Few-Shot Text Classification with Induction Network

Ruiying Geng[1,2] , **Binhua Li**[2] , **Yongbin Li**[2] and **Yuxiao Ye**[2] and **Ping jian**[1] **Jian Sun**
[1]Department of Computer Science, Beijing Institute of Technology
[2]Alibaba Group
{ruiying.gry, binhua.lbh, shuide.lyb}@alibaba-inc.com, pjian@bit.edu.cn, jian.sun@alibaba-inc.com

Abstract

Text classification tends to struggle when data is deficient or when it needs to adapt to unseen classes. In such challenging scenarios, recent studies often use meta learning to simulate the few-shot task, in which new queries are compared to a small support set on a sample-wise level. However, this sample-wise comparison may be severely disturbed by the various expressions in the same class. Therefore, we should be able to learn a general representation of each class in the support set and then compare it to new queries. In this paper, we propose a novel Induction Network to learn such generalized class-wise representations, innovatively combining the dynamic routing algorithm with the typical meta learning framework. In this way, our model is able to induce from particularity to university, which is a more human-like learning approach. We evaluate our model on a well-studied sentiment classification dataset (English) and a real-world dialogue intent classification dataset (Chinese). Experiment results show that, on both datasets, our model significantly outperforms existing state-of-the-art models and improves the average accuracy by more than 3%, which proves the effectiveness of class-wise generalization in few-shot text classification.

逐样本对比会因为同类但不同表达样本的扰动

需要学习到支持集中更泛化的表式

将动态路由组合到meta learning中

1 Introduction

Deep learning (DL) has achieved great success in many fields such as computer vision, speech recognition and machine translation [Kuang *et al.*, 2018; Peng *et al.*, 2018; Lin *et al.*, 2017a], due to the development in optimization technology, larger datasets and streamlined designs of deep neural architectures. However, DL is notorious for requiring large labeled datasets, which limits the scalability of deep models to new classes due to the annotation cost. Humans on the other hand are readily able to rapidly learn new classes with few examples. This significant gap between human and machine learning provides fertile ground for deep learning developments.

Few-shot learning devotes to resolving the data deficiency problem by recognizing novel classes from very few labeled examples. The limitation of only one or very few examples challenges the standard fine-tuning method in deep learning. Early studies [Schölkopf *et al.*, 2002] applied data augmentation and regularization techniques to alleviate the overfitting problem caused by data scarcity, only to a limited extent. Instead, researchers have explored meta-learning [Finn *et al.*, 2017] to leverage the distribution over similar tasks, inspired by human learning. Contemporary approaches to few-shot learning often decompose the training procedure into an auxiliary meta learning phrase, which includes many sub-tasks, following the principle that the testing and training conditions must match. They extract some transferable knowledge by switching the task from mini-batch to mini-batch. And then, the few-shot model can classify new classes with just a small labeled support set.

元学习：利用了相似任务的分布

However, existing approaches for few-shot learning are still faced with some problems, including imposed strong priors [Fe-Fei and others, 2003], complex gradient transfer between tasks [Munkhdalai and Yu, 2017], and fine-tuning the target problem [Long *et al.*, 2016]. The approaches proposed by Snell *et al.* [2017] and Yang *et al.* [2018], which combine non-parametric methods and metric learning, may provide possible solutions to those problems. The non-parametric methods allow novel examples to be rapidly assimilated, without suffering from catastrophic overfitting. Such non-parameter models only need to learn the representation of the samples and the metric measure. However, in previous studies, class-level representations are calculated by simply summing or averaging representations of samples in the support set. In doing so, key information may be lost in the noise brought by various forms of samples in the same class. This problem will be more serious when the support set becomes larger. In fact, expressions of the same class are interlinked with shared features, but they also have their own special characteristics. The information irrelevant to the classification would heap up if accumulated one by one. Instead, a better learning approach may be induction on a class-wise level: ignoring irrelevant details and encapsulating general semantic information from samples with various linguistic forms in the same class.

