuable observations from the comparison results of different baseline models, and the results are depicted in Table 2. (1) Overall, our model SaAML creates an average accuracy of 64.89% in 1-shot classification and 81.18% in 5-shot classification, substantially refreshing the best performance on all datasets. Notably, it attains a significant performance improvement of 5.73% over the state-of-the-art system MLADA

(Han et al., 2021). (2) We find that SaAML generally outperforms 5-shot classification on 1-shot classification, with an average accuracy improvement of 7.15% in 1-shot classification and 2.89% in 5-shot classification. This is understandable since the feature information available in low-shot scenes is extremely deficient, especially for 1-shot classification, whereas SaAML is very competent in leveraging the combination of the adversarial network architecture and the feature enhancer to grasp semantic information from a few examples, so that it can address this challenge rapidly. (3) We also notice that SaAML considerably improves the accuracy of 20 newsgroups by nearly 8.84%, which is better than other datasets in terms of performance improvement. It clearly illustrates that the model has a powerful feature learning ability and is more compatible with long texts with rich information. Besides, we have to acknowledge that the performance improvement of SaAML on Amazon is not apparent because of its affluent examples and diverse expressions. Note that the experimental results have 95% confidence intervals with variances below 0.01.

5.4 Ablation Study

We perform massive experiments to explore the effectiveness and adaptability of various components, .i.e., adversarial network (AN), temporal convolutional network (TCN), semantic extractor (SE), feature enhancer (FE) and difference loss (DL). The ablation results are written in Table 3.

Firstly, we remove the setup of the adversarial network (AN) architecture, which involves the meta-discriminator and the source set. That is, the discriminator no longer strengthens the features via the adversarial training. In this way, the classification performance of the model is significantly weakened, which justifies the necessity and effectiveness of adopting the adversarial network.

Secondly, we utilize the BiLSTM instead of the TCN (Bai et al., 2018) to build the representation generator. It is observed that the performance of the model with TCN is superior to the model with BiLSTM. This is because BiLSTM saturates at a very early training stage due to optimization difficulties, while TCN can capture the longer-distance dependence of feature information.

Thirdly, we examine the role of the semantic extractor (SE) and feature enhancer (FE) in performance improvement. Obviously, the classification accuracy of the model without the feature enhancer drops by 12.30%. The model without the semantic extractor also leads to the performance drop of 7.48%. It robustly demonstrates that the semantic extractor can gain the richer semantics of the examples via a fusion manner, and the feature enhancer can grasp the interaction knowledge between support examples and query examples.

Finally, we investigate the importance of difference loss (DL). We find that difference loss can facilitate the orthogonality between different classes and improve the performance on few-shot classification tasks. In addition, we further explore to replace the fastText (Joulin et al., 2016) with the BERT (Devlin et al., 2018) as the embedding encoder. It is regrettable that BERT does not bring performance improvement but will increase the model complexity. This indicates that in the few-shot scenario, powerful BERT may lead to the overfitting of the model, resulting in performance degradation.

6 Discussion

Our work follows the research idea proposed by Han et al. (Han et al., 2021) to exploit the adversarial network to catch feature representations. However, the implementation structure of MLADA (Han et al., 2021) is too simple to extract intrinsic sentence feature information and strengthen the generalization ability of the model well. In contrast, we upgrade the whole meta-learning framework with more cost-effective modules. The experimental results in Table 2 illustrate that the performance of our proposed SaAML is significantly better than MLADA.

Specifically, we build the generator with the temporal convolutional network (TCN) rather than bidirectional LSTM (BiLSTM) to excavate the inherent semantic knowledge of sentences. We also exploit the feature enhancer to grasp the rich feature interaction between different sentence representations through the multi-head attention mechanism. Moreover, we adopt the prototype network (PN) based on the difference loss as the classifier, improving the model's discriminative ability. Overall, our model SaAML can construct more comprehensive transferable representations and accomplish excellent performance.

In addition, we further discuss the adaptation potential of the SaAML and MLADA. We adopt RRML (Bertinetto et al., 2018) and PN (Snell

Table 3: The ablation results on Amazon, HuffPost, 20 Newsgroups, and Banking77 datasets.

