Induction Networks for Few-Shot Text Classification

Ruiying Geng[1,2], Binhua Li[2], Yongbin Li[2]* , Xiaodan Zhu[3], Ping Jian[1]*, Jian Sun[2]
[1]School of Computer Science and Technology, Beijing Institute of Technology
[2] Alibaba Group, Beijing
[3] ECE, Queen's University
{ruiying.gry,binhua.lbh,shuide.lyb,jian.sun}@alibaba-inc.com
zhu2048@gmail.com pjian@bit.edu.cn

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 have used meta-learning to simulate the few-shot task, in which new queries are compared to a small support set at the 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 a generalized class-wise representation, by innovatively leveraging the dynamic routing algorithm in meta-learning. In this way, we find the model is able to induce and generalize better. We evaluate the proposed 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, the proposed model significantly outperforms the existing state-of-the-art approaches, proving the effectiveness of class-wise generalization in few-shot text classification.

1 Introduction

Deep learning has achieved a great success in many fields such as computer vision, speech recognition and natural language processing (Kuang et al., 2018). However, supervised deep learning is notoriously greedy for large labeled datasets, which limits the generalizability of deep models to new classes due to annotation cost. Humans on the other hand are readily capable of rapidly learning new classes of concepts with few examples or stimuli. This notable gap provides a fertile ground for further research.

Few-shot learning is devoted 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 (Salamon and Bello, 2017) applied data augmentation and regularization techniques to alleviate the overfitting problem caused by data sparseness, 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 phase, which includes many meta-tasks, following the principle that the testing and training conditions must match. They extract some transferable knowledge by switching the meta-task from mini-batch to mini-batch. As such, few-shot models can classify new classes with just a small labeled support set.

However, existing approaches for few-shot learning still confront many important problems, including the imposed strong priors (Fei-Fei et al., 2006), complex gradient transfer between tasks (Munkhdalai and Yu, 2017), and fine-tuning the target problem (Qi et al., 2018). The approaches proposed by Snell et al. (2017) and Sung et al. (2018), which combine non-parametric methods and metric learning, provide potential solutions to some of those problems. The non-parametric methods allow novel examples to be rapidly assimilated, without suffering from catastrophic overfitting. Such non-parametric models only need to learn the representation of the samples and the metric measure. However, instances in the same class are interlinked and have their uniform fraction and their specific fractions. In previous studies, the class-level representations are calculated by simply summing or averaging represen-

*Corresponding authors: Y.Li and P.Jian.

tations of samples in the support set. In doing so, essential information may be lost in the noise brought by various forms of samples in the same class. Note that few-shot learning algorithms do not fine-tune on the support set. When increasing size of the support set, the improvement brought by a bigger data size will also be diminished by more sample level noises.

Instead, we explore a better approach by performing induction at the class-wise level: ignoring irrelevant details and encapsulating general semantic information from samples with various linguistic forms in the same class. 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.

Recently, capsule network (Sabour et al., 2017) has been proposed, which possesses the exciting potential to address the aforementioned issue. A capsule network uses "capsules" that perform dynamic routing to encode the intrinsic spatial relationship between parts and whole that constitutes viewpoint invariant knowledge. Following a similar spirit, we can regard samples as parts and class as a *whole*. We propose the Induction Networks, which aims to model the ability of learning generalized class-level representation from samples in a small support set, based on the dynamic routing process. First, an Encoder Module generates representations for a query and support samples. Next, an Induction Module executes a dynamic routing procedure, in which the matrix transformation can be seen as a map from the sample space to the class space, and then the generation of the class representation is all depending on the routing-by-agreement procedure other than any parameters, which renders a robust induction ability to the proposed model to deal with unseen classes. By regarding the samples' representations as input capsules and the classes' as output capsules, we expect to recognize the semantics of classes that is invariant to sample-level noise. Finally, the interaction between queries and classes is modelled—their representations are compared by a Relation Module to determine 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 generalizability and scalability to recognize unseen classes.

The specific contributions of our work are listed as follows:

- We propose the Induction Networks for few-shot text classification. To deal with sample-wise diversity in the few-shot learning task, our model is the first, to the best of our knowledge, that explicitly models the ability to induce class-level representations from small support sets.
- The proposed Induction Module combines the dynamic routing algorithm with typical meta-learning frameworks. The matrix transformation and routing procedure enable our model to generalize well to recognize unseen classes.
- Our method outperforms the current state-of-the-art models on two few-shot text classification datasets, including a well-studied sentiment classification benchmark and a real-world dialogue intent classification dataset.

