MCapsNet: Capsule Network for Text with Multi-Task Learning

Liqiang Xiao[1,2], **Honglun Zhang**[1,2], **Wenqing Chen**[1,2], **Yongkun Wang**[3], **Yaohui Jin**[1,2]
[1] State Key Lab of Advanced Optical Communication System and Network, Shanghai Jiao Tong University
[2] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[3] Network and Information Center, Shanghai Jiao Tong University
{jinyh}@sjtu.edu.cn

Abstract

Multi-task learning has an ability to share the knowledge among related tasks and implicitly increase the training data. However, it has long been frustrated by the interference among tasks. This paper investigates the performance of capsule network for text, and proposes a capsule-based multi-task learning architecture, which is unified, simple and effective. With the advantages of capsules for feature clustering, proposed task routing algorithm can cluster the features for each task in the network, which helps reduce the interference among tasks. Experiments on six text classification datasets demonstrate the effectiveness of our models and their characteristics for feature clustering.

1 Introduction

Multi-task learning (MTL) has achieved a great success in the field of natural language processing, which can share the knowledge among multiple tasks, implicitly increasing the volume of training data. The combination of multi-task learning and deep neural networks (DNNs) generates a further synergy via the regularization effect on the DNNs (Collobert and Weston, 2008), which helps alleviate the overfitting and learn a more universal presentation.

Inspired by this, more DNN-based multi-task learning models are proposed to improve the performance. As depicted in Figure 1, they can be categorized into three groups by structure: tree scheme (Collobert and Weston, 2008; Liu et al., 2015), parallel scheme (Liu et al., 2016) and mediate scheme (Zhang et al., 2017). Tree scheme reuses some shallow layers of network and separates the higher layers for different tasks, which is the most common architecture for MTL but can only share the low-level knowledge. To share deeper level knowledge among the tasks, more layers are linked in parallel and mediate schemes.

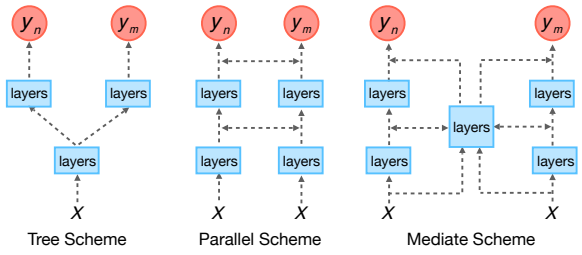

Figure 1: Three schemes of multi-task learning

But this would severely suffer from the interference among tasks. Useless features following the helpful ones are fully shared among tasks, which may contaminate the feature spaces of tasks by useless ones. Besides, models under these two schemes usually employ multiple subnets in the structures, which would contain more parameters and are hard to train.

Apparently, there is a contradiction between knowledge sharing and interference. Sharing too much between tasks would inevitably bring about the interference that feature space for each task may be contaminated by others. Shared useless features may mislead the prediction of network. This dilemma is caused by the lack of management for sharing process, in which the network can not discriminate the features and collect the appropriate features for each task.

Capsule network (Hinton et al., 2011; Mousa et al., 2017; Hinton et al., 2018) embeds the features into capsules and connects the neighbor layers via "routing-by-agreement". The *dynamic routing* algorithm has an ability to decide the routes of capsules, namely, to cluster the features for each category. So, intuitively this property of capsule network can be employed in MTL to discriminate the features for tasks.

In this paper, we explore the performance of capsule network for text (**CapsNet-1**, **CapsNet-1**) and show the benefits and potential of cap-

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4565–4574
Brussels, Belgium, October 31 - November 4, 2018.

sule network for NLP. Then we mainly propose a capsule-based architecture for multi-task learning (**McapsNet**), which is unified, simple, effective and can cluster the feature for tasks. We designed a **Task Routing** algorithm to route the feature flows to tasks and vote for the classes, which can reduce the interference. In extensive experiences, our approach achieves competitive results in single-task scenario and shows obvious improvement in multi-task scenario, which proves our approach effective and its ability to reduce the interference among multiple tasks. Also, our visualization experiments intuitively show the feature clustering mechanism and how it helps make right predictions.

The contribution of this paper are three-folds:

- This paper investigates the performance of capsule network on text and designs two effective capsule-based models for text classification, which give clear improvement to several benchmarks.
- We novelly combine the capsule and multi-task learning, which can help reduce the interference among tasks.
- Proposed task routing algorithm can route the capsules to multiple tasks, by which the features is clustered into groups for tasks.

