DOUBLE MULTI-HEAD ATTENTION-BASED CAPSULE NETWORK FOR RELATION CLASSIFICATION

Hongjun Heng[1] and Renjie Li[2]

[1]Department of Computer Science and Technology,
Civil Aviation University of China, Tianjin, China
[2]Sino-European Institute of Aviation Engineering,
Civil Aviation University of China, Tianjin, China

ABSTRACT

Semantic relation classification is an important task in the field of nature language processing. The existing neural network relation classification models introduce attention mechanism to increase the importance of significant features, but part of these attention models only have one head which is not enough to capture more distinctive fine-grained features. Models based on RNN (Recurrent Neural Network) usually use single-layer structure and have limited feature extraction capability. Current RNN-based capsule networks have problem of improper handling of noise which increase complexity of network. Therefore, we propose a capsule network relation classification model based on double multi-head attention. In this model, we introduce an auxiliary BiGRU (Bidirectional Gated Recurrent Unit) to make up for the lack of feature extraction performance of single BiGRU, improve the bilinear attention through double multi-head mechanism to enable the model to obtain more information of sentence from different representation subspace and instantiate capsules with sentence-level features to alleviate noise impact. Experiments on the SemEval-2010 Task 8 benchmark dataset show that our model outperforms most of previous state-of-the-art neural network models and achieves the comparable performance with F1 score of 85.3% in capsule network.

KEYWORDS

Relation Classification, Double Multi-head Attention, Auxiliary BiGRU, Capsule Network.

1. INTRODUCTION

Relation classification is the one of important tasks of Nature Language Processing (NLP), its purpose is to recognize the semantic relation between marked entities in sentence [1], which is premised on entity recognition tasks. For example, in sentence "The suspect dumped the dead <e1>body</e1> into a local <e2>reservoir</e2>.", relation classification is to automatically identify the relation "Entity-Destination" expressed by the given entity pairs marked with HTML. In the field of application, relation classification can be used to enhance the existed knowledge base and create knowledge graphs or ontology knowledge base, from which users can retrieve and use the required knowledge. In addition, relation classification is also widely used in question answering system [2], textual entailment [3] and so on. Accurate relation classification can provide better quality for the above tasks.

Early relation classification methods mainly use machine learning and feature design which usually relies on NLP tools and simple hand-crafted features [4] such as entities' type, distance of

David C. Wyld et al. (Eds): CCSIT, SIPP, PDCTA, AISC, NLPCL, BIGML, NCWMC - 2021
pp. 125-140, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110711

entities and dependency relation path. Recently, deep learning methods such as Convolutional Neural Network [5] (CNN), Recurrent Neural Network (RNN) [6] and other neural network architecture have been widely used for relation classification, these methods do not need to design feature manually and bring a certain performance improvement. Among them, RNN can capture local and global dependency information through gate mechanism. Representative RNN models include Long Short-Term Memory (LSTM) [7] and Gated Recurrent Unit (GRU) [8], which show satisfactory performance in processing sequential tasks, such as machine translation, speech recognition and relation classification especially. However, current RNN models for relation classification only use single layer to capture context features in sentence [9], which could be not enough. Because current NLP models prove that deeper neural network has stronger capability to represent semantic information and improve performance, such as transformer [10], residual network [11], etc. Therefore, it is necessary to explore the deep RNN network structure and improve performance of relation classification.

In order to alleviate the unrelated noise, attention mechanism is introduced to relation classification, which can help focus on important words associated with relation between entities. Frequently used attention includes word-level attention [12] and hierarchical attention [13], the latter is a combination of word-embedding level attention and feature level attention. Besides, multi-head mechanism is also introduced so as to capture distinctive fine-grained features from different representation subspaces, such as self-attention scaled dot product model [13]. However, the above attention models only have one head or single-level multi-head, there is still room to explore multi-level multi-head mechanism, which may help to further capture more distinctive features from sentence. Because sentence in relation classification is normally short, multi-level multi-head is more helpful to explore useful fine-grained information.

Capsule network [14, 15] is a new type of neural network proposed in terms of interpretability in recent years. Different from the previous classification methods, the capsule network combines features into a vector, which is called an instantiated capsule, and classifies by maximizing the length of capsule. Relation classification model based on capsule network has been explored, including CNN-based and LSTM-based capsule networks [16, 17]. The latter performs better than the former, but it has disadvantage. LSTM-based capsule network instantiates capsule through each hidden state of LSTM, but not all hidden states contribute to relation classification. Although some researchers have introduced attention, they do not perform weighted fusion of hidden states, which results in invalid noise fused into capsule and increases the computational complexity of dynamic routing process.

Motivated by above works, we propose the double multi-head attention-based capsule network model for relation classification. In this model, we design an auxiliary bidirectional GRU (BiGRU) architecture to deepen network in time dimension and boost the performance of single BiGRU. Besides, we propose a double multi-head mechanism and decrease the complexity brought by multi-head through max-pooling. Then the word-level features are weighted and merged into sentence-level features. Finally, we instantiate capsules through sentence-level features learned from different representation subspaces, and classify with help of dynamic routing algorithm. The contributions in this article would be summarized as follows:

(1) Firstly, we propose a feature extraction model with auxiliary BiGRU, which can make up for the lack of feature extraction performance of single BiGRU.

(2) Then, we propose a kind of double multi-head attention which enables the model to obtain more distinctive information of sentence from different subspaces.

(3) Our capsule instantiation strategy alleviates the noise fed to capsule network and reduces the complexity of network.

(4) Experimental results on SemEval-2010 Task 8 dataset show that our model achieves a state-of-the-art result with an F1-score of 85.3% in the field of capsule network.

