CrossSimCapsule Network based on pretrain model for few-shot text classification

Anonymous EMNLP submission

Abstract

Text classification models usually struggle with insufficient labeled data and the difficulty of adapting quickly to unknown classes. Few-shot learning aims to tackle these problems with meta-learning. Typical few-shot text classifier usually samples classes from the support set and measures the vector representation of the query and class according to the cosine distance. However, it is difficult for a single vector to contain the various semantic information for a sentence under the few-shot setting. And we need to measure the semantic similarity between two sentences at a richer level. To this end, from the perspective of meta-learning, we proposed a novel architecture named CrossSimCapsule Network (CSCN), which designed a semantic-caps matrix representation to enrich the semantic information and used a cross-attention block to catch inter-relation between two semantic-caps matrices. Meanwhile, to obtain complete semantics, we proposed the dynamic fuse module to merging the different level's syntactic and semantic knowledge automatically. We evaluate our model in three related text classification datasets. The experiment shows that our model can learn better text representation and outperform the existing state-of-art approaches in all datasets.

1 Introduction

Text classification plays a critical role in many NLP systems. Most deep learning approaches require large-scale human-labeled utterances to learn new concepts. However, gathering enough labeled utterances is highly expensive and even impractical in many applications. In contrast to humans, learning with only one or a few examples is still a challenge for models, which motivates us to develop new algorithms.

Recently, many researchers tackle this problem by developing few-shot learning algorithms that aim to learn new concepts with limited labeled samples. One of the most popular methods at present is to use meta-learning to simulate many "few-shot" tasks, each one with only a few annotated text as a support set (Finn et al., 2017). This method extracts transferable knowledge and gains reasoning ability for unknown classes through extensive cross-task training. However, in the multi-task migration under low resource conditions, overfitting problems are often encountered. (Munkhdalai and Yu, 2017； Qi et al., 2018). To tackle these problems, many methods (Yu et al., 2018; Shalyminov et al., 2019; Karlinsky et al., 2019) introduce metric learning, which learns semantic representations of queries and classes, and then measures the similarity between them. It works well in few-shot learning, one of the main reasons is that it does not include the task-related parameters, but only contains the sentence encoder parameters. This can reduce the risk of overfitting, especially in few-shot settings. Due to the abstractness and complexity of natural language, it is difficult to find some semantic information in-depth through a single vector. Meantime, it is hard to capture the correlation between different queries and classes semantics by measuring the cosine distance. Therefore, the way of directly optimizing the semantic gap between queries and classes should be improved in the few-shot scenario.

To solve these problems, we look at the capsule network which represents an object by multiple capsules and uses a dynamic routing algorithm to encoder the relationship between part and whole (Sabour et al., 2017). Recently, Xia et al., (2018) shows that different semantic meanings in a single sentence can be represented via different capsules, for each capsule, the orientation corresponds to the

semantic attribute while length reflects the probability of its existence. Geng et al. (2019) indicated that the dynamic-routing algorithm could be used to fuse the part-sample semantic to whole-class features.

We conducted in-depth research and proposed a novel architecture named CrossSimCapsuled Network (CSCN). First, we design a new encoder module that outputs a semantic-caps matrix to represent a single sentence by combining the multi-attention and residual connections. Next, based on the encoder model's output, we use a cross-attention block to catch the inner-relation between each query capsule and support capsule. We utilize this cross-attention block to replace the classification layer in the origin capsule network, and reduce the risk of overfitting. Further, we use a pre-trained model as a primary sentence encoder, which is combined with dynamic routing to fuse the semantic features of different layers. In the end, we acquire the robust text representation while reducing the risk of overfitting and achieve better results than related methods. In general, our contributions are as follows:

- Put forward a novel few-shot text classification model based on capsule networks and pre-trained models. To the best of our knowledge, it is the first work that combines the pre-trained and capsule model in a meta-learning fashion for few-shot text classification.

- Design a semantic layer to enrich the semantic information output by the pre-trained model such as BERT, represent each text by a semantic-caps matrix, and dynamically fuse different syntactic and semantic knowledge. The experiment shows that we can obtain a better text representation than the related works.