2 Convolutional Neural Network and Multi-Task Learning

Capsule network is based on the convolutional neural network (CNN) and uses a lot of convolution operations. The main differences between them are that capsule network uses vectors to represent the features and discards the pooling operation. CNN is good at feature extraction and can capture short and long range relations through the feature graph over text sequence (Kalchbrenner et al., 2014; Kim, 2014). In this section, we provide some formulations for CNN and some background knowledge for multi-task learning.

2.1 Single-Task CNN for Text Classification

Given a text sequence $x_{1:l} = x_1 x_2 \cdots x_l$ of length l, the target of CNN is to predict the category $\hat{y}$ of $x_{1:l}$ from a set $\{y_1, y_2, \cdots, y_C\}$, or a one-hot form of $\hat{\mathbf{y}}$, where C is the class number. Using $f(\cdot)$ denote the network, the prediction process can be formalized as $f(x_1, x_2, \cdots, x_l) = \hat{y}$.

For details, convolutional neural network $f(\cdot)$ first uses a lookup table to embed the word sequence $x_{1:l}$ into vectors $\mathbf{x}$. Then CNN produces the representation of the input sequence by stacking the layers of convolution, pooling and fully-connected in order.

$$\mathbf{F} = \mathbf{K} * \mathbf{x} \tag{1}$$

$$\hat{\mathbf{F}} = p(\mathbf{F}) \tag{2}$$

$$\hat{\mathbf{y}} = \mathbf{w}\hat{\mathbf{F}} + \mathbf{b}, \tag{3}$$

where $\mathbf{K}$ is the kernel of convolution operation $*$; $p(\cdot)$ denotes the pooling operation; $\mathbf{F}$ and $\hat{\mathbf{F}}$ represent the feature maps; $\mathbf{w}$ and $\mathbf{b}$ denote the weight and bias respectively in fully connected layer.

The parameters of the network are optimized via all kinds of SGD (stochastic gradient decent) algorithms to minimize the loss between prediction $\hat{\mathbf{y}}$ and ground truth label $\tilde{\mathbf{y}}$

$$l(\hat{\mathbf{y}}, \tilde{\mathbf{y}}) = -\sum_{i=1}^{N}\sum_{j=1}^{C} \tilde{\mathbf{y}}_j^i \log(\hat{\mathbf{y}}_j^i), \tag{4}$$

where i, j enumerate the training samples and classes respectively.

2.2 Multi-Task Learning

Multi-task learning model is usually the variant or combination of single-task ones (CNNs, RNNs or DNNs) like the architectures illustrated in Figure 1. Given K text classification tasks $\{T_1, T_2, \cdots, T_K\}$, a multi-task learning model $f(\cdot)$ shall have the ability to make prediction for samples $\mathbf{x}_i^{(k)}$ from each task T_k.

$$\hat{\mathbf{y}}_i^{(k)} = f(\mathbf{x}_i^{(k)}) \tag{5}$$

And the overall loss for all the tasks is usually a linear combination of the costs for each.

$$L = -\sum_{k=1}^{K} \lambda_k \sum_{i=1}^{N_k}\sum_{j=1}^{C_k} \tilde{\mathbf{y}}_{i,j}^{(k)} \log \hat{\mathbf{y}}_{i,j}^{(k)} \tag{6}$$

where λ_k,N_k and C_k denote the weight, number of training samples and class number of task T_k.

3 Capsule Networks for Text

Capsule network (CapsNet) is first proposed by Sabour et al. (2017) for image classification, which is position sensitive and shows strong performance on some classification tasks. As depicted in Figure 2, we propose several capsule networks for text, which are suitable for text representation and multi-task learning. They are

comprised of convolutional layer, primary capsule layer and class capsule layer. In the rest of this section, we will first give the formulation of single-task capsule networks (CapsNet-1 and CapsNet-2) for text classification, and then transfer it into a multi-task version (McapsNet).

3.1 Primary Capsule Layer

Given an embedded sample of $\mathbf{x} \in \mathbb{R}^{l \times d}$ with length of l and word vectors of d-dimension, capsule network first employs a plain convolution layer to extract the local features from N-grams. Each kernel $\mathbf{K}_i$ with a bias $\mathbf{b}$ emits a feature maps $\mathbf{F}_i$ by convolution.

$$\mathbf{F}_i = \mathbf{x} * \mathbf{K}_i + \mathbf{b} \tag{7}$$

By assembling I feature maps together, we have a I-channel layer

$$\mathbf{F} = [\mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_I] \tag{8}$$

The generated feature maps are then fed into the primary capsule layer, piecing the instantiated parts together via another convolution. Primary capsules use vectors instead of scales to preserve the instantiated parameters for each feature, which can not only represent the intensity of activation but also record some details of the instantiated parts in input. In this way, capsule can be regarded as a short representation of instantiated parts that are detected by kernel.