非参数方法允许新样本在不过拟合的情况下快速学习

之前的方法：类特征只简单将类内样本平均或求和

会受到噪声的严重影响

更好的归纳类特征的方法：忽略不相关特征，封装广义语义信息

As a result, there is a need for a perspective architecture that can reconstruct hierarchical representations of support sets and dynamically induce sample representations to class representations. In this work, we propose the Induction Network, which explicitly models the ability to learn generalized class-level representation from samples in a small support set, based on dynamic routing [Sabour *et al.*, 2017] and a typical meta learning framework. First an Encoder Module generates representations of the query and support samples. Then an Induction Module builds a dynamic routing procedure, which regards the sample representations as input capsules, and each capsule learns to recognize an implicitly defined semantic viewpoint of the class. With the output capsule representing the class, this module can preserve the deep semantic representation of each class, and alleviate the disturbance of irrelevant noisy information. Finally, it becomes an interaction problem between the query and the class. Their representations are compared by a Relation Module that determines if the query matches the class or not. Defining an episode based meta training strategy, the holistic model is meta trained end-to-end with the generalisable applicability and scalability to recognize unseen classes.

归纳网络使用动态路由策略让每个capsule识别该类潜在的深度语义观点

可以减轻不相关信息噪声的扰动

The contributions of this work are listed as follows:

- We propose an Induction Network to deal with the few-shot text classification problem, which combines the dynamic routing algorithm with a typical meta-learning framework to simulate human-like induction ability.
- Our method outperforms prior state-of-the-art system on two benchmark corpora and improves the average accuracy by more than 3%, including a well-studied sentiment classification dataset (English) and a real-world dialogue intent classification dataset (Chinese). We will release the source code and dataset after publication.

2 Related Work

2.1 Few-Shot Learning

The seminal work on few-shot learning dates back to the early 2000's [Fe-Fei and others, 2003; Fei-Fei *et al.*, 2006]. The authors combined generative models with complex iterative inference strategies. Recently, many approaches use a meta-learning [Finn *et al.*, 2017; Yang *et al.*, 2018] strategy in the sense that they extract some transferrable knowledge from a set of auxiliary tasks, which then helps them to learn the target few-shot problem well without suffering from overfitting. Generally, these approaches can be divided into two categories:

Optimization-based Methods: This kind of approaches aims at learning to optimize the model parameters given the gradients computed from the few-shot examples. Munkhdalai and Yu [2017] proposed the Meta Network, which learnt the meta-level knowledge across tasks and shifted its inductive biases via fast parameterization for rapid generalization. Mishra *et al.* [2018] introduced a generic meta-learner architecture called SNAIL which used a novel combination of temporal convolutions and soft attention.

Meta-Net利用快速参数化达到倾斜偏置以达到快速泛化.

时序卷积和软注意力.

Distance Metric Learning: Unlike the above approaches that entail some complexity when learning the target few-shot problem, this category aims to learn a set of projection functions that take query and sample from the target problem and classify them in a feed forward manner. Respectively, Vinyals *et al.* [2016] produced a weighted K nearest neighbor classifier measured by the cosine distance, which was called Match Network. Snell *et al.* [2017] proposed the Prototypical Network which learnt a metric space where classification could be performed by computing squared Euclidean distances to prototype representations of each class. Different from fixed metric measures, the Relation Network learnt a deep distance metric to compare the query with given examples [Yang *et al.*, 2018].

Recently, there are some studies focusing on few-shot text classification problem. Yu *et al.* [2018] argued that the optimal meta-model may vary across tasks, and they employed multi metrics model by clustering the meta tasks into several defined clusters. Rios and Kavuluru [2018] developed a few-shot text classification model for multi-label text classification when there was known structure over the label space. Differently, we solve the few-shot learning problem in a higher perspective and propose a dynamic routing induction method to encapsulate the abstract class representation from the samples, achieving state-of-the-art performance in two datasets.

2.2 Capsule Network

capsuleNet可以连续地学习部分-整体的不变关系

The Capsule Network was first proposed by Sabour *et al.* [2017], which allowed the network to learn part-whole invariant relationships consecutively. Lately, Capsule Network was explored in the natural language processing field. Zhao *et al.* [2018] successfully applied Capsule Network to text classification. However, their work still focused on the traditional text classification problem with large labeled data. Xia *et al.* [2018] proposed a capsule-based architecture for zero-shot intent classification with the ability to discriminate emerging intents. Their main framework was also a traditional supervised classifier which was trained with large labeled data. For zero-shot intents, they reused the supervised classifier.