2. RELATED WORK

As one of the methods of supervised learning, the deep learning model can automatically extract hidden features from the input sentence without manually constructing features, so it has received extensive concern from researchers. [18] proposed a Factor-based Compositional Model (FCM), which decomposes annotated sentences and extracts features from them. [19] proposed an enhanced dependency path structure to learn semantic representation. [6] constructed relative dependency features to capture the long-distance relation between entities by using Stanford dependency analysis tools, and used bidirectional LSTM to learn the hidden features and constructed lexical and sentence level features for semantic representation of sentence. [5] proposed to use the Shortest Dependency Path (SDP) to exclude the influence of irrelevant words or phrases, and introduced the negative sampling method into the CNN model to distinguish the directionality of the relation. [20] proposed SDP-LSTM model which uses LSTM to learn subtree feature of root node of SDP. [21] proposed a method of data enhancement using SDP, which uses the inversion of SDP between head and tail entities to add new data.

Since attention mechanism was applied to natural language tasks [22], attention-based models have been widely used in relation classification. [23] proposed context selective attention, using lexical level attention to selectively focus on words related to the target entity; [9] proposed a LSTM model based on attention that focuses on and integrates the word level features extracted by LSTM; [12] proposed a structured recurrent neural network model, which introduces attention into each layer of cascaded RNN network to pay attention to different lexical level features; [13] proposed an attention-based LSTM model, and introduced the multi-head self-attention mechanism proposed by Google Brain [10] in the word embedding layer to capture the meaning between words. At the same time, they added an entity-aware attention after LSTM layer to introduce information about entity as prior knowledge.

Capsule network is proposed to solve the representation limits of CNN and RNN network [14]. [15] replaced the scalar-feature of CNN with capsule and max-pooling with dynamic routing, they achieved the best performance in handwritten digit recognition task. [24] proposed matrix-capsule with EM (Expectation Maximization) routing algorithm, and achieved good performance in shape recognition task. For NLP tasks, [25] and [26] explored capsule networks for text classification. [27] proposed RNN-based capsule network in sentiment analysis. [16] first applied capsule network model to relation extraction, and achieved state-of-the-art performance on distant supervision relation extraction. [17] proposed an attention-based dynamic routing algorithm, which selectively focuses on different capsules for classification.

In this article, we will apply capsule network to relation classification, and explore multilayer RNN architecture, multi-head attention mechanism and instantiation of capsule.

3. MODEL

In this section, we introduce capsule network model based on double multi-head attention in detail. As shown in Figure 1, our model consists of four parts: (1) **Input Representation** layer maps each word in sentence to a fixed-dimensional vector and concatenates other features including relative position and part of speech. (2) **Feature Extraction** layer extracts low-level features from sentence through bidirectional Gated Recurrent Unit (BiGRU), and establishes the dependency relation between words; This layer also uses auxiliary BiGRU to make up for the lack of single BiGRU. (3) **Double Multi-Head Attention** layer calculates the attention weights of the corresponding low-level features, and then selects the most significant features through max-pooling. Double multi-head mechanism is used to capture distinctive fine-grained information from different representation subspace. (4) **Capsule Network** layer divides the sentence-level features of attention layer into low-level capsules, and merges them into high-level capsules (classification capsules) through dynamic routing. Finally, length of capsules is calculated for classification.

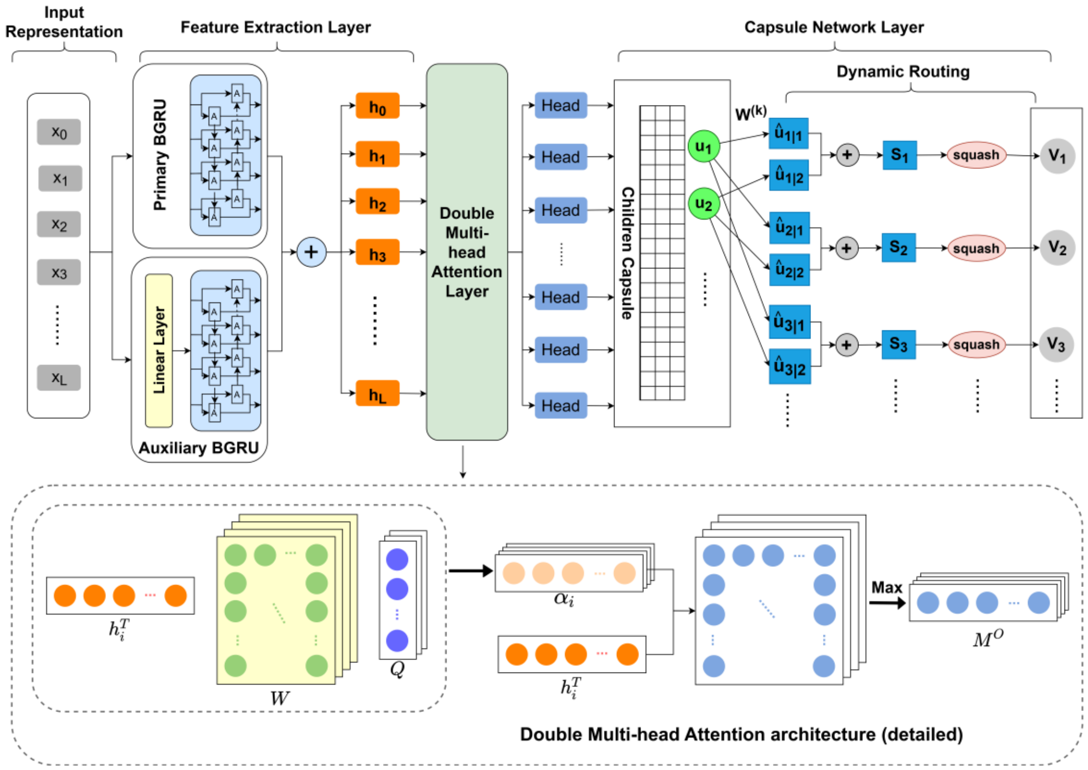


Figure 1. Double Multi-Head Attention-based Capsule Network Model

3.1. Input Representation

Word Embedding. Given a sentence $S = \{w_1, w_2, ..., w_n\}$ containing n words, we need to convert them into numbers that computer can recognize. Traditional method is encoding a word into a vocabulary-size vector through One-Hot, but this vector size is too large and there is no semantic correlation between words. Therefore, we adopt Word2Vec [28] proposed by Google. This method uses a word embedding matrix $W_{word} \in \mathbb{R}^{|V| \times d_w}$ to map each word to a low-dimensional dense vector that contains semantic meaning, where $|V|$ represents size of vocabulary and d_w is the dimension of word vector. In this article, the word embedding matrix is trained using the latest Wikipedia corpus, and the training model is Skip-gram. Finally, each word w_i in sentence is mapped to a vector $w_i^d \in \mathbb{R}^{d_w}$.