Sliding over the feature map $\mathbf{F}$, each kernel $\mathbf{K}_j$ would output a series of capsules $p_j \in \mathbb{R}^d$ of d-dimension. These capsules comprise a channel $\mathbf{P}_j$ of primary capsule layer.

$$\mathbf{P}_j = g(\mathbf{K}_j * \mathbf{F} + \mathbf{b}) \tag{9}$$

where g is the nonlinear squash function and b is the capsule bias term. All the J channels can be arranged as

$$\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2, \cdots, \mathbf{P}_J] \tag{10}$$

3.2 Connection Between Capsule Layers

Capsule network generates the capsules in next layer using "routing-by-agreement". This process takes the place of pooling operation that usually discards the location information, which helps augment the robust of the network and also helps cluster features for prediction.

Between two neighbor layers l and $l+1$, a "prediction vectors" $\hat{\mathbf{u}}_{j|i}$ is first computed from the capsule $\mathbf{u}_i$ in lower layer l, by multiplying a weight matrix $\mathbf{W}_{ij}$

$$\hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij}\mathbf{u}_i \tag{11}$$

Then, in the higher layer $l+1$ a capsule $\mathbf{s}_j$ is generated by the linear combination of all the prediction vectors with weights c_{ij}

$$\mathbf{s}_j = \sum_i c_{ij}\mathbf{u}_{j|i} \tag{12}$$

where c_{ij} are coupling coefficients decided by the iterative **dynamic routing** process.

Coupling coefficients are calculated by a "routing softmax" function on original logits b_{ij}, which are the log prior probability that capsule i should be coupled to capsule j.

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_j exp(b_{ij})} \tag{13}$$

This process of "routing softmax" guarantee the sum of all the coefficients for capsule j is 1.

The length of capsule represents the probability that the input sample has the object capsule describes, that is the activation of capsule. So the length of capsule is limited in range $[0, 1]$ with a non-linear squashing function.

$$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1 + \|\mathbf{s}_j\|} \frac{\mathbf{s}_j}{\|\mathbf{s}_j\|^2} \tag{14}$$

By that, the short vectors are pushed to shrink to zero length, and long ones are pushed to one.

3.3 Dynamic Routing

Suppose capsule layer l has been generated. We have to decide the intensity of the connections between capsule i and j from l-th layer to $(l+1)$-th layer, that is the coupling coefficient c_{ij}. The initial digit of coupling coefficient b_{ij} is updated with routing by agreement a_{ij}, which is calculated by a scale product between capsules in two layers.

$$a_{ij} = \hat{\mathbf{u}}_{j|i} \cdot \mathbf{v}_j \tag{15}$$

Value of agreement a_{ij} is added to the digit to calculated the capsules in the next layer.

$$b_{ij} \leftarrow b_{ij} + a_{ij} \tag{16}$$

And the whole process for update (Eq.(13)→(12)→(14)→(15)→(16)) is conducted iteratively to optimize the coupling coefficients and the capsules in the next layer.