- Improve on capsule network, the classify layer in the proposed CSCN model replaced by a novel cross-attention-block. This can reduce the risk of overfitting as in many metric-leaning methods and achieve better results.

- Outperforms the current state-of-art models on three few-shot text classification datasets, including the well-studied entity-relationship classification and the sentiment classification benchmarks.

2 Related work

Few-shot learning aims to train models for new classes with only a few labeled samples. Metric-learning is one of the most popular methods of few-shot learning. The key idea in this kind of approach is to train encoding networks to learn a robust embedding space where embedded samples belonging to the same class are located closely. Vinyals et al., (2016) proposed a Matching Network to determine the embedding space using a few-labeled support and query samples, where queries are classified based on the similarity scores between query and support. Snell et al., (2017) proposed a Prototype Network that computes the similarity between query samples and class prototypes instead of calculating the similarity between samples. The prototype representations of each class is acquired by averaging the per-class embedded samples and classifying queries by finding the nearest prototype in the embedding space.

Pre-trained models have attracted widespread attention, e.g. BERT (Devlin et al., 2018) can learn universal language representations, which is beneficial for downstream NLP tasks and can avoid training a new model from scratch (Qiu et al., 2020). Recently, some works used pre-trained models in few-shot settings. Chen et al. (2019) applied pre-trained models for few-shot natural language generation, achieved better results than other methods, and generalized well across multiple domains. Zhang et al. (2019) proposed PMAML which combines the pre-trained models and meta-learning methods for text classification in few-shot scenes. This work shows that the use of pre-trained models can bring benefits for few-shot learning. Our approach also uses a pre-trained model as a basic sentence encoder, but instead of directly using the output of the pre-trained model we designed a semantic module to enhance the output information and automatically integrate different levels of semantic knowledge. Experiments show that our method can learn more robust text representation.

We also improve the capsule network which is first proposed by Sabour et al. (2017). In the capsule-based model, the object is represented by a group of caps and learns the invariant in part and whole relationships are learned via a routing-by-agreements mechanism. Yang et al. (2018) successfully applied the capsule model to the text classification problem with large labeled datasets. Xia

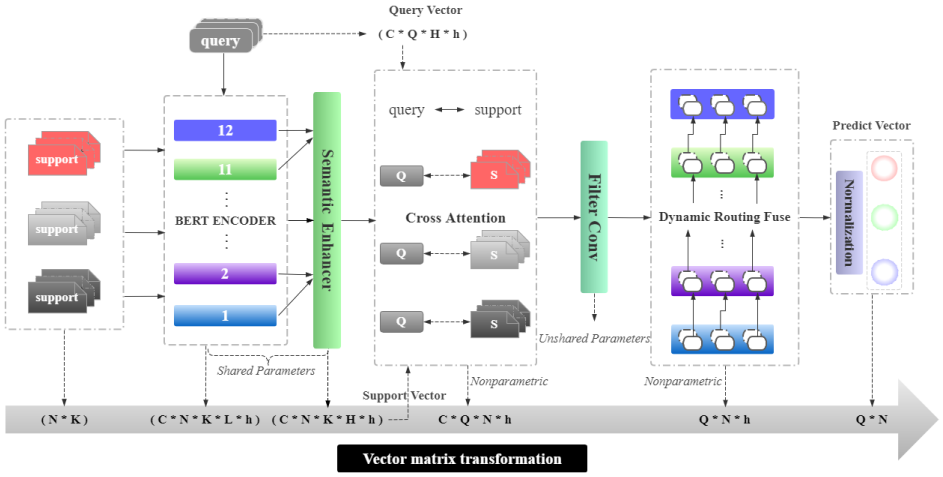

Figure 1: CrossSimCapsule Networks architecture for a N-way K-shot ($N = 3$, $K = 3$) problem with Q query examples ($Q = 3$). Where N and K correspond to the type and number of supports; C represents the number of layers of Bert; H represents the number of heads with multi-attention; h represents the size of hidden layer.