Position Embedding. In order to capture additional information about the relation between two target entities, we introduce position feature [29] to represent the relative distance between each word and two marked entities. For the given sentence in section 1, the relative distances between the word "dumped" and the two entities "body" and "reservoir" are respectively -4 and -7.

Therefore, the position embedding of each word w_i relative to two entities is expressed as $w_{i1}^p, w_{i2}^p \in \mathbb{R}^{d_p}$, where d_p is the dimension of the position embedding.

POS Embedding. Part of speech (POS) is the classification of word characteristics at the grammatical level. Adding POS features helps understand the attribute category of each word and identifies the relation between the components of sentence, and improves the robustness of the model. In our experiment, we use the NLTK tool to obtain the POS tags of words. The POS embedding of each word is represented as $w_i^{pos} \in \mathbb{R}^{d_{pos}}$, where d_{pos} is the dimension of the POS vector.

Finally, by concatenating these three types of features, the input representation of each word is $x = [w_i^d, w_i^{pos}, w_{i1}^p, w_{i2}^p]$, where position embedding and POS embedding are uniformly initialized by Xavier method [30].

3.2. Feature Extract

Recurrent neural network is a type of neural network with short-term memory capabilities, which has been widely used in natural language processing tasks. The simplest recurrent neural network only has one hidden layer, called a simple recurrent neural network [31]. However, it has long-term dependency problem and suffers from gradient vanishing and explosion [32], which causes the network to lose its ability to remember long-term information. To solve this problem, gated mechanism [7] is introduced gate to control the speed of information accumulation, including selectively adding new information and selectively forgetting previously accumulated information. The most representative gated recurrent neural networks are LSTM [7] and GRU [8]. Although both can solve the long-term dependency problem, GRU has one less gate than LSTM, and has a smaller computational complexity. Therefore, we use GRU for lexical feature extraction.

GRU controls the flow of information through reset gate and update gate. Note that the input of network is $x_t \in \mathbb{R}^{d_w + d_{pos} + 2d_p}$, where t is current time step, $t \in \{1, 2, ..., L\}$ and L is length of sentence. $h_t \in \mathbb{R}^{d_h}$ is the hidden state at time t, where d_h is dimensionality of hidden state, h_t is updated by equation (1)-(4). Among them, r_t is reset gate that is used to control whether calculation of the candidate state $\tilde{h}_t$ depends on the state h_{t-1} at the previous moment; z_t is update gate that is used to control how much information the current state needs to retain from historical state, and how much new information needs to be received from the candidate state; W_i and U_i ($i \in \{r, z\}$) are weight matrices, $b_i (i \in \{r, z, h\})$ is bias, σ and $\tanh$ are sigmoid and hyperbolic function respectively; $\odot$ is element-wise product, which means the product of the corresponding elements of two matrices; The size of all state vectors is the same as h_t.

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{1}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{2}$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h) \tag{3}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{4}$$

For many sequential tasks, the current output is not only related to the past, but also related to the future. The bidirectional GRU enhances the capability of standard GRU by introducing a network layer that transmits information in reverse order of time. Therefore, we use BiGRU to capture the global sequential characteristics, final hidden state $\boldsymbol{h}_t^p \in \mathbb{R}^{d_h}$ can be expressed as $\boldsymbol{h}_t^p = [\vec{h}_t \oplus \overleftarrow{h}_t]$, which is the element-wise sum of the forward state and the backward state. The size of the hidden state $\boldsymbol{h}_t^p$ is determined by hyperparameter d_h.

In recent years, deep neural networks have shown more excellent performance in tasks such as image recognition and machine translation. Inspired by the residual network [11], considering the characteristics of the strong fitting ability of RNN, we propose the parallel auxiliary recurrent neural network structure. Because RNN has a strong ability to fit data, the vertical stacking of traditional residual network structure is likely to cause serious overfitting, so the parallel structure is adopted to deepen the number of layers of the network at the time dimension. As shown in Figure 1, the feature extraction layer contains two layers of BiGRU. The upper layer is called primary BiGRU, which models the original sequence and outputs the hidden features $\boldsymbol{h}_t^p$. The lower layer is auxiliary BiGRU, which is merged into primary BiGRU in parallel to enhance the feature extraction performance of single BiGRU.

The auxiliary BiGRU layer consists of a linear layer, a BiGRU and a nonlinear activation function tanh. Auxiliary BiGRU receives the linear transformation of the input, after non-linear activation and encoding by itself, it outputs the hidden layer state $\boldsymbol{h}_t^a$, then accepts activation of *relu*, and finally is summed with $\boldsymbol{h}_t^p$. The above can be described as equation (5) and equation (6):

$$\boldsymbol{h}_t^a = BiGRU(\tanh(f(x_t))) \tag{5}$$

$$\boldsymbol{h}_t = \tanh(\boldsymbol{h}_t^p + relu(h_t^a)) \tag{6}$$

Where *BiGRU* means process of equation (1)-(4), *f* is linear transformation and $\boldsymbol{h}_t \in \mathbb{R}^{d_h}$ is the output of feature extraction layer. The purpose of the auxiliary BiGRU is to learn the features lost by the primary BiGRU, and augments the capability of feature extraction layer.