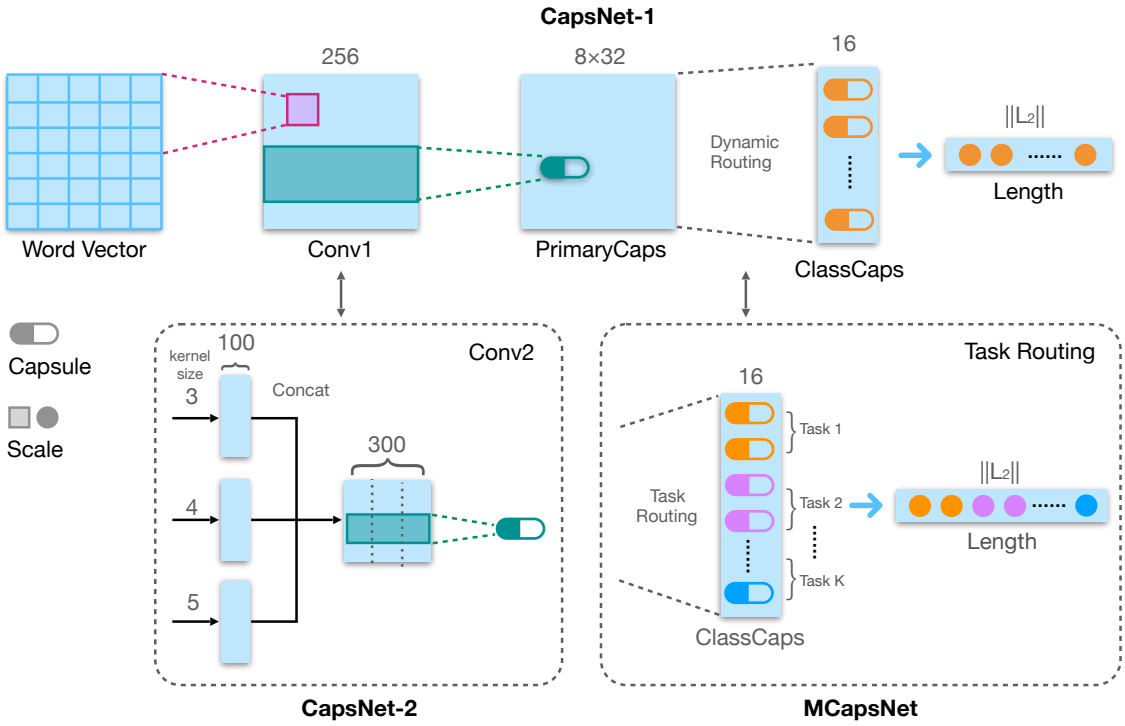

Figure 2: Architectures of capsule networks for text

3.4 Class Capsule Layer and Loss

Class capsule layer, as the top-level layer, is comprised of C class capsules, in which each one corresponds to a category. The length of instantiated parameters in each capsule represents the probability that the input sample belongs to this category, and the direction of each set of instantiated parameters preserves the characteristics of the features, which could be regarded as an encoded vector for the input sample.

Margin Loss To increase the difference between the lengths of classes, CapsNet utilizes a separated margin loss:

$$\begin{aligned} L_j = & G_j \max(0, m^+ - \|\mathbf{v}_j\|)^2 + \\ & \lambda(1 - G_j) \max(0, m^- - \|\mathbf{v}_j\|)^2 \end{aligned} \tag{17}$$

where $\mathbf{v}_j$ is the capsule for class j; m^+ and m^- is the top and bottom margins respectively, which help push the length to shrink beyond two margins; $G_j = 1$ if and only if class j is the ground truth:

$$G_j = \begin{cases} 1 & \tilde{\mathbf{y}}_j = 1 \\ 0 & \tilde{\mathbf{y}}_j = 0 \end{cases} \tag{18}$$

λ is the weight for the absent classes, which reduces the weight of absent classes, avoiding shrinking the lengths of all the capsules too much at prophase training. In this paper, λ is set to 0.5.

Orphan Category A drawback of CapsNet is that it tends to account for everything in the input sampling, including some "background" information such as stop word and punctuations that would interfere the prediction. So an orphan category is added in class capsules in the output layer, which belongs to none of the categories of the task. The orphan category would help collect the less contributive capsules that contain too much "background" information, which reduces the interference for normal categories.

3.5 Substitutional Modules for Multi-Task

Task Routing

Dynamic routing algorithm is first proposed by Sabour et al. (2017), which displaces the pooling operation used in conventional convolution neural network. It maintains the position information for features, which is beneficial to both image and text representation. More importantly, this routing-by-agreement method has an ability to cluster the features into each class.

Inspirited by this, we employ this thought to cluster the features for different tasks and propose the **Task Routing** algorithm, which gives a simple and efficient solution to the problem that existing

Algorithm 1 Task Routing Algorithm

function ROUTING($\hat{\mathbf{u}}_{j|i}^{(k)}, a_{ij}^{(k)}, r, l$)
  for $i = 0 \rightarrow r$ **do**
    for all capsule i in layer l and capsule j in task k:
      $\mathbf{c}_i^{(k)} = softmax(\mathbf{b}_i^{(k)})$
    for all capsule j in layer $l+1$:
      $\mathbf{v}_j^{(k)} = g(\sum_i c_{j|i}^{(k)} \hat{\mathbf{u}}_{j|i}^{(k)})$
    for all capsule i in layer l and capsule j in task k:
      $b_{ij}^{(k)} = b_{ij}^{(k)} + a_{ij}^{(k)}, a_{ij}^{(k)} = \hat{\mathbf{u}}_{j|i}^{(k)} \cdot \mathbf{v}_j^{(k)}$
  end for
return $\mathbf{v}_j^{(k)}$
end function

MTL models (Liu et al., 2017; Ruder et al., 2017; Fang et al., 2017) want to address: *"What feature should be shared and what should not among tasks?"* By that, network can decide the contribution of the features for each tasks and set the appropriate coupling coefficients between features and tasks.