2 Related Work

2.1 Few-Shot Learning

The seminal work on few-shot learning dates back to the early 2000s (Fe-Fei et al., 2003; Fei-Fei et al., 2006). The authors combined generative models with complex iterative inference strategies. More recently, many approaches have used a meta-learning (Finn et al., 2017; Mishra et al., 2018) strategy in the sense that they extract some transferable knowledge from a set of auxiliary tasks, which then helps them to learn the target few-shot problem well without suffering from overfitting. In general, these approaches can be divided into two categories.

Optimization-based Methods This type of approach aims to learn 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-learning architecture called SNAIL which used a novel combination of temporal convolutions and soft attention.

Distance Metric Learning These approaches are different from the above approaches that entail some complexity when learning the target few-

shot problem. The core idea in metric-based few-shot learning is similar to nearest neighbours and kernel density estimation. The predicted probability over a set of known labels is a weighted sum of labels of support set samples. Vinyals et al. (2016) produced a weighted K-nearest neighbour classifier measured by the cosine distance, which was called Matching Networks. Snell et al. (2017) proposed the Prototypical Networks 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 (Sung et al., 2018).

Recently, some studies have been presented focusing specifically on few-shot text classification problems. Xu et al. (2018) studied lifelong domain word embeddings via meta-learning. Yu et al. (2018) argued that the optimal meta-model may vary across tasks, and they employed the multi-metric 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 where there was a known structure over the label space. Xu et al. (2019) proposed a open-world learning model to deal with the unseen classes in the product classification problem. We solve the few-shot learning problem from a different perspective and propose a dynamic routing induction method to encapsulate the abstract class representation from samples, achieving state-of-the-art performances on two datasets.

2.2 Capsule Network

The Capsule Network was first proposed by Sabour et al. (2017), which allowed the network to learn robustly the invariants in part-whole relationships. Lately, Capsule Network has been explored in the natural language processing field. Yang et al. (2018) successfully applied Capsule Network to fully supervised text classification problem with large labeled datasets. Unlike their work, we study few-shot text classification. Xia et al. (2018) reused the supervised model similar to that of Yang et al. (2018) for intent classification, in which a capsule-based architecture is extended to compute similarity between the target intents and source intents. Unlike their work, we propose Induction Networks for few-shot learning, in which we propose to use capsules and dynamic routing to learn generalized class-level representation from samples based. The dynamic routing method makes our model generalize better in the few-shot text classification task.

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 a 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, and extract transferable knowledge that will allow us to deliver better few-shot learning on the support set and thus classify the test set more accurately.

3.2 Training Procedure

The training procedure has to be chosen carefully 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 phase and mimic the few-shot learning setting via episode-based training, as proposed in Vinyals et al. (2016). We construct an meta-episode to compute gradients and update our model in each training iteration. The meta-episode is formed by randomly selecting a subset of classes from the training set first, and then choosing a subset of examples within each selected class to act as the support set S and a subset of the remaining examples to serve as the query set Q. The meta-training procedure explicitly learns to learn from the given support set S to minimise a loss over the query set Q. We call this strategy as episode-based meta training, and the details are shown in Algorithm 1. 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

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**

$$\begin{pmatrix} 159 \\ 5 \end{pmatrix} = 794,747,031 \text{ possible } 5-way \text{ tasks.}$$

4 The Models

Our Induction Networks, 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-episode.

4.1 Encoder Module

This module is a bi-direction recurrent neural network with self-attention as shown in Lin et al. (2017). 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:

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

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

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 $h_t s$ 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:

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

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

$$e = \sum_{t=1}^{T} a_t \cdot h_t \quad (4)$$

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_{ij}^s to class vector c_i:

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

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 $W_s \in R^{2u \times 2u}$ and bias b_s, so that the model is flexible enough to handle the support set at any scale. Each sample prediction vector $\hat{e}_{ij}^s$ is computed by:

$$\hat{e}_{ij}^s = squash(W_s e_{ij}^s + b_s) \quad (5)$$

where $squash$ is a non-linear squashing function through the entire vector, which leaves the direction of the vector unchanged but decreases its magnitude. Given input vector x, $squash$ is defined as:

$$squash(x) = \frac{\|x\|^2}{1 + \|x\|^2} \frac{x}{\|x\|} \quad (6)$$

Eq 5 encodes important invariant semantic relationships between lower level sample features and higher level class features (Hinton et al., 2011). To ensure the class vector encapsulates 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":

$$d_i = softmax\left(b_i\right) \quad (7)$$

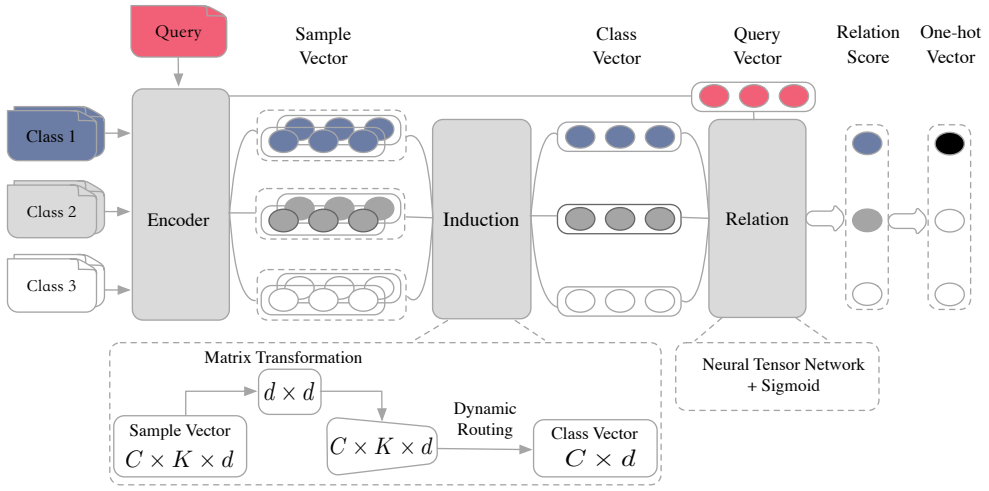


Figure 1: Induction Networks architecture for a C-way K-shot ($C = 3$, $K = 2$) problem with one query example

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:

$$\hat{c}_i = \sum_j d_{ij} \cdot \hat{e}^s_{ij} \tag{8}$$

then a non-linear "squashing" function is applied to ensure that the length of the vector output of the routing process will not exceed 1:

$$c_i = squash(\hat{c}_i) \tag{9}$$

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 adjustment is very effective and robust for the few-shot learning scenario because it does not need to restore any parameters. Each b_{ij} is updated by:

$$b_{ij} = b_{ij} + \hat{e}^s_{ij} \cdot c_i \tag{10}$$

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

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

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 all samples $j = 1, ..., K$ in class i:
  $\hat{e}^s_{ij} = squash(W_s e^s_{ij} + b_s)$
for *iter* iterations **do**
  $d_i = softmax(b_i)$
  $\hat{c}_i = \sum_j d_{ij} \cdot \hat{e}^s_{ij}$
  $c_i = squash(\hat{c}_i)$
  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

Module, the next essential procedure is to measure the correlation between each pair of query and class. The 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 the 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:

$$v(c_i, e^q) = f\left({c_i}^T M^{[1:h]} e^q\right) \tag{11}$$

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.

$$r_{iq} = sigmoid(W_r v(c_i, e^q) + b_r) \quad (12)$$

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:

$$L(S, Q) = \sum_{i=1}^{C} \sum_{q=1}^{n} (r_{iq} - \mathbf{1}(y_q == i))^2 \quad (13)$$

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 are trained jointly by backpropagation. The Adagrad (Duchi et al., 2011) is used on all parameters in each training episode. Our model does not need any fine-tuning on the classes it has never seen due to its generalization nature. The induction and comparison ability are 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 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 the test set, and there are only five examples as support set for each label in the test set. We create 5-shot learning models on this dataset.

	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

Model	Mean Acc
Matching Networks (Vinyals et al., 2016)	65.73
Prototypical Networks (Snell et al., 2017)	68.17
Graph Network (Garcia and Bruna, 2017)	82.61
Relation Network (Sung et al., 2018)	83.07
SNAIL (Mishra et al., 2018)	82.57
ROBUSTTC-FSL (Yu et al., 2018)	83.12
Induction Networks (ours)	**85.63**

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

Open Domain Intent Classification for Dialog System (ODIC) We create this dataset by fetching the log data on a real-world conversational platform. The enterprises submit various dialogue tasks with a great number of intents, but many intents have only a few labeled samples, which is a typical few-shot classification application. Following the definition of the few-shot learning task, we divide the ODIC into a training set and a testing set and ensure that 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.