3.3. Double Multi-head Attention

When bidirectional GRU deals with sequence, sentence can be encoded as a vector representation as time step *t* progresses. However, the length of the sentence is changeable. For a sentence that is too long, a single vector will lose the information at the head or tail of sentence. So, we retain the hidden vectors of BiGRU at each moment, and selectively weights and fuses the hidden features $H = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_L\}$ through the attention mechanism. Attention distribution α_i of every hidden feature $\boldsymbol{h}_i$ can be expressed as equation (7):

$$\alpha_i = \mathrm{softmax}(s(\boldsymbol{h}_i, q)) \tag{7}$$

Where *q* is relation query vector, $s(\boldsymbol{h}_i, q)$ is attention score function and softmax is used to normalize the score. We use bilinear attention score function, as shown in equation (8). Where *W* is a learnable bilinear matrix.

$$s(\boldsymbol{h}_i, q) = \boldsymbol{h}_i^T W q \tag{8}$$

In order to learn sentence from different subspaces, we introduce a double multi-head mechanism. Multi-head is introduced into bilinear matrix *W* and query vector *q*. The attention score function with double multi-head is shown in equation (9).

$$s(\boldsymbol{h}_i, Q) = \boldsymbol{h}_i^T W Q \tag{9}$$

Where $Q \in \mathbb{R}^{d_q \times d_c}$ is relation query matrix that is composed of d_c relation query vector with size of d_q ; The matrix *W* becomes a three-dimension matrix from original two dimension, that means $W \in \mathbb{R}^{d_a \times d_h \times d_q}$, where d_a is a hyper parameter and represents number of multi-head.

Double multi-head mechanism is embodied by matrix *W* and *Q*, but it makes multi-head nested, which means the parameter amount of attention layer is increased to $d_a \times d_c$ times of original bilinear attention. In order to reduce complexity of network, we use maximum pooling operation, as shown in equation (10).

$$M^o = \max(\sum\nolimits_{i=1}^{L} \alpha_i \boldsymbol{h}_i) \tag{10}$$

Where $M^o \in \mathbb{R}^{d_a \times d_h}$ is final output of double multi-head attention, the maximum pooling operation maximizes the weighted hidden layer features, which highlights the most salient features that have been paid attention. After fusion at each time t, d_a types of vector representations of sentence are output.

3.4. Capsule Network

Primary capsule: After the output of double multi-head attention, we need to resolve problem of how to instantiate capsule. For capsule network proposed by [15], capsule of a group of neurons whose activity vector represents the instantiation parameters of a specific type of entity. In our work, capsule is the instantiation parameters of relation and built by sentence-level features. We combine *d* neurons into a capsule $u_i \in \mathbb{R}^d$, and obtain $d_a \times m$ primary capsule (children capsule) by splitting the matrix M^o where *m* is the division of d_h by *d*. The equation (11) is the list of children capsules:

$$U = [u_1, u_2, ..., u_{d_a \times m}] \in \mathbb{R}^{d_a \times m \times d} \tag{11}$$

$$\boldsymbol{u}_i = squash(u_i) = \frac{\| u_i \|^2}{0.5 + \| u_i \|^2} \frac{u_i}{\| u_i \|} \tag{12}$$

After obtaining the primary capsule, the length of capsule is squeezed into 0 and 1 by the activation of squash function in equation (12), because capsule network uses the length of capsule to represent the probability of relation classification.

Dynamic routing: The basic idea of dynamic routing is to map appropriate children capsules to parent capsules through non-linear loop iteration. We need a linear transformation on the children capsule to generate prediction vector $\hat{\boldsymbol{u}}_{j|i} \in \mathbb{R}^d$, where *i* and *j* are respectively children capsules and parent capsules. The linear transformation is realized by equation (13):

$$\hat{\boldsymbol{u}}_{j|i} = W_j^t \boldsymbol{u}_i + \hat{b}_{j|i} \tag{13}$$

Where $W_j^t \in \mathbb{R}^{I \times J \times d \times d}$ is a non-shared weight matrix and $\hat{b}_{j|i} \in \mathbb{R}^{I \times J \times d}$ is bias, *I* and *J* represent the number of children capsules and parent capsules respectively; Here $I = d_a \times m$ and *J* is the number of relation types.

See **Algorithm 1** and Figure 2 for dynamic routing, this algorithm controls the connection strength between children capsules and parent capsules through the coupling coefficient $c_{j|i}$ that is initialized uniformly, which means each children capsule is treated equally in first iteration;

then the coefficient is adjusted to select appropriate children capsules through later iteration. However, not all children capsules are effective for relation classification and there is still interference from noisy children capsules [26]. Therefore, we replace original softmax with leaky-softmax to update the connection strength, which is used to route noisy children capsules to additional dimensions.

Algorithm 1 Dynamic Routing Algorithm

```
procedure ROUTING ($\hat{u}_{j|i}, r, l$)
    for all capsule $i$ in layer $l$ and capsule $j$ in layer $(l+1)$: $b_{j|i} = 0$ .
    for $r$ iterations do
        for all capsule $i$ in layer $l$: $c_{j|i} = \text{leaky-softmax}(b_{j|i})$
        for all capsule $j$ in layer $(l+1)$: $v_j = squash(\sum c_{j|i}\hat{u}_{j|i})$
        for all capsule $i$ in layer $l$ and capsule $j$ in layer $(l+1)$:
            $b_{j|i} = b_{j|i} + \hat{u}_{j|i} \cdot v_j$
    return $a_j = \|v_j\|$
```