More concretely, we introduce a coupling coefficient $c_{ij}^{(k)}$ between capsule i in l-th layer and capsule j in class capsule $(l+1)$-th layer for task k, which is the result of a softmax function on $\mathbf{b}_{ij}^{(k)}$.

$$c_{ij}^{(k)} = softmax(\mathbf{b}_{ij}^{(k)}) \tag{19}$$

Then, instantiated parameter $\mathbf{v}_j^{(k)}$ of capsule j in task k is calculated by

$$\mathbf{v}_j^{(k)} = \sum_i c_{ij}^{(k)} \cdot \hat{\mathbf{u}}_{j|i}^{(k)} \tag{20}$$

where $\hat{\mathbf{u}}_{j|i}^{(k)} = \mathbf{W}_{k,j,i}\mathbf{u}_i$

Coupling coefficient $c_{ij}^{(k)}$ is restricted in range $[0, 1]$, which represents the probability that capsule i belongs to class capsule j in task k. And it is update by the algorithm is described in Algorithm 3.5.

Multi-Task Loss

The loss for each task is the sum of margin losses for all the classes $\sum_{j=1}^{C} L_j^{(k)}$. By linearly combining the loss for every task, we get multi-task loss

$$L = \sum_{k=1}^{K} \beta^{(k)} \sum_{j=1}^{C} L_j^{(k)}. \tag{21}$$

where $\beta^{(k)}$ is the weight for each loss and $\sum_{k=1}^{K} \beta^{(k)} = 1$. In this paper, all the $\beta^{(k)}$ are set to be $1/K$ to make a balance among K tasks.

Multi-Task Training

In order to juggle several tasks in a unified network, following (Collobert and Weston, 2008), each task is trained alternatively in a stochastic manner. The steps can be described as follows: 1. Pick up a task k randomly; 2. Select an arbitrary sample s from the task k; 3. Feed the sample s into the McapsNet and update the parameters; 4. Go back to 1.

3.6 Architectures of CapsNets for Text

As illustrated in Figure 2, we propose a capsule-based multi-task learning architecture McapsNet, which is base on the single-task structures CapsNet-1 and CapsNet-2. Architectures for them are detailed as following.

CapsNet-1 As depicted in Figure 2, CapsNet-1 is a fundamental framework with three layers. The first layer is a plain convolution operation with 256 kernels with window size of 3 and stride of 1. For activation function, we use ReLU to augment non-linearity. This layer helps extract local features from the input sequences, which is the base to construct primary capsules.

Primary capsule layer employs 32 kernels with window size of 3 and stride of 1. The emitted primary capsules are 8-dimensional, which have bigger respective field, helping reassemble the piece features into wholes.

Last one is the class capsule layer, which is comprised of 16-dimensional capsules for the classes. They are connected to PrimaryCaps with routing-by-agreement and the coupling coefficients are updated by dynamic routing algorithm.

CapsNet-2 On this basis of CapsNet-1, CapsNet-2 upgrades the convolutional layer and uses multiple kernel sizes, which enriches the features. And concatenating[1] them up allows primary capsule see the features with different kernel sizes in the same time.

MCapsNet McapsNet is a unified multi-task structure based on CapsNet-2. It replaces the dynamic routing with task routing (Algorithm 3.5), which enables the network to route the features to

[1]We use padding to ensure the sizes of feature maps are equal.

Dataset	Train	Dev	Test	Classes	Type
MR	9500	-	1100	2	review
SST-1	8544	1101	2210	5	sentiment
SST-2	6920	872	1821	2	sentiment
Subj	9000	-	1000	2	subjectivity
TREC	5900	-	500	6	question
AG's	120k	-	7600	4	news

Table 1: Statistics for six datasets

Dataset	MR	SST-1	SST-2	Subj	TREC	AG's
LSTM	75.9	45.9	80.6	89.3	86.8	86.1
BiLSTM	79.3	46.2	83.2	90.5	89.6	88.2
LR-LSTM	81.5	48.2	87.5	89.9	-	-
VD-CNN	-	-	-	-	-	91.3
DCNN	-	48.5	86.8	-	**93.0**	-
CNN-MC	81.1	47.4	**88.1**	93.2	92.2	-
CapsNet-1	**81.5**	48.1	86.4	**93.3**	91.8	91.1
CapsNet-2	**82.4**	**48.7**	87.8	**93.6**	92.9	**92.3**
- Orphan	81.9	48.3	87.2	93.4	92.6	91.7

Table 2: Single-task results. Row "- Orphan category" denotes a variant of CapsNet-2 without orphan category

each tasks. And the whole network is optimized in a stochastic way with multi-task training (Section 3.5).