et al. (2018) proposed a novel capsule-based model for zero-shot intent detection, used an intent capsule network to extract semantic features from utterances and aggregated them by a dynamic routing algorithm. Geng et al. (2019) also proposed a capsule related model for few-shot text classification named Induction Network which learns class prototypes by using dynamic routing algorithms to aggregate per-class embedded samples. In this paper, we proposed a novel capsule-based network combined with pre-trained model named CrossSimCapsule Network for few-shot text classification. Different from Xia et al. (2018), we design a new semantic layer that uses pre-trained model as a primary sentence encoder and is equipped with the multi-head attention and residual connections to acquire more robust text representation. We also use the dynamic routing, different from Geng et al. (2019), to fuse separate layer's syntactic and semantic knowledge. And we replace the classification layer in capsule network by a novel cross-attention-block (without any learnable parameters). This is different from the above capsule-base models. With these novel structures, we outperform current state-of-art models on three few-shot text classification datasets.

3 Methodology

3.1 Notations and Definitions

Few-shot learning is a task from the field of meta-learning to solve the problem of a small number of samples or weak label annotation data (Qi et al., 2019). Precisely the task needs a large labeled training set D with a set of class C_{train}. Where D represents an annotated dataset, which is divided into $D_{train}(\forall_{class} \in \{1,2,\dots,M\})$ and D_{test} $(\forall_{class} \in \{M+1,\dots,C\})$ according to the labels. Note that the label space represented by D_{train} and D_{test} has no intersection. Our work ultimate goal can be summarized as training a classification model using D_{train}(containing a large amount of labeled classification data) employing meta-learning, and helps our model to obtain better generalization performance in D_{test}(containing a small amount of labeled new category data). Finally achieve a better classification effect in the test set.

3.2 Train Procedure

Vinyals et al. (2016) proved that "episode-based" training strategies are effective. We construct a meta episode to calculate the gradient and update the model through a large number of iterations. Formally speaking, in the training episode, we first stochastically sample N categories from D_{train} $(N \in M)$, and randomly select K examples from each class ($N \times K$ data in total) as the support set S. Then select q samples from the remaining data in the N categories ($N \times q$ data in total) as the query set T. We feed the S and T to the model and optimize the loss function. The purpose of this strategy is to help the model to learn how to distinguish these K categories from N $\times$ K data. This task is called an N-way K-shot problem.

4 Framework

Our CSCN model depicted in Figure 1, contains three modules: Semantic Encoder, Cross-attention-model, Semantic Fuse module. In the rest of this section, we will introduce these modules in turn.

4.1 Semantic Encoder

As shown in Figure 2, the semantic encoder uses the pre-trained model (BERT-Base) as a primary encoder with multi-head attention and residual connection. Given an input text $x = w_1, w_2, .. w_t$, we first input it to the BERT model and acquire all layer's output and pooled output as described in (Devlin et al., 2018). $L_i \in R^{l \times h}$ is the i^{th} output feature, where l is the sequence length and h indicates the hidden size. $P \in R^h$ is the pooled output. For each layer's output we use the attention mechanism to weight output features of different positions. Attention weight matrix A is computed as:

$$A = softmax(W_{a2} tanh\ (W_{a1} L_i^T + b)) \quad (1)$$

where $w_{a1} \in R^{h_a \times h}$ and $w_{a2} \in R^{H \times h_a}$ are the weight matrix and $b \in R^{h_a}$ is the bias. h_a is the hidden dimension size of multi-attention and H indicate the number of attention heads. The output is the weighted sum of L_i add the pooled output:

$$output_i = (\ A \times L_i\ ) + P \quad (2)$$

Where $output_i \in R^{H \times h}$. Therefore, the output of each layer is represented by a matrix and different heads in this matrix are different semantic caps as in (Xia et al. 2018). Note that the parameters in this module are shared across different layers' output.

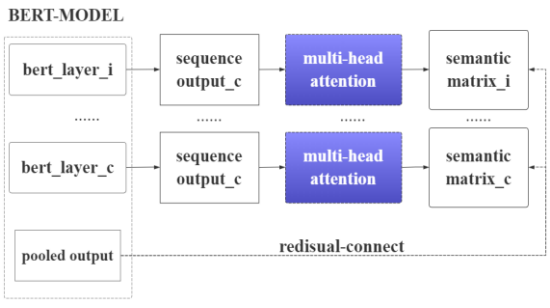

Figure 2: The architecture of Semantic encoder.