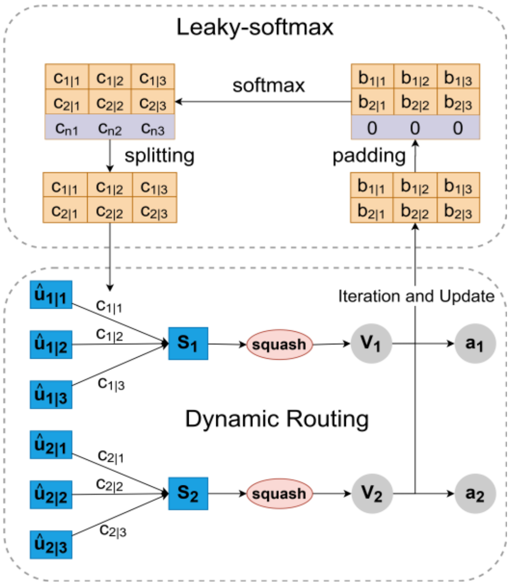


Figure 2. Dynamic routing with leaky-softmax

3.5. Training Procedure

For capsule network, the length of the instantiation vector v_j is used to represent the probability of relation. Unlike traditional neutral networks that use cross entropy loss, we use separate margin loss to calculate the loss L_j of each relation capsule *j*. In order to alleviate the overfitting of network, we use dropout [33] and L2 regularization. Dropout method improves the performance of the neural network by preventing the joint action of feature detectors during the forward propagation process. L2 regularization limits the weight update during the backward propagation process. In our model, we use dropout mechanism on the embedding layer and feature extraction layer. The loss function of each relation is shown in equation (14).

$$L_j = Y_j \max(0, m^+ - a_j)^2 + \lambda_1 (1 - Y_j) \max(0, a_j - m^-)^2 + \lambda_2 \| \theta \|_F^2 \tag{14}$$

Where $Y_j = 1$ if relation j is present; m^+ and m^- are threshold, λ_1 is the penalty rate for false positive and false negative, these three empirical parameters are usually set to 0.9, 0.1 and 0.5; λ_2 is coefficient of L2 regularization, θ represents weight parameters of our network (except for capsule network), $\|\cdot\|_F$ represents Frobenius norm. The total loss is sum of losses for all relations.

4. Experiment

4.1. Dataset and Experimental Setup

Dataset: Our experiment adopts the public dataset SemEval-2010 Task 8 [1], which contains 9 types of relation and an “Other” type. The 9 types are Cause-Effect, Component-Whole, Content-Container, Entity-Destination, Entity-Origin, Instrument-Agency, Member-Collection, Message-Topic, Product-Producer; “Other” type is not of any of these nine types. In our experiment, we do not distinguish the direction of relations, so the total number of relations is 10. SemEval-2010 Task 8 dataset consists of 8000 sentences for training and 2717 sentences for testing. In order to compare our results with previous state-of-the-art models, we adopt precision P, recall R and F1 score to evaluate performance between our model and others. The definition of three metrics is shown in equation (15)-(17). The macro precision, macro recall and macro F1 are respectively the average of precision, recall and F1 of all relation categories.

$$P = \frac{TP}{TP + FP} \tag{15}$$

$$R = \frac{TP}{TP + FN} \tag{16}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{17}$$

Setup: we randomly select 800 samples from the training set as development set for tuning hyperparameters. The best hyperparameters are shown in Table 1. Our experiment adopts 300-dimensional word vector pretrained in the latest Wikipedia corpus. Feature extraction layer of our model uses orthogonal initializer, other weights of network are initialized by Xavier method [30]. We train our model by Pytorch framework on platform Ubuntu and use one Geforce GTX 1650.

Table 1. Hyperparameters

Parameter	**Description**	**Value**
B	Size of batch	32
d_h	Hidden size of GRU	256
d_w	Size of word embedding	300
d_p	Size of position embedding	40
d_{pos}	Size of POS embedding	30
d_q	Size of query vector	17
d_a	Number of first head (W)	8
d_c	Number of second head (Q)	10
d	Size of capsule	8
lr	Learning rate	0.001
λ_2	Weight decay	0.0001
$dropout$	Embedding layer dropout	0.5
	Feature extraction layer dropout	0.1

4.2. Overall Experiment

Table 2 compares our double multi-head attention-based capsule network model with other four types of state-of-the-art models. Among the non-neural network models, the top is the support vector machine (SVM) [34]. This model uses manually created features and SVM classifier for relation classification, and achieves the best performance (82.2%) during the official competition. Models based on the Shortest Dependency Path (SDP) show excellent performance, including FCM [18], DepNN [19], depLCNN+NS [5], SDP-LSTM [20], BLSTM [6], DRNN [21]. SDP can ignore unrelated words between entities and construct a semantically directly related dependency path, which helps the model capture the dependency relationship between words more quickly. However, building of dependency tree often resorts to existing NLP tools, it is not always accurate and affected by sentence length, which costs time a lot. Introduction of attention has brought a very effective improvement to relation classification. By selectively assigning different weights, it highlights the most important words of sentence. Representative models are Hier-BLSTM [12], Att-BLSTM [9], Attention-CNN [23] and EA-BLSTM [13]. Recently, the capsule network model, which has received widespread concern in the field of image classification, has been used in NLP tasks, and a series of variants have been produced. Among them, [17] proposed a capsule network Att-CapNet, good results have been achieved with an F1 score of 84.5%.

Double multi-head attention-based capsule network proposed by us achieves an F1 score of 85.3% on SemEval-2010 Task 8 dataset. Although the performance of our model is not the best, it outperforms most other models without using external features like WordNet and SDP. Besides, compared with the capsule network relation classification model, our model achieves state-of-the-art result.