Implement Details For word embedding, we use the word vectors in *Word2Vec* (Mikolov et al., 2013), which is 300-dimensional and has 3M vocabularies. And all the routing logits $b_{ij}^{(k)}$ is initialized to zero, so that all the capsules in adjacent layers $(\hat{\mathbf{u}}_{j|i}, \mathbf{v}_j)$ are connected with equal possibility c_{ij}. The coupling coefficients are updated by routing with 3 iterations, which performs best for our approach. For training, we use Adam optimizer (Kingma and Ba, 2014) with exponentially decaying learning rate. Moreover, we use mini-batch with size of 8 for all the datasets.

4 Experiment

We test our capsule-based models on six datasets in both single-task and multi-task scenarios to demonstrate the effectiveness of our approaches. We also in this section conduct some investigations like ablation study and visualization to give a comprehensive understanding to the characteristics of our models.

4.1 Datasets

For both single-task and multi-task scenarios, we conduct extensive experiments on six benchmarks: movie reviews (MR) (Bo and Lee, 2005), Stanford Sentiment Treebank (SST-1 and SST-2) (Socher et al., 2013), subjectivity classification (Subj) (Pang et al., 2004), question dataset (TREC) (Li and Roth, 2002), AG's news corpus (Mousa et al., 2017). These datasets cover a wide range of text classification tasks, which can fully test a model and the details are listed in Table 1.

4.2 Competitors

To demonstrate the effectiveness of our capsule network, we compare the single-task architectures with several state-of-the-art models, involving LSTM/BiLSTM (Cho et al., 2014), LSTM regularized by linguistic knowledge (LR-LSTM) (Qian et al., 2016), very deep network (VD-CNN) (Conneau et al., 2016), dynamic CNN (DCNN) (Kalchbrenner et al., 2014), CNN with multiple channels (CNN-MC) (Kim, 2014). Also, we compare the multi-task architecture (Figure 2) with several strong baselines of multi-task learning, including a general architecture for multi-task learning (MT-GRNN) (Zhang et al., 2017), recurrent neural network based multi-task learning (MT-RNN) (Liu et al., 2016), convolutional neural network with multi-task learning (MT-DNN) (Collobert and Weston, 2008), deep neural network with multi-task learning (MT-CNN) (Liu et al., 2015).

4.3 Single-Task Learning Results

We first test our approach on six datasets for text classification under the scheme of single-task. As Table 2 shows, our single-task network enhanced by capsules is already a strong model. CapsNet-1 that has one kernel size obtains the best accuracy on 2 out of 6 datasets, and gets competitive results on the others. And CapsNet-2 with multiple kernel sizes further improves the performance and get best accuracy on 4 datasets. This proves our capsule networks are effective for text. Particularly, our capsule network outperforms conventional CNNs like DCNN, CNN-MC and VD-CNN with a large margin (by average 1.1%, 0.7% and 1.0% respectively), which shows the advantages of capsule network over conventional CNNs for clustering features and leveraging the position information.

Routing Iteration The coupling coefficients c_{ij} are updated by dynamic routing algorithm, which determines the connections between the capsules. To find the best updating iteration for coupling co-

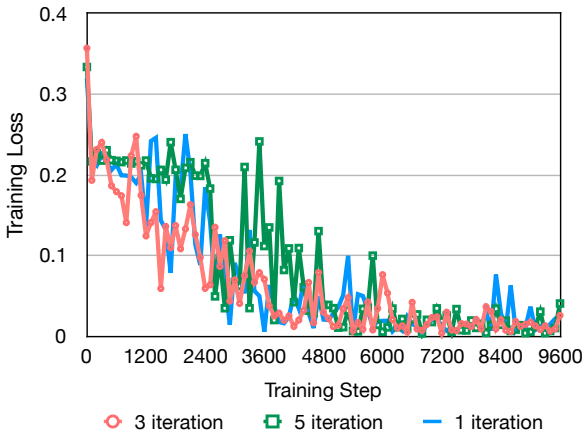

Figure 3: Influence of routing iteration

efficients, we test the CapsNet-2 with a series of iterations (1, 3 and 5) on MR dataset. As shown in Figure 3, network with 3 iterations convergences fast and performs best, which stays in line with the conclusion in (Sabour et al., 2017). So we utilize 3 iterations in all our experiments.

Ablation Study on Orphan Category Orphan category in class capsule layer helps collect the noise capsules that contain the 'background' information like stop words, punctuations or any unrelated words. We conduct the ablation experiment on orphan category, and result (Table 2) shows that network with orphan category perform better than the without one by 0.4%. This demonstrates the effectiveness of orphan category.