	Amazon		HuffPost		20 Newsgroups		Banking77	
Model	5-way		5-way		5-way		10-way	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
-AN	0.6918	0.8470	0.4969	0.6764	0.7025	0.8302	0.6567	0.8360
-TCN	0.6966	0.8363	0.5057	0.6872	0.6745	0.8240	0.6789	0.8448
-SE	0.6108	0.7887	0.4683	0.6404	0.5589	0.7514	0.6456	0.8264
-FE	0.6005	0.7886	0.3773	0.5877	0.5443	0.7503	0.5646	0.6731
-DL	0.7075	0.8490	0.5065	0.6874	0.7043	0.8381	0.6837	0.8415
+BERT	0.6647	0.8340	0.4888	0.6815	0.5136	0.6737	0.6845	**0.8526**
SaAML	**0.7147**	**0.8637**	**0.5126**	**0.6944**	**0.7079**	**0.8430**	**0.6860**	0.8483

Table 4: The results for inserting augmentation components into PN and RRML on different datasets.

	Amazon		HuffPost		20 Newsgroups		Banking77	
Model	5-way		5-way		5-way		10-way	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
PN	0.3760	0.5214	0.3573	0.4474	0.3780	0.4535	0.4156	0.6975
MLADA-PN	0.5587	0.7220	0.3106	0.4632	0.5088	0.6453	0.4662	0.6055
SaAML-PN	**0.7147**	**0.8637**	**0.5126**	**0.6944**	**0.7079**	**0.8430**	**0.6860**	**0.8483**
RRML	0.5022	0.7275	0.3610	0.4966	0.3760	0.5724	0.5256	**0.8148**
MLADA-RRML	0.6842	**0.8600**	0.4502	0.6491	0.5961	0.7780	0.6055	0.8089
SaAML-RRML	**0.6856**	0.8422	**0.4588**	**0.6605**	**0.6680**	**0.8062**	**0.6130**	0.7859

et al., 2017) as the classifier to build the model, respectively. As exhibited in Table 4, regardless of whether RRML or PN is used as the classifier, SaAML obtains some performance progress across different datasets in contrast to the published original model and MLADA. It reveals that our SaAML has an enormous opportunity for updating diverse meta-learning systems.

7 Conclusion

In this paper, we propose a novel meta-learning framework Sentence-aware Adversarial Meta-Learner (SaAML) to address few-shot text classification task. Exactly, under the architecture of the adversarial network, we explore the representation generator with TCN as the core module to encourage more discriminative representation learning and exploit the feature enhancer to facilitate more consistent and comprehensive feature expression, which exceedingly strengthens the adaptability and generalization of SaAML for new classes. We develop comprehensive experiments on four benchmark datasets to demonstrate that the proposed model gains substantial improvements over existing state-of-the-art meta-learning approaches.

References

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2019. Few-shot text classification with distributional signatures. *arXiv preprint arXiv:1908.06039*.

Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. 2018. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Inigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

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*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. *arXiv preprint arXiv:1902.10482*.

A. Gupta, K. Thadani, and N. O'Hare. 2020. Effective few-shot classification with transfer learning. In *Proceedings of the 28th International Conference on Computational Linguistics*.

ChengCheng Han, Zeqiu Fan, Dongxiang Zhang, Minghui Qiu, Ming Gao, and Aoying Zhou. 2021. Meta-learning adversarial domain adaptation network for few-shot text classification. *arXiv preprint arXiv:2107.12262*.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*.

Xiang Jiang, Mohammad Havaei, Gabriel Chartrand, Hassan Chouaib, Thomas Vincent, Andrew Jesson, Nicolas Chapados, and Stan Matwin. 2018. Attentive task-agnostic meta-learning for few-shot text classification.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

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

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. 2019. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10657–10665.

Rishabh Misra. 2018. News category dataset.

Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563. PMLR.

C. Pan, J. Huang, J. Gong, and X. Yuan. 2019. Few-shot transfer learning for text classification with lightweight word embedding based models. *IEEE Access*.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. 2019. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32.

Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning.

J. Salamon and J. P. Bello. 2017. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, PP(3):1–1.

J. Schmidhuber. 1987. Evolutionary principles in self-referential learning. *genetic programming*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Pengfei Sun, Yawen Ouyang, Wenming Zhang, and Xinyu Dai. 2021. Meda: Meta-learning with data augmentation for few-shot text classification. In *IJCAI*, pages 3929–3935.

Shengli Sun, Qingfeng Sun, Kevin Zhou, and Tengchao Lv. 2019. Hierarchical attention prototypical networks for few-shot text classification. 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 476–485.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208.

C. Tong. 2019. Metagan: An adversarial approach to few-shot learning.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu. 2020. Dpgn: Distribution propagation graph network for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13390–13399.