Table 2. Comparison with previous models on SemEval-2010 Task 8 (WAN represents words around nominals)

Models	**Macro F1(%)**
Non neural model	
SVM	**82.2**
SDP Model	
FCM	83.0
DepNN	83.6
depLCNN+NS	**85.6**
SDP-LSTM	83.7
BLSTM	84.3
DRNN	**86.1**
Attention-based Model	
Hier-BLSTM	84.3
Att-BLSTM	84.0
Attention-CNN	84.3
+WordNet, WAN	**85.9**
EA-BLSTM	84.7
Capsule Network Model	
Att-CapNet	84.5
Our model	**85.3**

For fair comparison with other models, we implement four of these models and use the same data pre-processing method and pretrained word vectors, which ensures that the input of each model is the same. Table 3 shows the result of precision, recall and F1 score of BLSTM [6], Att-BLSTM [9], Attention-CNN [23], Att-CapNet [17] and our model. It shows that the macro precision of our model is lower than that of Att-CapNet [17], but the macro recall exceeds others by 2.2%-4.1%, so the macro F1 score is increased by 0.9%-2.1%. According to the analysis above, we believe that our model is superior to the comparative models.

In order to explore the recognition effect of models on each relation, Table 4 lists the F1 score of five types of models for all relations (except Other). The comparison results show that our model is less effective in identifying “Component-Whole”, “Entity-Origin” and “Member-Collection”, F1 is lower than Att-CapNet and BLSTM. However, our model is better than other models in recognizing other relations, which has a greater contribution to the metric of macro-average F1 score.

Table 3. Fair comparison between our model and other four models

Models	**Macro P (%)**	**Macro R (%)**	**Macro F1 (%)**
BLSTM	81.7	87.3	84.3
Att-BLSTM	80.7	86.6	83.5
Attention-CNN	81.2	85.4	83.2
Att-CapNet	**82.4**	86.6	84.4
Our model	81.8	**89.5**	**85.3**

Table 4. Comparison of F1 (%) for each relation type

Relation Types	**BLSTM**	**Att-BLSTM**	**Attention-CNN**	**Att-CapNet**	**Our model**
Cause-Effect	92.9	90.7	91.4	92.0	**93.6**
Component-Whole	79.7	81.8	80.9	**83.3**	81.9
Content-Container	86.3	86.2	84.5	86.2	**86.8**
Entity-Destination	88.3	89.7	88.2	89.7	**91.0**
Entity-Origin	**85.8**	84.8	85.5	85.2	84.7
Instrument-Agency	74.7	72.8	73.5	74.2	**76.0**
Member-Collection	**85.1**	83.0	84.1	84.6	82.4
Message-Topic	83.0	84.5	82.7	85.5	**88.1**
Product-Producer	82.6	77.6	78.0	78.3	**83.4**

4.3. Ablation Study

In order to reflect the effects brought by auxiliary BiGRU, double multi-head attention and capsule instantiation strategy, we conduct an ablation study. The multiple variants derived from the model are shown in Table 5, we remove some components in our original model successively. “No multi-head (*W*)” and “No multi-head (*Q*)” respectively represent the situations of only removing multi-head of *W* and multi-head of *Q*; “No multi-head (*W* and *Q*)” represents removal of all multi-head of *W* and *Q* which means that our multi-head attention becomes the basic bilinear attention. "No caps-ins-strategy" represents that we remove our capsule instantiation strategy.

Comparing the four models in the first, the second, the sixth and the last rows, it shows that our auxiliary BiGRU, double multi-head attention and capsule instantiation strategy effectively improve the overall performance. In specific, the auxiliary BiGRU boosts the precision P, double multi-head attention has a greater improvement in recall R, which slow down the impact of the decline in precision, so F1 is increased. Capsule instantiation strategy increases the precision. Comparing "No multi-head (*W*)", "No multi-head (*Q*)" and "No multi-head (*W* and *Q*)", results show that single multi-head does not improve the model, because improvement of recall is lower than impact of precision. But composition of these two multi-head brings an improvement of 0.6% for F1.

Table 5. Comparison with all variants in ablation study

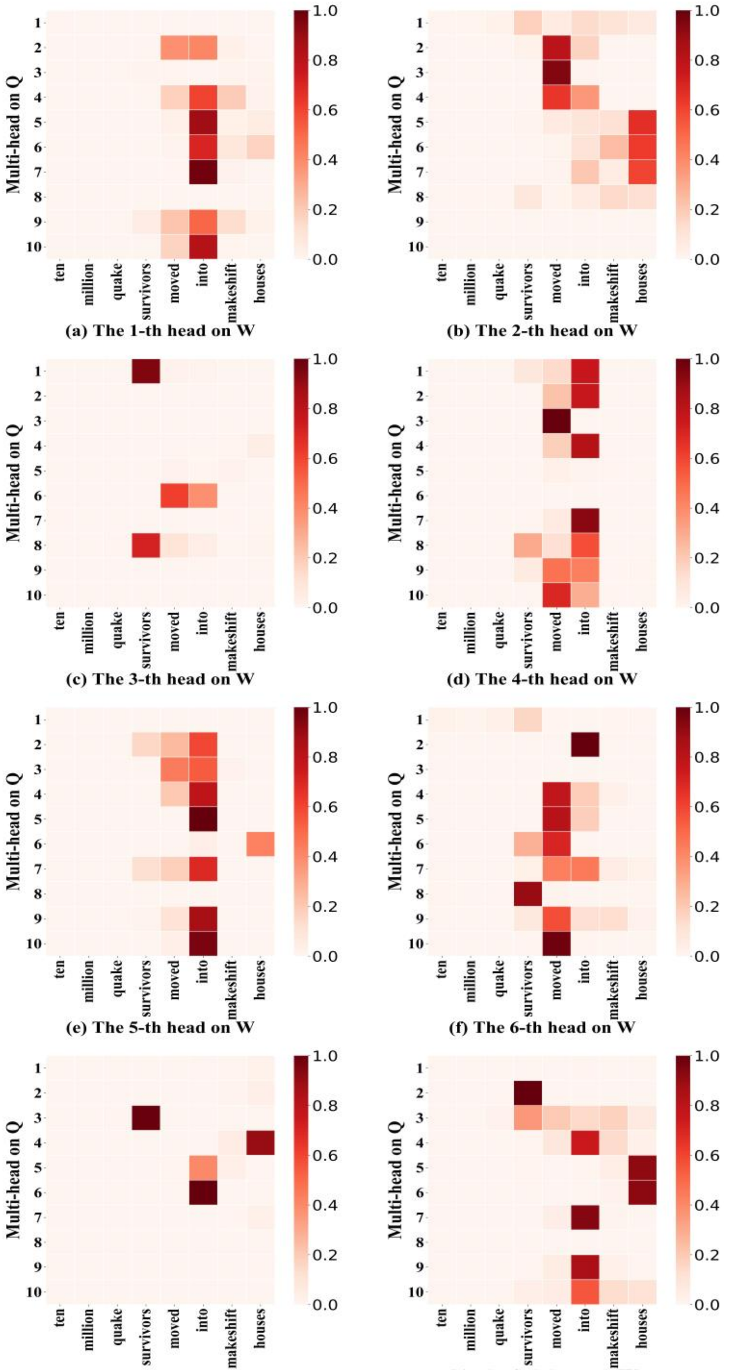
Models	**Macro P (%)**	**Macro R (%)**	**Macro F1 (%)**
Our model (original)	81.8	89.5	85.3
No auxiliary BiGRU	80.8	89.6	84.9
No multi-head (*W*)	80.7	88.4	84.3
No multi-head (*Q*)	80.4	87.9	83.9
No multi-head (*W* and *Q*)	83.0	85.7	84.3
No attention layer	81.8	85.5	83.5
No caps-ins-strategy	78.9	85.9	82.2