4.4 Multi-Task Learning Results

Up to now, we have obtained an optimized single-task architecture. In this section, we equip CapsNet-2 with the *task routing* and multi-task training procedure, namely the model MCapsNet, so that this capsule based architecture can learn several datasets in a unified network. Extensive experiments are conducted in this section to demonstrate the effectiveness of our multi-task learning architecture, as well as its ability for feature clustering.

Multi-Task Performance
We simultaneously train our model McapsNet on six tasks in Table 1 and compare it with single-task scenario (Table 3). We can see that our multi-task architecture clearly improves the performance over the single task models, which demonstrates the benefits of our multi-task architecture.

Dataset	MR	SST-1	SST-2	Subj	TREC	AG's	Avg.△
BiLSTM	79.3	46.2	83.2	90.5	89.6	88.2	+0
MT-GRNN	-	49.2	87.7	89.3	93.8	-	+2.6
MT-RNN	-	49.6	87.9	94.1	91.8	-	+3.5
MT-DNN	82.1	48.1	87.3	93.9	92.2	91.8	+2.9
MT-CNN	81.6	49.0	86.9	93.6	91.8	91.9	+3.0
CapsNet-1	81.5	48.1	86.4	93.3	91.8	91.1	+2.5
CapsNet-2	82.4	48.7	87.8	93.6	92.9	92.3	+3.3
MCapsNet	83.5	49.7	88.6	94.5	94.2	93.8	+4.6

Table 3: Multi-task results of MCapsNet. In column Avg.△, we use BiLSTM as baseline and calculate the average improvements over it.

As Table 3 shows, MCapsNet also outperforms the state-of-the-art multi-task learning models by at least 1.1%. This shows the advantages of our task routing algorithm, which can cluster the features for each task, instead of freely sharing the features among tasks.

4.5 Routing Visualization

To show the mechanism how capsule benefits the multi-task learning, we visualize the coupling coefficient $c_{ij}^{(k)} \in [0, 1]$ between primary and class capsules. We use kernel with size 1 for primary capsule layer so that every capsule represents only one 3-gram phrase. The strength of these connections indicates the importance of these 3-grams to their corresponding task and class.

We feed a random sample from the dataset MR into MCapsNet. In the first row of Table 4, we show the most important 3-gram phrases for two tasks MR and Subj (two classes for each) with word cloud. The sizes of the grams represent the weights of coupling coefficients. We can see that task routing algorithm helps lead the grams into the most related tasks, which allows each task only consider the helpful features for them. In another word, task routing builds a feature space for each task and avoids they contaminate each other. This demonstrates that MCapsNet has the ability of feature clustering, which can benefit MTL by reducing the interference.

We also illustrate the coupling coefficients sequentially for each task. The height of the blue and gray lines represents the polarity of positivity and subjectivity respectively. It is clear that MCapsNet can focus on the appropriate positions for each task, which helps make the final correct predication for every task.

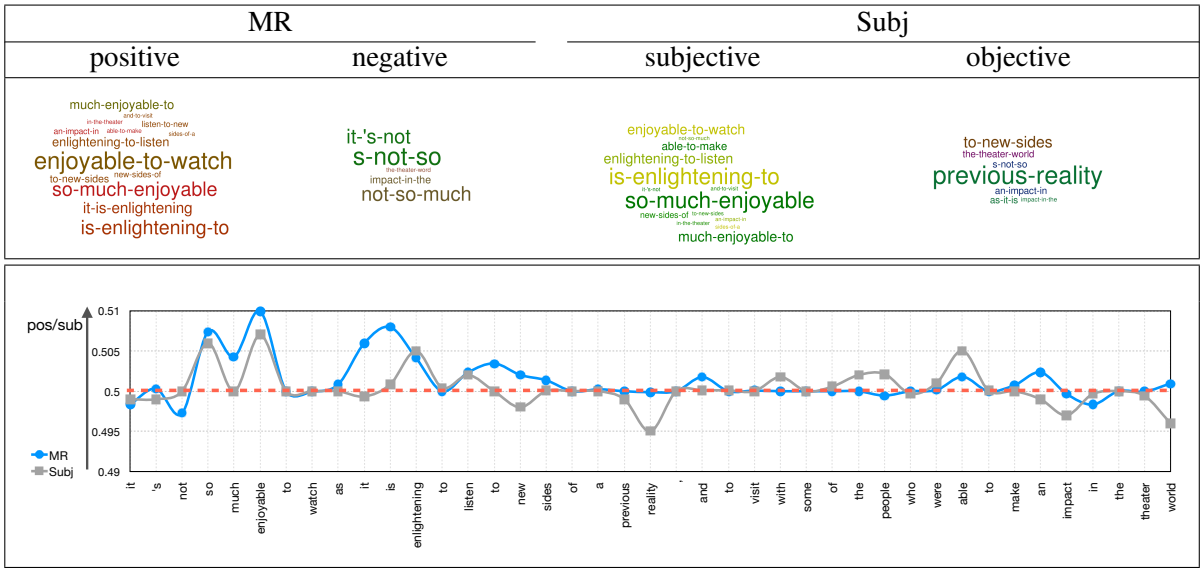

Table 4: Visualization of the task routing for a positive sample from MR, "it 's not so much enjoyable to watch as it is enlightening to listen to new sides of a previous reality , and to visit with some of the people who were able to make an impact in the theater world"

5 Related Work

Related work can be divided into two threads. The first thread is capsule network, which has been proven effective on many classification tasks.