4.2 Cross-attention model

We represent each query and support sample in a semantic-caps matrix form. Each capsule in the matrix is a vector that represents a single semantic feature. But as different classes focus on different semantic attributes, different caps will contribute differently to the classification. Previous work (Sabour et al., 2017; Xia et al., 2018) weighted different caps by first using a high-order weight matrix as a classification layer to encode each semantic capsule then using the dynamic routing algorithms to catch how much each semantic contribute to different classes. However, the high-order weight matrix increases the risk of overfitting and reduces the model's transferability between various tasks. To solve these problems, we propose a novel non-parametric block named cross-attention block to compute the similarity information between each query capsule and support capsule and weight the query semantic matrix by using the similarity information. Therefore, a single query semantic feature is assigned by a high weight if it is similar to the corresponding support feature. For i^{th} layer's output, we describe the cross-attention block in Algorithm 1.

Algorithm 1 Cross-attention block

Procedure cross attention (T, H, Sn)

For each query q_i in query set T:
  For each support s_i in Sn $\{s_1, ... s_n\}$:
    For j in range H do:
      $q_{ij} = \frac{q_{ij}}{||q_{ij}||}$
      $s_{ij} = \frac{s_i}{||s_i||}$
      $r_{ij} = q_{ij} \cdot s_{ij}$
      Combine H $r_{ij} \rightarrow A_i \in R^H$
      Repeat A_i h times $\rightarrow RA_i \in R^{H \times h}$
      Weight q_i by $wq_i = q_i \odot RA_i$
  Combine N $wq_i \rightarrow Wq_i$
Combine Q $Wq_i \rightarrow WQ_i$
Return WQ_i
End procedure

Where H is the number of attention heads, Q is the number of queries in each task, $\odot$ means element wise product, $q_i \in R^{H \times h}$ and $s_i \in R^{H \times h}$ are the semantic encoder's outputs for the single query and support sample and q_{ij}, s_{ij} indicate the j^{th} capsule in q_i, s_i respectively. Sn is the class level support set which is computed by average per-class embedded support samples. So i^{th} element s_i in Sn indicates the semantic information of i^{th} class. A_i is the cosine similarity between q_i and i^{th} class s_i among different

caps which can indicates how well q_i matches class i. As show in algorithm1, Q is the number of queries, N is the number of classes for q_i to a single class s_i, we use this matching information vector A_i to weight q_i then obtain $wq_i \in R^{H\times h}$ (Line 8 and Line 9); For q_i to all classes set Sn, we obtain weighted query $Wq_i \in R^{N\times H\times h}$ (Line 10); For all query to all classes the whole weighted query semantic matrix based on the i^{th} layer's output is a fourth-order matrix $WQ_i \in R^{Q\times N\times H\times h}$ (Line 11). By doing so, we weighted each query semantic matrix by compute the similarity information between each query and all support samples via different semantic caps.

4.3 Semantic Fusion

Based on the output of the cross-attention model, the weighted query semantic matrix is a fourth-order matrix, a high-order matrix may contain redundant information, so we use a convolution layer to filter out the excess information while aggregating the information between different semantic caps. First, we transpose the WQ_i to $WQ_i^T \in R^{Q\times N\times h\times H}$, as in image processing, we treat the information on different caps as different image channels and the weighted information matrix corresponding to a capsule ($\in R^{N\times h}$) is used as a feature map.

$$ConvQ_i = Conv\ (\ WQ_i^T\ ) \in R^{Q\times N\times h} \quad (3)$$

Note the convolution layer is not shared across all layers, different layers' outputs correspond to separate convolution layers. Suppose the number of layers is c, we concatenate all layers' output to get $ConvQ_i \in R^{Q\times N\times c\times h}$, containing all layers' weighted query semantic information, Rogers A et al., (2020) shows that different layers' output in the pre-trained model corresponds to different levels' semantic. And (Xia et al., 2018; Geng et al., 2019) indicate the dynamic routing algorithm is a non-parametric method that can dynamically integrate low-level local semantics to obtain complete high-level semantics. Therefore, it is natural to think of using a dynamic routing algorithm to synthesize different layers' semantic information. We use a dynamic routing algorithm to weight sum different layers' information.