- Matching Networks: a few-shot learning model using a metric-based attention method (Vinyals et al., 2016).
- Prototypical Networks: a deep metric-based method using sample average as class prototypes (Snell et al., 2017).
- Graph Network: a graph-based few-shot learning model that implements a task-driven message passing algorithm on the sample-wise level (Garcia and Bruna, 2017).
- Relation Network: a few-shot learning model which uses a neural network as the distance metric and sums up sample vectors in the support set as class vectors (Sung et al., 2018).
- SNAIL: a class of simple and generic meta-learner architectures that use a novel combi-

Model	5-way Acc.		10-way Acc.	
	5-shot	10-shot	5-shot	10-shot
Matching Networks (Vinyals et al., 2016)	82.54±0.12	84.63±0.08	73.64±0.15	76.72±0.07
Prototypical Networks (Snell et al., 2017)	81.82±0.08	85.83±0.06	73.31±0.14	75.97±0.11
Graph Network (Garcia and Bruna, 2017)	84.15±0.16	87.24±0.09	75.58±0.12	78.27±0.10
Relation Network (Sung et al., 2018)	84.41±0.14	86.93±0.15	75.28±0.13	78.61±0.06
SNAIL (Mishra et al., 2018)	84.62±0.16	87.31±0.11	75.74±0.07	79.26±0.09
Induction Networks (ours)	**87.16**±0.09	**88.49**±0.17	**78.27**±0.14	**81.64**±0.08

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

nation of temporal convolutions and soft attention (Mishra et al., 2018).

- ROBUSTTC-FSL: This approach combines several metric-based methods by clustering the tasks (Yu et al., 2018).

The baseline results on ARSC are reported in Yu et al. (2018) and we implemented the baseline models on ODIC with the same text encoder module.

Implementation Details We use 300-dimension Glove embeddings (Pennington et al., 2014) for ARSC dataset and 300-dimension Chinese word embeddings trained by Li et al. (2018) for ODIC. We set the hidden state size of LSTM $u = 128$ and the attention dimension $d_a = 64$. 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. In addition to 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.

Evaluation Methods We evaluate the performance by few-shot classification accuracy following previous studies in few-shot learning (Snell et al., 2017; Sung et al., 2018). To evaluate the proposed model with the baselines objectively, 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). Consequently, we just need to run the test episode once for each of the target tasks. The mean accuracy of the 12 target task is compared to the baseline models following Yu et al. (2018).

5.3 Experiment Results

Overall Performance Experiment results on ARSC are presented in Table 2. The proposed Induction Networks achieves a 85.63% accuracy, outperforming the existing state-of-the-art model, ROBUSTTC-FSL, by a notable 3% improvement. We due the improvement to the fact that ROBUSTTC-FSL builds a general metric method by integrating several metrics at the sample level, which faces the difficulty of getting rid of the noise among different expressions in the same class. In addition to that, the task-clustering-based method used by ROBUSTTC-FSL must be found on the relevance matrix, which is inefficient when applied to real-world scenarios where the tasks change rapidly. Our Induction Networks, however, is trained in the meta-learning framework with more flexible generalization and its induction ability can hence be accumulated through different tasks.

We also evaluate our method with a real-world intent classification dataset ODIC. The experiment results are listed in Table 3. We can see that our proposed Induction Networks achieves best classification performances on all of the four experiments. In the distance metric learning models (Matching Networks, Prototypical Networks, Graph Network and Relation Network), all the learning occurs in representing features and measuring distances at the sample-wise level. Our work builds an induction module focusing on the class-wise level of representation, which we claim to be more robust to variation of samples in the support set. Our model also outperforms the latest optimization-based method—SNAIL. The difference between Induction Networks and SNAIL shown in Table 3 is statistically significant under the paired at the 99% significance level. In

	Iteration	Accuracy
Routing+Relation	3	**85.63**
Routing+Relation	2	85.41
Routing+Relation	1	85.06
Routing + Cosine	3	84.67
Sum +Relation	-	83.07
Attention + Relation	-	85.15

Table 4: Ablation study of Induction Networks on ARSC dataset

addition, the performance difference between our model and other baselines in the 10-shot scenario is more significant than in the 5-shot scenario. This is because 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.