Concept of capsule is first proposed by Hinton et al. (2011), which first use vector to describe the pose of object. This work improves the representation ability of the neural networks against the vanilla CNNs and also enhances the robust of network for transformation. Then dynamic routing algorithm is proposed in (Sabour et al., 2017), which is aimed to displace the pooling operation, building a part-whole relationship for object recognition. Dynamic routing can maintain the position information of features for objects that pooling operations generally discard. And the result shows the proposed method improves the state-of-the-art performance for MNIST dataset. Next, Hinton et al. (2018) employs the matrix to depict the pose and, based on EM algorithm designs a new routing procedure between capsule layers. This work shows strong ability for addressing transformation problem and gains significant improvement on smallNORB dataset.

All these methods are proposed for computer vision, while in this paper we investigate the benefits of capsules for text.

The other thread is about multi-task learning. The earliest idea can be traced back to (Caruana, 1997) and there have been some work completed in this field to augment the performance. Collobert and Weston (2008) develop a multi-task learning model based on CNN. It shares only one lookup table to train a better word embedding. And Liu et al. (2015) propose a DNN-based model for multi-task learning, which shares some low layers but separate the high-level layers to complete several different tasks.

Some models are proposed to share deeper layers of networks, which can exchange high-level knowledge among tasks and gain better performance. (Zhang et al., 2017) and (Liu et al., 2016) introduce some RNN architectures and design different schemes for knowledge sharing. These trials promote the performance of models, but they give no consideration to the interference in multi-task learning. Liu et al. (2017) add the adversarial losses in multi-task RNNs, which can alleviate the interference among tasks by finding a common feature space for tasks. However, the model has multiple subnets and various losses, which requires more computation and training skills.

Different from these methods, we use the thought of capsule in natural language processing (NLP) field. And proposed a capsule based multi-task learning architecture with task routing algorithm. This approach can cluster the features for each task, reducing the interference among them.

6 Conclusion and Future Work

This paper investigates the performance of capsule network for text representation, and proposes several effective architectures. By means of the characteristics of capsule network, we design a unified, sample yet effective architecture with task routing for multi-task learning, which has the ability to clustering the features, building a private feature space for every task.

In future work, we would like to investigate the relations of various tasks in multi-task learning by exploiting the potential of capsule network.

7 Acknowledge

We appreciate the valuable comments from anonymous reviewers. We also thank Xuan Luo for building and maintaining the GPU platforms. And this research was funded by Major State Research Development Program under Grant No. 2018YFC0830400.

References

Pang Bo and Lillian Lee. 2005. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. pages 115–124.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Computer Science*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, pages 160–167.

Alexis Conneau, Holger Schwenk, Loc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for text classification. pages 1107–1116.

Yuchun Fang, Zhengyan Ma, Zhaoxiang Zhang, Xu-Yao Zhang, and Xiang Bai. 2017. Dynamic multi-task learning with convolutional neural network. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1668–1674.

Geoffrey Hinton, Nicholas Frosst, and Sara Sabour. 2018. Matrix capsules with em routing.

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

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 655–665.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Computer Science*.

Xin Li and Dan Roth. 2002. Learning question classifiers. *Coling*, 12(24):556–562.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1–10.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 912–921.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119.

Amr Mousa, Bjrn Schuller, Amr Mousa, Bjrn Schuller, Amr Mousa, and Bjrn Schuller. 2017. Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis. In *Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1023–1032.

Pang, Bo, Lee, and Lillian. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. *Proceedings of Acl*, pages 271–278.

Qiao Qian, Minlie Huang, and Xiaoyan Zhu. 2016. Linguistically regularized lstms for sentiment classification. *CoRR*, abs/1611.03949.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Sgaard. 2017. Sluice networks: Learning what to share between loosely related tasks.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules.

R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank.

Honglun Zhang, Liqiang Xiao, Yongkun Wang, and Yaohui Jin. 2017. A generalized recurrent neural architecture for text classification with multi-task learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 3385–3391.