$$s_n = \sum_c^C c_{nc} P_{nc} \quad (4)$$

where c_{nc} is the weight indicating how much the c^{th} layer's semantic feature contributes to the n^{th} class. The query semantic features are denoted as $P_{nc} \in R^h$ corresponding to c^{th} layer and n^{th} class. For a single query $Conv\_q_i \in R^{N\times c\times h}$, we summarize this algorithm in Algorithm 2:

Algorithm 2 Dynamic routing algorithm

procedure dynamic routing (p, $iter$)
1. For all layer's semantic information C and class n: initialize $b_{nc} = 0$.
2. For $iter$ iterations do:
3. $c_{nc} = softmax(b_{nc})$
4. $s_n = \sum_c^C c_{nc} P_{nc}$
5. $v_n = squash(s_n)$
6. Update b_{nc} by $b_{nc} \longleftarrow b_{nc} + P_{nc} \bullet v_k$
7. End for
8. Return v_k

End procedure

Where the squashing function is used to get an activation vector v_k for each class.

$$v_k = \frac{||s_k||^2}{1+||s_k||^2} \frac{s_k}{||s_k||} \quad (5)$$

The orientation of the activation vector represents the semantic properties, while the norm indicates how it exits. These characteristics can be achieved by designing a suitable learning process.

4.4 Objective Function

As in capsule network (Sabour et al., 2017), the vector norm of the prediction matrix $predict_q \in R^{Q\times n\times h}$ indicates the probability belonging to a specific class. In order to learn those attributes, we use the max-margin loss to optimize the vector norm in the prediction matrix.

$$\begin{aligned} L = \sum\{ & I(y = y_q) \cdot max(0, m^+ - \|predict_q\|^2)\} \\ & + \lambda I(y \neq y_q) \cdot \max(0, \|predict_q\|^2 - m^-)^2) \\ & + \alpha \|AA^T - \mathbb{I}\|_F^2 \end{aligned} \quad (6)$$

where I is an indicator function, y is the ground truth class label for input x, y_q is the model's predicted label for the query, λ is a weighting coefficient, m^+ and m^- are margins. α is a non-negative trade-off coefficient that encourages the discrepancies among different attention heads.

5 Experiments

In this section, we will introduce the experiment related details of our model. We evaluate our model on three few-shot text classification datasets and achieve state-of-the-art results. Additionally, we

will show how our model works by ablation study and visualization of different query instances. The details are as follows.

5.1 Datasets

FewRel 1.0 Few-Shot Relation Classification [1] (Han et al., 2018) is a new large-scale supervised dataset. It consists of 70000 instances on 100 relations derived from Wikipedia, and each relation includes 700 instances. It annotated the head and tail entities in each instance. FewRel 1.0 has separated 64, 16, and 16 relations for training, validation, and testing respectively.

Amazon Review Sentiment Classification (ARSC[2]) According to (Geng et al., 2019), we use multiple tasks with the multi-domain sentiment classification dataset. The dataset contains user reviews of 23 products from the Amazon platform. For each product domain, there are three different binary classification tasks. The whole buckets consist of $23 \times 3 = 69$ tasks. Following Yu et al. (2018), they selected $3 \times 4 = 12$ tasks from 4 domains to form the test set as the meta-test (target) tasks for all 23 domains. To this end, we train our model according to the 5-shot learning task on this dataset. Compared with the experiments of Yu et al. (2018) and Geng et al. (2019), the purpose of our research is to evaluate the accuracy of the model on new categories not seen before. Therefore, we excluded the training data of 12 tasks related to the test set from the training set and only retained the remaining $19 \times 3 = 57$ tasks as the training set. This dramatically increases the difficulty of generalization of the model.

SemEval-2010 task 8 SemEval-2010 Task 8 [3] (Hendrickx et al., 2009) focuses on the multi-way classification of semantic relations between pairs of nominals. The task was designed to compare different approaches to semantic relation classification. The dataset has 10717 annotated examples in total which 8000 training sets and 2717 test sets. It covers 9 relationships and a different category to assess models' ability to make such fine-grained distinctions. In order to evaluate the model's performance to learn the unseen class under few-shot settings, SemEval-2010 Task 8 has separated subsets for training, verification, and testing, covered by 6, 2, and 2 relations, respectively.