Ablation Study To analyze the effect of varying different components of the Induction Module and Relation Module, we further report the ablation experiments on the ARSC dataset as shown in Table 4. We can see that the best performance is achieved when we used 3 iterations, corresponding to the best result reported in Table 2 (more rounds of iterations did not further improve the performance), and the table shows the effectiveness of the routing component. We also changed the Induction Module with sum and self-attention and changed Relation Module with cosine distance. Changes in the performances validate the benefit of both the Relation Module and Induction Module. The Attention+Relation models the induction ability by self-attention mechanism, but the ability is limited by the learnt attention parameters. Conversely, the proposed dynamic routing induction method captures class-level information by automatically adjusting the coupling coefficients according to inputted support sets, which is more suitable for the few-shot learning task.

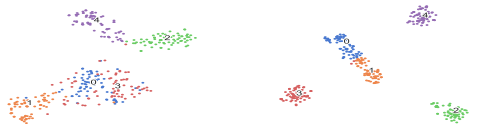
5.4 Further Analysis

We further analyze the effect of transformation and visualize query text vectors to show the advantage of the Induction Networks.

Effect of Transformation Figure 2 shows the t-SNE (Maaten and Hinton, 2008) visualization of support sample vectors before and after matrix transformation under the 5-way 10-shot scenario. We randomly select a support set with 50 texts (10 texts per class) from the ODIC testing set, and obtain the sample vectors $\left\{e^s_{ij}\right\}_{i=1,...5,j=1...10}$ after the encoder module and the sample prediction vector $\left\{\hat{e}^s_{ij}\right\}_{i=1,...5,j=1...10}$ after transformation. We can 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.

(a) Before transformation (b) After transformation

Figure 2: Effect of Transformation under the 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 Networks

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

Query Text Vector Visualization We also find out that our induction module does not only work well in generating effective class-level features, but also helps the encoders to learn better text vectors, as it can give different weights to instances and features during backpropagation. Figure 3 shows the t-SNE (Maaten and Hinton, 2008) visualization of text vectors from the same randomly selected five classes, learnt by the Relation Network and our Induction Networks. It is clear that the text vectors learnt by Induction Networks are better separated semantically than those of Relation Network.

6 Conclusion

In this paper, we propose the Induction Networks, a novel neural model for few-shot text classification. We propose to induce the class-level representations from support sets to deal with sample-wise diversity in few-shot learning tasks. The Induction Module combines the dynamic routing algorithm with a meta-learning framework, and the

routing mechanism makes our model more general to recognize unseen classes. The experiment results show that the proposed model outperforms the existing state-of-the-art few-shot text classification models. We found that both the matrix transformation and routing procedure contribute consistently to the few-shot learning tasks.

Acknowledgments

The authors would like to thank the organizers of EMNLP-IJCNLP2019 and the reviewers for their helpful suggestions. This research work is supported by the National Key Research and Development Program of China under Grant No. 2017YFB1002103, the National Natural Science Foundation of China under Grant No. 61751201, and the Research Foundation of Beijing Municipal Science & Technology Commission under Grant No. Z181100008918002.

References

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. 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.

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

Li Fe-Fei et al. 2003. 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.

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

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org.

Victor Garcia and Joan Bruna. 2017. Few-shot learning with graph neural networks. *CoRR*, abs/1711.04043.

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

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

Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. 2011. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer.

Shaohui Kuang, Junhui Li, António Branco, Weihua Luo, and Deyi Xiong. 2018. 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.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical reasoning on chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 138–143.

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

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

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. 2018. A simple neural attentive meta-learner. *In Proceedings of ICLR*.

Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2554–2563. JMLR. org.

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

Hang Qi, Matthew Brown, and David G Lowe. 2018. Low-shot learning with imprinted weights. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5822–5830.

Anthony Rios and Ramakanth Kavuluru. 2018. 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.

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

Justin Salamon and Juan Pablo Bello. 2017. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283.

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

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

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.

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

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

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3090–3099.

Hu Xu, Bing Liu, Lei Shu, and P Yu. 2019. Open-world learning and application to product classification. In *The World Wide Web Conference*, pages 3413–3419. ACM.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2018. Lifelong domain word embedding via meta-learning. *arXiv preprint arXiv:1805.09991*.

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 the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3110–3119.

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 the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1206–1215.