4.4. Analysis of Double Multi-head Attention

Local analysis: Local analysis is to understand how the model makes decisions for a certain sample or group of samples. Figure 3 visualizes the attention weight α_i (see equation (7)) through the heat matrix diagram, which shows the importance of different words in a sentence for relation classification. The greater the importance of the word, the greater the attention score given to it, and the darker the corresponding colour in heat map. The 8 sub-graphs in Figure 3 respectively represent 8 heads on the bilinear matrix W, the vertical axis of sub-graph shows the 10 heads on the query matrix *Q*, the horizontal axis represents words in sentence, and the colour-bar on the right side indicates the size of attention score, which is between 0 and 1.

In order to explain the double multi-head attention more clearly, we only focus on the darker colour of each head (attention score greater than 0.6). Take the sentence provided in Figure 3 as an example, the two entities "survivors" and "houses" express relation "Entity-Destination"; In sub-figure (a), the 10 heads mainly focus on "into"; For other sub-figures, "moved", "survivors" and "houses" are the main objects focused by attention layer. We can find that multi-head attention proposed in this paper focuses on the two entities and words expressing their relation, which is consistent with the focus of human. Therefore, our attention model finally distinguishes the sentence as relation "Entity-Destination".

Global analysis: Global analysis is to explain the semantically meaningful components in the model and to understand how the model makes decisions on the entire dataset. According to the local analysis method, we extract the words (except for two entities) in sentence on the Testing set, and performs statistics according to the 9 types of relations; Due to space limitation, only the top four words with the largest frequency in each relation type are given, the statistical results are shown in Table 6. We can find that for each type of relation, our attention model can identify the key words that express their relation. For example, the key words expressing the relation "Entity-Origin" are "derived", "from", etc., and for relation "Message-Topic", they are "about", "on", etc. In addition, we also count the proportion of entity pairs that our attention focuses on under each type of relation, and Table 6 shows that more than 90% of entity pairs can be captured by our attention. Thus, our double multi-head attention can identify the common patterns (feature words) of specific relation, and provides strong support for model's further decision-making.



(a) The 1-th head on W

(b) The 2-th head on W

(c) The 3-th head on W

(d) The 4-th head on W

(e) The 5-th head on W

(f) The 6-th head on W

(g) The 7-th head on W

(h) The 8-th head on W

Figure 3. Heat Map of attention weight matrix for sentence “ten million quake Survivors moved into makeshift houses”

Table 6. The top four words with highest frequency and rate of entity pairs focused by the attention layer

Relation Types	Words focused	Rate of Entity pairs
Cause-Effect	Caused, by, from, cause	91.5%
Component-Whole	Of, with, in, has	96.2%
Content-Container	In, was, inside, with	80.2%
Entity-Destination	Into, to, put, in	88.4%
Entity-Origin	from, derived, of, away	95.0%
Instrument-Agency	With, using, of, by	98.1%
Member-Collection	Of, in, into, was	96.6%
Message-Topic	In, to, on, about	84.3%
Product-Producer	By, of, from, with	99.6%
Total	-	92.1%

5. CONCLUSIONS

We propose a double multi-head attention-based capsule network model for relation classification and auxiliary BiGRU that improves capability of single BiGRU for feature extraction. Our model achieves F1 score of 85.3% on SemEval-2010 Task 8 dataset using only word embedding, relative position embedding and POS embedding, and outperforms most of previous study. Ablation study shows that proposed auxiliary BiGRU, double multi-head attention and capsule instantiation strategy are effective. In addition, we analyse how the double multi-head attention highlights the words that contribute to relation classification from the local and global perspectives, as well as the common pattern recognition mechanism for specific relation types. In the future, we will use large-scale pre-trained language models such as Bert to further improve performance, and explore the potential of our model in the joint extraction of entity and relation as well as event extraction.

ACKNOWLEDGEMENTS

We would like to express our gratitude to the anonymous reviewers for their hard work and kind comments, which will further improve our work in the future.

REFERENCES

[1] I. Hendrickx et al., (2010) “SemEval-2010 Task 8: Multi-way classification of semantic relations between pairs of nominals”, in Proceedings of the 5th International Workshop on Semantic Evaluation, pp33–38.

[2] D. Ravichandran and E. Hovy, (2002) “Learning surface text patterns for a question answering system”, in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, No. July, pp41–47.