[1] https://github.com/thunlp/fewrel
[2] https://github.com/Gorov/DiverseFewShot_Amazon

5.2 Experiment Setup

Baselines In this section, we introduce the five state-of-the-art few-shot learning models of the above datasets. Details are as follows:

Proto Prototypical Network (Snell et al., 2017) is proposed in the context of few-shot learning at first. It applies the idea of the metric learning and classifies by calculating the distance from a query to the representation of the class prototype.

Siamese Siamese Neural Network (Koch et al., 2015) measures the similarity of two vectors by distance. Guo et al. (2017) proved that it has great potential in realizing fast calculation based on matching, so it is widely used in the text-similarity calculation.

BERT-PAIR BERT-PAIR Gao et al. (2019) based on the sequence classification model in BERT. The model matches the query instances in each class with all support instances, and sends each pair in series as a sequence to the BERT sequence classification model to obtain the scores of two instances expressing the same relationship.

PMAML The method of meta-learning using the MAML framework with pre-trained language representations is called PMAML (Zhang et al., 2019). It is to complete the downstream few-shot text classification task in combination with the MAML framework on the premise of the pre-training task for BERT.

Optimize settings Experimental parameters are shown in Table 1.

Parameters	Value
LearningRate (BERT)	$1e^{-5}$
LearningRate (Multi-Head + Filter_Conv)	$5e^{-4}$
HiddenSize (CrossAttention)	50
Multi-head Numbers	200
Max Length (FewRel 1.0, ARSC)	128
Max Length (SemEval-2010 Task 8)	88
Filter Kernel Size	[N, 3]
Filters Number	1

Table 1: Specific parameter settings, where N is the number of ways.

5.3 Evaluation Metrics

For FewRel 1.0 and ARSC datasets, we evaluate the performance by few-shot classification accuracy according to previous research in few-shot learnings (Snell et al., 2017; Baldini et al., 2019).

[3] https://github.com/CrazilyCode/SemEval2010-Task8

Model	5-Way Acc. 1-Shot	5-Way Acc. 5-Shot	10-Way Acc. 1-Shot	10-Way Acc. 5-Shot
BERT - Proto	80.16 ± 0.13	89.60 ± 0.06	70.32 ± 0.17	81.52 ± 0.09
BERT - Siamese	73.36 ± 0.06	81.33 ± 0.37	67.26 ± 0.64	79.64 ± 0.17
BERT - PAIR	**85.66** ± 0.64	89.48 ± 0.13	**76.84** ± 0.17	81.76 ± 0.12
CSCN (ours)	85.46 ± 0.23	**93.51** ± 0.09	75.81 ± 0.18	**87.49** ± 0.05

Table 2: Comparison of mean accuracy (%) on FewRel 1.0. Note that BERT-Proto and BERT-PAIR refer to the public work published by Gao et al., (2019). See footnote 1 for code details.

Model	2-Way Acc. 5-Shot	2-Way Acc. 5-Shot (new class)	Avg Acc.
BERT - Proto	78.63	72.71	75.67
BERT - Siamese	85.46	79.23	82.35
BERT - PAIR	83.49	73.31	78.40
PMAML	85.28	84.88	84.83
CSCN (ours)	**86.60**	**85.45**	**86.03**

Table 3: Comparison of mean accuracy (%) on ARSC. Among them, PMAML refers to the public work of Zhang et al., (2019). For experimental details, refer to https://github.com/zxlzr/FewShotNLP.

Referring to the experiments of Gao et al. (2019), We constructed 1-shot and 5-shot tasks for FewRel 1.0 dataset. Since the test set was not public and the submission period is too long, we compare the effects of different models on the verification set. The training process of the ARSC dataset is different from that of Yu et al. (2018). During the training period, we used 57 tasks of train data for training, and the remaining 12 for testing. In this way, the model is equivalent to identifying 12 unknown tasks during the testing phase. Consequently, we just need to run the test episode once for each of the target tasks. We implemented all baseline models on ARSC with an adjusted few-shot task.