In this work, we propose the Induction Network which explicitly model the ability to learn generalized class-level representation from samples in a small support set. With the spirit of dynamic routing, our model can filter the irrelevant details of the utterances and get the essence of the class, which is much familiar with humans' induction learning process: encapsulating from concretization to abstraction, and from particularity to universality.

3 Problem Definition

3.1 Few-Shot Classification

Few-shot classification [Vinyals *et al.*, 2016; Snell *et al.*, 2017] is a task in which a classifier must be adapted to accommodate new classes not seen in training, given only a few examples of each of these new classes. We have a large labeled training set with a set of classes C_{train}. However, after training, our ultimate goal is to produce classifiers on the testing set with a disjoint set of new classes C_{test}, for which only a small labeled support set will be available. If the support set contains K labeled examples for each of the C unique classes, the target few-shot problem is called C-way K-shot problem. Usually, the K is too small to train a supervised classification model. Therefore we aim to perform meta-learning on the training set, in order to extract transferrable knowledge that will allow us to perform better few-

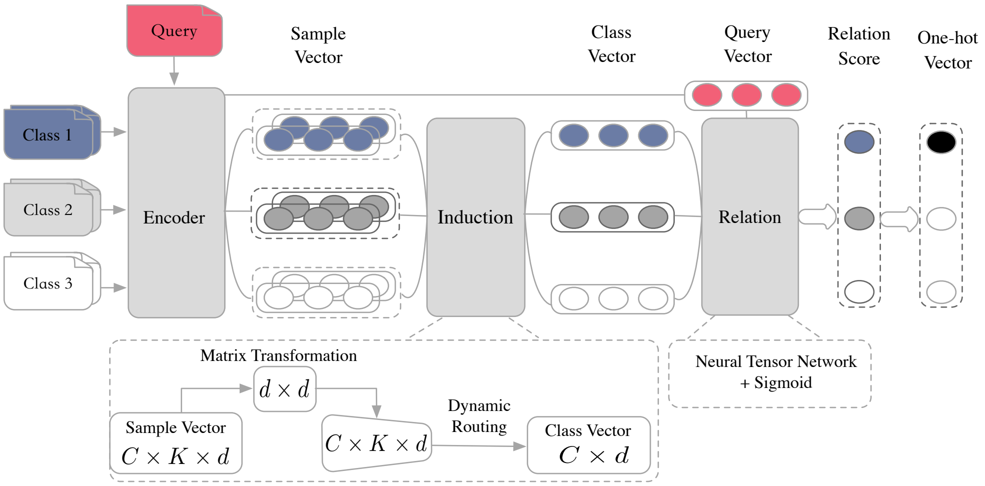


Figure 1: Induction Network architecture for a C-way K-shot ($C = 3$, $K = 2$) problem with one query example.

shot learning on the support set and thus classify the test set more successfully.

3.2 Training Procedure

The training procedure has to be chosen carefully so as to match inference at test time. An effective way to exploit the training set is to decompose the training procedure into an auxiliary meta learning phrase, and mimic the few-shot learning setting via episode based training, as proposed in Vinyals *et al.* [2016]. We construct an "episode" to compute gradients and update our model in each training iteration. The training episode is formed by randomly selecting a subset of classes from the training set, then choosing a subset of examples within each selected class to act as the support set S and a subset of the remainder examples to serve as the query set Q. Training on such episodes is done by feeding the support set S to the model and updating its parameters to minimize the loss in the query set Q. We call this strategy as episode-based meta training and the details are shown in Algorithm 1. The use of episodes makes the training procedure more faithful to the test environment and thereby improves generalization. It is worth noting that there are exponentially many possible meta tasks to train the model on, making it hard to overfit. For example, if a dataset contains 159 training classes, this leads to $\begin{pmatrix} 159 \\ 5 \end{pmatrix} = 794,747,031$ possible $5-way$ tasks.