[3] I. Szpektor, H. Tanev, I. Dagan, and B. Coppola, (2004) “Scaling web-based acquisition of entailment relations”, in Proceedings of EMNLP, Vol. 4, No. March, pp41–48.

[4] F. M. Suchanek, G. Ifrim, and G. Weikum, (2006) “Combining linguistic and statistical analysis to extract relations from web documents”, in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, Vol. 2006, pp712–717.

[5] K. Xu, Y. Feng, S. Huang, and D. Zhao, (2015) “Semantic relation classification via convolutional neural networks with simple negative sampling”, in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp536–540.

[6] S. Zhang, D. Zheng, X. Hu, and M. Yang, (2015) “Bidirectional long short-term memory networks for relation classification”, in Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, pp73–78.

[7] S. Hochreiter and J. Schmidhuber, (1997) “Long short-term memory”, Neural Comput., Vol. 9, No. 8, pp1735–1780.

[8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, (2015) “Gated feedback recurrent neural networks”, in Proceedings of the 32nd International Conference on Machine Learning, Vol. 3, pp2067–2075.

[9] P. Zhou et al., (2016) “Attention-based bidirectional long short-term memory networks for relation classification”, in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp207–212.

[10] A. Vaswani et al., (2017) “Attention is all you need”, in Proceedings of the 31st International Conference on Neural Information Processing Systems, pp6000–6010.

[11] Y. Y. Huang and W. Y. Wang, (2017) “Deep residual learning for weakly-supervised relation extraction”, in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp1803–1807.

[12] M. Xiao and C. Liu, (2016) “Semantic relation classification via hierarchical recurrent neural network with attention”, in Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics, pp1254–1263.

[13] J. Lee, S. Seo, and Y. S. Choi, (2019) “Semantic relation classification via bidirectional LSTM networks with entity-aware attention using latent entity typing”, Symmetry (Basel)., Vol. 11, No. 6.

[14] G. E. Hinton, A. Krizhevsky, and S. D. Wang, (2011) “Transforming auto-encoders”, in Proceedings of the ICANN, Vol. 6791, pp44–51.

[15] S. Sabour, N. Frosst, and G. E. Hinton, (2017) “Dynamic routing between capsules”, in Proceedings of the International Conference on Neural Information Processing Systems, pp3859–3869.

[16] N. Zhang, S. Deng, Z. Sun, X. Chen, W. Zhang, and H. Chen, (2018) “Attention-based capsule networks with dynamic routing for relation extraction”, in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp986–992.

[17] X. Zhang, P. Li, W. Jia, and H. Zhao, (2018) “Multi-labeled relation extraction with attentive capsule network”, in Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, pp7484–7491.

[18] M. R. Gormley and M. Dredze, (2014) “Factor-based compositional embedding models”, in NIPS Workshop on Learning Semantics.

[19] Y. Liu, F. Wei, S. Li, H. Ji, M. Zhou, and H. Wang, (2015) “A dependency-based neural network for relation classification”, in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Vol. 2, pp285–290.

[20] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, (2015) “Classifying relations via long short term memory networks along shortest dependency paths”, in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp1785–1794.

[21] Y. Xu et al., (2016) “Improved relation classification by deep recurrent neural networks with data augmentation”, in Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics, pp1461–1470.

[22] D. Bahdanau, K. Cho, and Y. Bengio, (2015) “Neural machine translation by jointly learning to align and translate”, in Proceedings of the 3rd International Conference on Learning Representations.

[23] Y. Shen and X. Huang, (2016) “Attention-based convolutional neural network for semantic relation extraction”, in Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics, pp2526–2536.

[24] G. Hinton, S. Sabour, and N. Frosst, (2018) “Matrix capsules with EM routing”, in Proceedings of the 6th International Conference on Learning Representations.

[25] L. Xiao, H. Zhang, W. Chen, Y. Wang, and Y. Jin, (2018) “MCApsNet: Capsule network for text with multi-task learning”, in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp4565–4574.

[26] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, and Z. Zhao, (2018) “Investigating capsule networks with dynamic routing for text classification”, in Proceedings of the 2018 Conference on Empirical Methods in Natural Language, pp3110–3119.

[27] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, (2018) “Sentiment analysis by capsules”, in Proceedings of the 2018 World Wide Web Conference on World Wide Web, pp1165–1174.

[28] T. Mikolov, K. Chen, G. Corrado, and J. Dean, (2013) “Efficient estimation of word representations in vector space”, in Proceedings of the 1st International Conference on Learning Representations.

[29] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, (2014) "Relation classification via convolutional deep neural network", in Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers, pp2335–2344.

[30] X. Glorot and Y. Bengio, (2010) "Understanding the difficulty of training deep feedforward neural networks", in Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, Vol. 9, pp249–256.

[31] J. Elman, (1990) "Finding structure in time", Cogn. Sci., Vol. 14, No. 2, pp179–211.

[32] Y. Bengio, P. Simard, and P. Frasconi, (1994) "Learning long-term dependencies with gradient descent is difficult", IEEE Trans. Neural Networks, Vol. 5, No. 2, pp157–166.

[33] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, (2013) "Improving neural networks by preventing co-adaptation of feature detectors", Computer Science, Vol. 3, No. 4, pp212-223.

[34] B. Rink and S. Harabagiu, (2010) "UTD: Classifying semantic relations by combining lexical and semantic resources", in Proceedings of the 5th International Workshop on Semantic Evaluation, pp256–259.

AUTHORS

Hongjun Heng received the Ph.D. degree from Nankai University under the supervision of Prof. Z. Wang. He is currently an Associate Professor with the Department of Computer Science and Technology, Civil Aviation University of China. His research interests include intelligent information processing and computer application, specifically include natural language processing and knowledge graph.

Renjie Li received the B.S. degree from the Sino-European Institute of Aviation Engineering, Civil Aviation University of China, China, in 2018, where he is currently pursuing the M.S. degree. His current interests include knowledge graph and nature language processing.