SemEval-2010 task 8 (Hendrickx et al., 2009) focuses on the semantic relationship between two nouns. We refer to the work of predecessors (Gao et al., 2019) and construct it as the 2-way 5-shot classification task under the few-shot scenario. Our purpose is to evaluate the performance of CSCN in unknown classes, for which there are eight classes for training and two for testing. We compare the average F1 score obtained by CSCN on the test set with the results of the five SOTA benchmarks introduced in Chapter 5.2.

5.4 Experiment Results

Overall Performance Table 2 records the test results of the FewRel 1.0 data set. The capsule network has reached an accuracy of 93.51% on the 5 way 5-shot task, which is a significant improvement of 4.03% compared to the current latest model BERT-PAIR. Benefiting from multi-attention and cross-attention capsule networks, our model achieves excellent performance. In few-shot learning, the generalization performance of the model on unseen classes is very critical. BERT-PAIR combines each query and each support separately under the same class, which makes the model affected by noise and cannot better summarize the abstract features of the class. It exploits sample-level semantic representation to obtain the correlation index of two instances, which is difficult to effectively filter the noise of different expressions of the same class. Besides, each query instance needs to be used in conjunction with all supporting examples under the same task, which will lead to an increase in the number of samples and calculations during encoding and is not suitable for applications in scenarios where tasks change rapidly. On the contrary, our model CSCN can extract a high-level summary of the class features from the support examples, so that each query and class features can dynamically interact during optimization. The experiment results prove that our CSCN network can have a good generalization of the unseen new class.

Simultaneously, we also evaluate our method on datasets from two different fields (ARSC and Semeval-2010 task 8). Table 3 and Table 4 compare the results of ARSC and Semeval-2010 task 8 in the baseline. Meanwhile, PMAML is the latest SOTA method on ARSC. It combines the pre-trained model and the MAML framework based on

a stochastic gradient strategy to realize the downstream few-shot text classification task. To ensure the fairness of the results, we used BERT-Base as the encoder of all benchmark models in the experiment. From the perspective of the model approach, these baselines are measured for sample-level distances. Our work focuses on the semantic representation and enhancement of the same task level, which can improve the model's ability to abstract and summarize the entire class so that the model has an excellent performance in both relationship classification and emotion classification tasks.

Model	2-Way F1. 5-Shot
Bert-Proto	78.63
Bert-Siamese	85.46
Bert-PAIR	83.49
PMAML	85.28
CSCN (ours)	**87.67**

Table 4: Comparison of mean F1 score (%) on SemEval-2010 task 8.

Ablation Study In order to analyze the effect and influence of specific modules in the CSCN structure on text classification, we conducted an ablation study on the 5-way 5-shot task on the FewRel 1.0 dataset. As shown in Table 5, our model achieves 93.51% accuracy in 5-way 5-shot few-shot task. Experiments show that the accuracy obtained by the model is reduced by 1.2, 1.74, 1.93, 73.56 percents respectively when the four modules of Residual Connection, dynamic fusion of different layers' outputs, filtering convolution and cross-attention block are removed respectively. This further clarifies that in few-shot scenarios, the framework we proposed based on the capsule network is of great significance. It can effectively improve the model's semantic summarization and generalization capabilities. Note in the ablation study, we replace the Dynamic Fusion by simple average the different layer's semantic information, replace the Filter Convolution by average among different caps and replace the Cross Attention with a classification layer.

Model	5-Way 5-Shot	Reduce value
CSCN (ours)	**93.51**	-
- Residual Connection	92.31	↓ 1.2
- Dynamic Fusion	91.77	↓ 1.74
- Filter Convolution	91.58	↓ 1.93
- Cross Attention	19.95	↓ 73.56

Table 5: Ablation study for CSCN on Fewrel 1.0 dataset of 5-way-5-shot task.

Comparing Encoder Performance In order to facilitate the demonstration of the effectiveness of our method, we use t-SNE(Maaten and Hinton, 2008) to visualize the text vectors output by CSCN and BERT-Siamese in the 5-way 5-shot scene. Figure 3 shows the visualization results of query vectors randomly selecting 5 categories (each category randomly draws 100 data) from the FewRel 1.0 training set. It is clear that the text embedding learned by CrossSimCapsule Networks is better separated semantically than those of the BERT-Siamese. To a certain extent, our CSCN methods can capture not only effective features to complete the semantic measurement of support and query, but also give different weights to samples and features during backpropagation through cross-attention and dynamic routing algorithms. Therefore the encoder can learn better text expression.