采取与RN中相同的训练策略

由于每次选C个类，因此可能的组合是指数级的，这让训练更不会过拟合

C_N^k 种组合

Algorithm 1 Episode-Based Meta Training

```
1: for each $episode\ iteration$ do
2:     Randomly select $C$ classes from the class space of the training set;
3:     Randomly select $K$ labeled samples from each of the $C$ classes as support set $S = \{(x_s, y_s)\}_{s=1}^{m}$ ($m = K \times C$), and select a fraction of the reminder of those $C$ classes' samples as query set $Q = \{(x_q, y_q)\}_{q=1}^{n}$;
4:     Feed the support set $S$ to the model and update the parameters by minimizing the loss in the query set $Q$;
5: end for
```

类内样本也是随机选出的

4 Model

Our Induction Network, depicted in Figure 1 (the case of 3-way 2-shot model), consists of three modules: Encoder Module, Induction Module and Relation Module. In the rest of this section, we will show how these modules work in each meta-training episode.

4.1 Encoder Module

This module is a bi-direction recurrent neural network with self-attention as shown in [Lin *et al.*, 2017b]. Given an input text $x = (w_1, w_2, ..., w_T)$, represented by a sequence of word embeddings. We use a bidirectional LSTM to process the text:

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(w_t, h_{t-1}) \tag{1}$$

$$\overleftarrow{h_t} = \overleftarrow{LSTM}(w_t, h_{t+1}) \tag{2}$$

将前向状态与后向状态相连接成为隐藏态

And we concatenate $\overrightarrow{h_t}$ with $\overleftarrow{h_t}$ to obtain a hidden state h_t. Let the hidden state size for each unidirectional LSTM be u. For simplicity, we note all the T h_ts as $H = (h_1, h_2, ..., h_T)$. Our aim is to encode a variable length of text into a fixed size embedding. We achieve that by choosing a linear combination of the T $LSTM$ hidden vectors in H. Computing the linear combination requires the self-attention mechanism, which takes the whole LSTM hidden states H as input, and outputs a vector of weights a:

长度为T的文本的T个隐藏态的线性组合生成定长输出
线性组合的方式通过自注意力机制

$$a = softmax(W_{a2}tanh(W_{a1}H^T)) \tag{3}$$

here $W_{a1} \in R^{d_a \times 2u}$ and $W_{a2} \in R^{d_a}$ are weight matrixes and d_a is a hyperparameter. The final representation e of the text is the weighted sum of H:

$$e = \sum_{t=1}^{T} a_t \cdot h_t \tag{4}$$

encoder是双向循环网络
+
自注意机制

4.2 Induction Module

This section introduces the proposed dynamic routing induction algorithm. We regard these vectors e obtained from the support set S by Eq 4 as sample vectors e^s, and the vectors e from the query set Q as query vectors e^q. The most important step is to extract the representation for each class in the support set. The main purpose of the induction module is to design a non-linear mapping from sample vector e^s_{ij} to class vector c_i:

$$\left\{e^s_{ij} \in R^{2u}\right\}_{i=1,\ldots C, j=1\ldots K} \mapsto \left\{c_i \in R^{2u}\right\}_{i=1}^{C}.$$

We apply the dynamic routing algorithm [Sabour *et al.*, 2017] in this module, in the situation where the number of the output capsule is one. In order to accept *any*-way *any*-shot inputs in our model, a weight-sharable transformation across all sample vectors in the support set is employed. All of the sample vectors in the support set share the same transformation weights $W_s \in R^{2u\times 2u}$, so that the model is flexible enough to handle the support set at any scale. Each sample prediction vector $\hat{e}^s_{ij}$ is computed by:

使用权值共享的转换矩阵来转换所有样本的语义信息

$$\hat{e}^s_{ij} = W_s e^s_{ij} \tag{5}$$

①先利用转换矩阵将样本转换

which encodes important invariant semantic relationships between lower level sample features and higher level class features. To ensure the class vector encapsulate the sample feature vectors of this class automatically, dynamic routing is applied iteratively. In each iteration, the process dynamically amends the connection strength and makes sure that the coupling coefficients d_i sum to 1 between class i and all support samples in this class by a "routing softmax":

$$d_i = softmax\left(b_i\right) \tag{6}$$

②再对同类的关联系数 Softmux 求得加权值

where b_i is the logits of coupling coefficients, and initialized by 0 in the first iteration. Given each sample prediction vector $\hat{e}^s_{ij}$, each class candidate vector $\hat{c}_i$ is a weighted sum of all sample prediction vectors $\hat{e}^s_{ij}$ in class i:

$$\hat{c}_i = \sum_j d_{ij} \cdot \hat{e}^s_{ij} \tag{7}$$

③利用转换后的表示和加权值求得类表示向量

then a non-linear "squashing" function is applied to ensure that length of the vector output of the routing process will not exceed 1 as shown in Equation 8. The short vectors get shrunk to almost zero length and the long vectors get shrunk to a length slightly below 1. The squash function leaves the direction of the vector unchanged but decreases its magnitude.

$$c_i = \frac{\|\hat{c}_i\|^2}{1+\|\hat{c}_i\|^2}\frac{\hat{c}_i}{\|\hat{c}_i\|} \tag{8}$$

④对类表示向量进行形变 使短向量接近0，长向量模长近1. 方向不变

The last step in every iteration is to adjust the logits of coupling coefficients b_{ij} by a "routing by agreement" method. If the produced class candidate vector has a large scalar output with one sample prediction vector, there is a top-down feedback which increases the coupling coefficient for that sample and decreases it for other samples. This type of dynamic routing is far more effective and robuster than summing or averaging the sample vectors. Each b_{ij} is updated by:

$$b_{ij} = b_{ij} + \hat{e}^s_{ij} \cdot c_i \tag{9}$$

Formally, we call our induction method as dynamic routing induction and summarize it in Algorithm 2.

⑤利用求得的类向量与样本转换向量相乘求得一致度来更新关联系数

Algorithm 2 Dynamic Routing Induction

```
Require: sample vector $e^s_{ij}$ in support set $S$ and initialize the
    logits of coupling coefficients $b_{ij} = 0$
Ensure: class vector $c_i$
for $iter$ iterations do
    $d_i = softmax\left(b_i\right)$
    for all samples $j = 1, ..., K$ in class $i$:
        $\hat{e}^s_{ij} = W_s e^s_{ij}$
    $\hat{c}_i = \sum_j d_{ij} \cdot \hat{e}^s_{ij}$
    $c_i = \frac{\|\hat{c}_i\|^2}{1+\|\hat{c}_i\|^2}\frac{\hat{c}_i}{\|\hat{c}_i\|}$
    for all samples $j = 1, ..., K$ in class $i$:
        $b_{ij} = b_{ij} + \hat{e}^s_{ij} \cdot c_i$
end for
Return $c_i$
```

4.3 Relation Module

After the class vector c_i is generated by the Induction Module and each query text in the query set is encoded to a query vector e^q by the Encoder Module, the next essential procedure is to measure the correlation between each pair of query and class. Output of the Relation Module is called the relation score, representing the correlation between c_i and e^q, which is a scalar between 0 and 1. Specifically, we use neural tensor layer [Socher *et al.*, 2013] in this module, which has shown great advantages in modeling the relationship between two vectors [Wan *et al.*, 2016; Geng *et al.*, 2017]. We choose it as an interaction function in this paper. The tensor layer outputs a relation vector as follows:

$$v(c_i, e^q) = f\left({c_i}^T M^{[1:h]} e^q\right) \tag{10}$$

where $M^k \in R^{2u\times 2u}, k \in [1, ..., h]$ is one slice of the tensor parameters and f is a non-linear activation function called RELU [Glorot *et al.*, 2011]. The final relation score r_{iq} between the i-th class and the q-th query is calculated by a fully connected layer activated by a sigmoid function.

$$r_{iq} = sigmoid(W_r v(c_i, e^q) + b_r) \tag{11}$$

4.4 Objective Function

We use the mean square error (MSE) loss to train our model, regressing the relation score r_{iq} to the ground truth y_q: matched pairs have similarity 1 and the mismatched pair have similarity 0. Given the support set S with C classes and query set $Q = \{(x_q, y_q)\}_{q=1}^{n}$ in an episode, the loss function is defined as:

$$L\left(S, Q\right) = \sum_{i=1}^{C}\sum_{q=1}^{n}\left(r_{iq} - \mathbf{1}(i == y_q)\right)^2 \tag{12}$$

conceptually we are predicting relation scores, which can be considered as a regression problem and the ground truth is within the space $\{0, 1\}$.