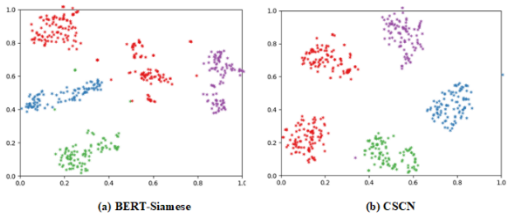

Figure 3: Query text vector visualization learn by (a) BERT-Siamese, (b) CSCN.

6 Conclusion

In this paper, we propose a novel model for few-shot text classification which combines pre-trained model and capsule network with a meta-learning framework. We call this CrossSimCapsule Network which introduces a novel semantic encoder module to represent different semantics in a semantic capsule matrix and a novel non-parametric cross-attention-block to catch the inter-relation between the query and support semantic-caps. Domain adaptation and NOTA detection are new challenges from the real world, and it is difficult for the existing few-shot classification model to deal with these problems. In future work, we will pay more attention to the above issues and conduct in-depth exploration.

References

Baldini Soares, Livio and FitzGerald, Nicholas and Ling, Jeffrey and Kwiatkowski, Tom. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of ACL.*

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of ICML*.

Congying Xia, Chenwei Zhang, Xiaohui Yan,Yi Chang, and Philip Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of EMNLP*.

Guo-Jun Qi and Jiebo Luo. 2019. Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods. arXiv preprint arXiv:1903.11260.

Guo, Qing, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. 2017. Learning dynamic siamese network for visual object tracking. In *Proceedings of ICCV*.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *Proceedings of ICML*.

Hang Qi, Matthew Brown, and David G Lowe. 2018. Low-shot learning with imprinted weights. In *Proceedings of CVPR*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O S eaghdha, Sebastian Pado, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8:Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of SEW-2009*, pages 94–99.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Proceedings of NIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

Kim Jongmin, Kim Taesup, Kim Sungwoon, and Chang D. Yoo. 2019. Edge-labeling graph neural network for few-shot learning. In *Proceedings of CVPR*.

Karlinsky, Leonid, Joseph Shtok, Sivan Harary, Eli Schwartz, Amit Aides, Rogerio Feris, Raja Giryes, and Alex M. Bronstein. 2019. RepMet: Representative-based metric learning for classification and few-shot object detection. In *Proceedings of CVPR*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. Journal of machine learning research, 9(Nov):2579–2605.

Min Yang, Wei Zhao, Jianbo Ye, Zeyang Lei, Zhou Zhao, and Soufei Zhang. 2018. Investigating capsule networks with dynamic routing for text classification. In *Proceedings of EMNLP*.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. In *Proceedings of NAACL*.

Ningyu Zhang, Zhanlin Sun, Shumin Deng, Jiaoyan Chen, and Huajun Chen. 2019. Improving few-shot text classification via pretrained language representations. arXiv preprint arXiv:1908.08788v1.

Ruiying Geng, Binhua Li, Yongbin Li, Yuxiao Ye, Ping Jian, and Jian Sun. 2019. Few-shot text classification with induction network. In *Proceedings of EMNLP*.

Rogers, A. , Kovaleva, O. , and Rumshisky, A. .2020. A primer in bertology: what we know about how bert works.

Shalyminov, Igor, Sungjin Lee, Arash Eshghi, and Oliver Lemon. 2019. Few-Shot Dialogue Generation Without Annotated Data: A Transfer Learning Approach. In *Proceedings of SIGDIAL*.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Proceedings of NIPS*.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of EMNLP*.

Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *Proceedings of ICML*.

Vinyals, Oriol, Charles Blundell, Timothy Lillicrap, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Proceedings of NIPS*.

Victor Garcia and Joan Bruna. 2018. Few-shot learning with graph neural networks. In *Proceedings of ICLR*.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained Models for Natural Language Processing: A Survey. arXiv preprint arXiv:2003.08271.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of EMNLP*.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020. Few-shot NLG with pre-trained language model. In *Proceedings of ACL*.