All parameters of the three modules, including the Encoder Module, the Induction Module and the Relation Module are trained jointly by Back Propagation. Specially, the Adagrad

	Training Set	Testing Set
Class Num	159	57
Data Num	195,775	2,279
Data Num/Class	≥ 77	$20 \sim 77$

Table 1: Details of ODIC

Duchi *et al.* [2011] is used on all parameters in each training episode. Our model doesn't need any fine-tuning on the classes it has never seen due to its generalization nature. The induction and comparison ability would be accumulated in the model along with the training episodes.

5 Experiments

We evaluate our model by conducting experiments on two few-shot text classification datasets. All the experiments are implemented with Tensorflow.

5.1 Experiment Datasets

Amazon Review Sentiment Classification(ARSC): Following Yu *et al.* [2018], we use the multiple tasks with the multi-domain sentiment classification [Blitzer *et al.*, 2007] dataset. The dataset comprises English reviews for 23 types of products on Amazon. For each product domain, there are three different binary classification tasks. These buckets then form $23 \times 3 = 69$ tasks in total. Following Yu *et al.* [2018], we select $12(4 \times 3)$ tasks from 4 domains(Books, DVD, Electronics and Kitchen) as test set, and there are only 5 examples as support set for each labels in the test set. We create 5-shot learning models on this dataset.

Open Domain Intent Classification for Dialog System(ODIC): This dataset is drawn from the log data of a real conversational platform. The enterprises submit various dialogue tasks with lots of intents, but many intents have only few labeled samples, which is a typical few-shot classification scenario. Following the definition of few-shot learning task, we divide the ODIC into training set and testing set, and the labels of the two sets have no intersection. The details of the set partition are shown in Table 1.

5.2 Experiment Setup

Baselines: In this section, the baseline models in our experiments are introduced as follows.

- Match Network: a few-shot learning model using metric based attention method [Vinyals *et al.*, 2016].
- Prototypical Network: a deep matrix based method using sample average as class prototypes [Snell *et al.*, 2017].
- Relation Network: a metric-based few-shot learning model which uses neural network as the distance measurement, and calculate class vector by summing sample vectors in the support set [Yang *et al.*, 2018]. We re-implement Relation Network with the same encoder module and relation module with our work.
- ROBUSTTC-FSL: This approach combines several metric methods by clustering the tasks [Yu *et al.*, 2018].

Model	Mean Acc
Match Network [Vinyals *et al.*, 2016]	65.73
Prototypical Network [Snell *et al.*, 2017]	68.17
Relation Network [Yang *et al.*, 2018]	83.74
ROBUSTTC-FSL [Yu *et al.*, 2018]	83.12
Induction-Network-Attention	84.31
Induction-Network-Routing	**85.47**

Table 2: Comparison of mean accuracy(%) on ARSC

The baseline results on ARSC are reported in Yu *et al.* [2018] and we implemented the baseline modes on ODIC with the same text encoder module. In addition, to compare with the proposed dynamic induction routing method, we also implement the Induction Module in a self-attention method to compare with the proposed dynamic routing method, witch is called Induction-Network-Attention.

Implement Details: We use 300 dim Glove embeddings [Pennington *et al.*, 2014] for ARSC dataset and 300 dim Chinese word embeddings trained by Li *et al.* [2018] for ODIC. We set hidden state size of LSTM $u = 128$, and the attention dim $d_a = 64$. The iteration number $iter$ used in dynamic routing algorithm is 3. The relation module is a neural tensor layer with $h = 100$ followed by a fully connected layer activated by sigmoid. We build 2-way 5-shot models on ARSC following Yu *et al.* [2018], and build episode-based meta training with $C = [5, 10]$ and $K = [5, 10]$ for comparison on ODIC. Besides K sample texts as support set, the query set has 20 query texts for each of the C sampled classes in every training episode. This means for example that there are $20 \times 5 + 5 \times 5 = 125$ texts in one training episode for the 5-way 5-shot experiments.

5.3 Experiment Results

Evaluation Methods: We evaluate the performance by few-shot classification accuracy following previous studies in few-shot learning [Snell *et al.*, 2017; Yang *et al.*, 2018]. To evaluate the proposed model objectively with the baselines, we compute mean few-shot classification accuracies on ODIC over 600 randomly selected episodes from the testing set. We sample 10 test texts per class in each episode for evaluation in both 5-shot and 10-shot scenarios. Note that for ARSC, the support set for testing is fixed by Yu *et al.* [2018], therefore we just need to run the test episode once for each of the target task. The mean accuracy of the 12 target task is compared to the baseline models following Yu *et al.* [2018].

Results: Experiment results on ARSC are illustrated in Table 2. The results of the proposed Induction-Network-Routing is almost 3% higher than the ROBUSTTC-FSL, which was the latest state-of-the-art on this dataset. The reason is that the ROBUSTTC-FSL devotes to building a general metric method by integrating several metrics on the sample level, which can not get rid of the noise among different expressions in the same class. In addition, the task clustering based method used by ROBUSTTC-FSL must base on the relevance matrix, which is too inefficient when applied in real-world scenario where the tasks will change rapidly. On the contrary, our Induction Network is trained in the meta-

Model	**5-way Acc.**		**10-way Acc.**	
	5-shot	10-shot	5-shot	10-shot
Match Network [Vinyals *et al.*, 2016]	82.54	84.63	73.64	76.72
Prototypical Network [Snell *et al.*, 2017]	81.82	85.83	73.31	75.97
Relation Network [Yang *et al.*, 2018]	84.41	86.93	75.28	78.61
Induction-Network-Attention	85.28	87.64	76.48	80.83
Induction-Network-Routing	**87.16**	**88.49**	**78.27**	**81.64**

Table 3: Comparison of mean accuracy(%) on ODIC

learning framework, which is more flexible and its induction ability can be accumulated through different tasks.

We also evaluate our method with a real-world intent classification dataset ODIC. The experiment results are shown in Table 3. We can see that the proposed Induction-Network-Routing achieves better classification results on all of the four experiments. The baseline models [Vinyals *et al.*, 2016; Snell *et al.*, 2017; Yang *et al.*, 2018] can be seen as distance metric learning, where all the learning occurs in representing features and measuring distances on the sample-wise level. This work creatively builds an induction module focusing on the class-wise level representation, which is more robust to variations of samples in the support set. In addition, the result difference between our model and other baselines in the 10-shot scenario is bigger than that in the 5-shot scenario. It is because that in the 10-shot scenario, for the baseline models, the improvement brought by a bigger data size is also diminished by more sample level noises.

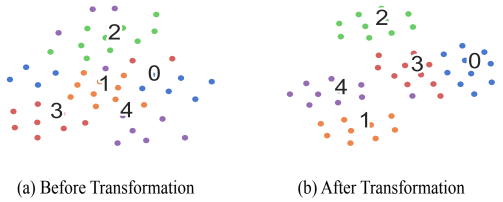
Induction-Network-Routing also outperforms Induction-Network-Attention on both two datasets. Induction-Network-Attention models the induction ability by the attention mechanism and outperforms the baseline models, but the ability is limited by the learned attention parameters. On the contrary, the proposed dynamic induction routing method can capture important information of the class by automatically adjusting the coupling coefficients without attention parameters.

5.4 Experimental Analysis

We further analyze the effect of Transformation and the query text vector visualization to prove the advantage of the Induction Network in our experiments.

Effect of Transformation Figure 2 shows the visualizations of support sample vectors before and after matrix transformation under 5-way 10-shot scenario. In particular, we randomly select a support set with 50 texts (10 texts per class) on ODIC testing set, and get the sample vectors $\left\{e_{ij}^s\right\}_{i=1,...5,j=1...10}$ after encoder module and the sample prediction vector $\left\{\hat{e}_{ij}^s\right\}_{i=1,...5,j=1...10}$. We visualize these two types of vectors with t-SNE [Maaten and Hinton, 2008] respectively. We can clearly see that the vectors after matrix transformation are more separable, demonstrating the effectiveness of matrix transformation to encode semantic relationships between lower level sample features and higher level class features.

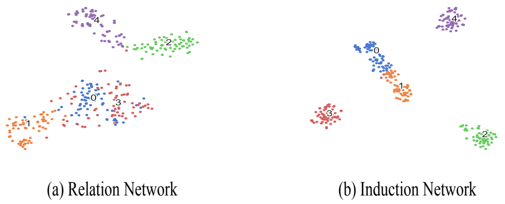
Query Text Vector Visualization We find out that our induction module not only works out fine by generating effective class-level features, but it also helps encoders to learn better text vectors due to it gives different weights to instances and features during backpropagation. Figure 3 shows the t-SNE [Maaten and Hinton, 2008] visualizations of text vectors learnt by Relation Network and our Induction Network under 5-way 10-shot scenario. Specifically, we select 5 classes from ODIC testing set and the embedded texts are then projected into 2-dimensional space using t-SNE. It is clear that the text vectors learnt by Induction Network are better semantically separated than those of Relation Network.



(a) Before Transformation

(b) After Transformation

Figure 2: Effect of Transformation under 5-way 10-shot scenario. (a) The support sample vectors before matrix transformation. (b) The support sample vectors after matrix transformation.



(a) Relation Network

(b) Induction Network

Figure 3: Query text vector visualization learnt by (a) Relation Network and (b) Induction Network.

6 Conclusion

In this paper, we have introduced the Induction Network, a new neural model targeting on few-shot text classification. The proposed model reconstructs the hierarchical representations of support training samples and dynamically induces sample representations into class representations. We combine the dynamic routing algorithm with a typical meta learning framework to simulate human-like induction ability. Results show that our model outperforms other state-of-the-art few-shot text classification models.

References

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447, 2007.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Li Fe-Fei et al. A bayesian approach to unsupervised one-shot learning of object categories. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1134–1141. IEEE, 2003.

Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

Ruiying Geng, Ping Jian, Yingxue Zhang, and Heyan Huang. Implicit discourse relation identification based on tree structure neural network. In *Asian Language Processing (IALP), 2017 International Conference on*, pages 334–337. IEEE, 2017.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

Shaohui Kuang, Junhui Li, António Branco, Weihua Luo, and Deyi Xiong. Attention focusing for neural machine translation by bridging source and target embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1767–1776, 2018.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. Analogical reasoning on chinese morphological and semantic relations. *arXiv preprint arXiv:1805.06504*, 2018.

Weiyao Lin, Yang Mi, Jianxin Wu, Ke Lu, and Hongkai Xiong. Action recognition with coarse-to-fine deep feature integration and asynchronous fusion. *arXiv preprint arXiv:1711.07430*, 2017.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

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

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. 2018.

Tsendsuren Munkhdalai and Hong Yu. Meta networks. *arXiv preprint arXiv:1703.00837*, 2017.

Minlong Peng, Qi Zhang, Yu-gang Jiang, and Xuanjing Huang. Cross-domain sentiment classification with target domain specific information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2505–2513, 2018.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Anthony Rios and Ramakanth Kavuluru. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3132–3142, 2018.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017.

Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934, 2013.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. A deep architecture for semantic matching with multiple positional sentence representations. In *AAAI*, volume 16, pages 2835–2841, 2016.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S Yu. Zero-shot user intent detection via capsule neural networks. *arXiv preprint arXiv:1809.00385*, 2018.

Flood Sung Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. 2018.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. Diverse few-shot text classification with multiple metrics. *arXiv preprint arXiv:1805.07513*, 2018.

Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Suofei Zhang, and Zhou Zhao. Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*, 2018.