Designing an Adaptive Attention Mechanism for Relation Classification

Pengda Qin, Weiran Xu, Jun Guo
School of Information and Communication Engineering
Beijing University of Posts and Telecommunications
Beijing, China
{qinpengda, xuweiran, guojun}@bupt.edu.cn

***Abstract*—Entity pair provide essential information for identifying relation type. Aiming at this characteristic, Position Feature is widely used in current relation classification systems to highlight the words close to them. However, semantic knowledge involved in entity pair has not been fully utilized. To overcome this issue, we propose an Entity-pair-based Attention Mechanism, which is specially designed for relation classification. Recently, attention mechanism significantly promotes the development of deep learning in NLP. Inspired by this, for specific instance(entity pair, sentence), the corresponding entity pair information is incorporated as prior knowledge to adaptively compute attention weights for generating sentence representation. Experimental results on SemEval-2010 Task 8 dataset show that our method outperforms most of the state-of-the-art models, without external linguistic features.**

I. Introduction

Relation classification is a crucial ingredient in the field of natural language processing (NLP) [1]. It has wide applications including information retrieval [2], question answering [3]–[5] and knowledge base completion [6], [7]. Given a specific entity pair, relation classification system devotes to identify the semantic relation between these two entities. For instance, according to the sentence below,

"The $[owl]_{e1}$ held the mouse in its $[claw]_{e2}$.",
entity owl and $claw$ have the relation of **Component-Whole(e2,e1)**.

Conventional methods is over-reliant on expensive linguistic features made by experts and well-established NLP tools, which suffer from error propagation and limitation of novel domains. To alleviate this issue, deep learning techniques have attracted considerable attentions. Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and their variations are the current most commonly used frameworks for tackling relation classification [8], [9]. Compared with CNN, RNN is better at remembering long-distance context information, especially Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). Under the same performance, computational cost of GRU is less than LSTM.

Recently, attention mechanism gets a wide range of applications. For NLP, it is capable of automatically concentrating on valuable words for targeted task. The concept of attention is initially put forward for sequence-to-sequence learning. In the case of machine translation, attention weights are calculated based on the hidden state of the predicted words. Various predicted words provide various $priori\ knowledge$ to attention layer for generating corresponding attention weights. However, relation classification is a sequence-to-label problem, which means this strategy is not suitable. Prevailing solution is to randomly initialize an $unique$ vector to calculate attention weights and fine-tune this vector via end-to-end system training process. Obviously, this unique vector can not provide rational priori knowledge and is suboptimal for the heterogeneous structure of relation expressions. Moreover, because this vector is completely trained from training set, over-fitting is another issue.

Entity pair is the essential component for relation classification. The semantic knowledge and appearance order of them reflect crucial information. More importantly, it varies from instance to instance. As instances shown below, the expression patterns between different entity pairs are both "$A\ of\ B$". Hence, without the information of entity pairs, it is difficult to recognize their relation types.

- I was attacked by a $[flock]_{e1}$ of $[pigeons]_{e2}$ today. (**Member-Collection(e2,e1)**)
- He decided to pad the $[heel]_{e1}$ of $[shoes]_{e2}$ with a shock absorbing insole or heel pad. (**Component-Whole(e1,e2)**)

Consequently, better utilization of entity pair information is bound to bring positive effect to system performance.

This paper designs an Entity-pair-based Attention Mechanism specially for tackling relation classification task, which sufficiently takes advantage of entity pair information. Based on this, we construct an Entity-pair-based Attention Bidirectional GRU (EAtt-BiGRU) model. Bidirectional GRU is utilized to capture valuable word-level information. This structure can effectively alleviate the biased problem of unidirectional version (later inputs are more dominant than earlier inputs). More importantly, for different input instances (entity pair, sentence), we incorporate the corresponding entity pair information as priori knowledge into attention layer, which means the generated attention weights are adaptive to corresponding instances. Through visualization, the attention weight distribution is more reasonable than previous attention mechanism. We achieve a F1-score of 84.7% on SemEval-2010 benchmark, which is higher than most of the existing methods, without external linguistic features.

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

II. Related Works

Apart from supervised version, distant supervised methods are also commonly used for relation classification [10], [11]. However, due to the characteristic of neural network, most works of relation classification are in supervision. Traditional approaches have excessive dependence on features extracted from linguistic analysis modules [12]. Since the advent of representative CNN architecture [8], a number of elaborate variants have been released [9]. CNN performs better on recognize consecutive patterns for relation mentions; however, in some cases, relation patterns consist of inconsecutive text segments. In consideration of this issue, we build our method on RNN, which is suitable for mining long-distance discrete information [13].

SDP-LSTM [14] employs LSTM as neural architecture and utilizes the shortest dependency path (SDP) between entity pair as input. SDP is beneficial to eliminating irrelevant words in the sentence. More than that, SDP-LSTM combines extra linguistic knowledge to make another step forward in performance, such as POS, grammatical relations and WordNet hypernyms. However, the proposed method simply leverages raw word sequence as input, without external linguistic features.

In order to alleviate the biased problem of RNN, some related works combine forward RNN with backward RNN to jointly extract word-level information [13]. BLSTM [15] adopts a bidirectional LSTM to identify relation and confirms the superiority against unidirectional framework. In proposed method, we follow this strategy; however, in consideration of computational cost, LSTM is substituted by GRU.

Attention mechanism is originally proposed for machine translation [16] and have attracted a lot of interests currently. Subsequently, this concept is further used in parsing [17] and question answering [18]. Unlike these sequence-to-sequence systems, relation classification is a sequence-to-label problem. Aiming at this issue, Hierarchical Attention Networks (HN-ATT) puts forward a solution [19]. This model randomly initializes an unique vector, and the attention weights are calculated by dot product operation between this vector and corresponding word-level feature vectors. Inspired by this, Attention-based Bidirectional LSTM (Att-BLSTM) employs a similar attention strategy to capture the most important semantic information in sentence [20]. However, comparing to this, our model incorporates entity pair information as priori knowledge into attention layer, which is conducive to obtain more adaptive attention weights.

III. Methodology

Given an input sentence S with a pair of annotated entity mentions e_1 and e_2, relation classification system is to calculate the probability distribution of each potential relation class and assign the maximum one for input. Figure 1 depicts the overview of EAtt-BiGRU network. Overall, the proposed model mainly consists of four components:

- Input embedding layer: Every token in input sentence is parametrized into real-valued embedding, involving word and position information.
- Bi-GRU layer: Bi-GRU layer transforms each input embedding into high-level feature representation.
- Entity-pair-based Attention layer: Entity pair, the particular ingredient of relation classification, is utilized to produce attention weight vector. Through being weighted by this vector, outputs of Bi-GRU are integrated into sentence-level vector.
- Output layer: Relation type is identified relying on the obtained sentence-level vector.

In below subsections, these four components are described in detail.

A. Input Representation

With regard to relation classification, input is composed of word sequence $S = \{w_1, w_2, ..., w_n\}$ and entity pair $[w_{e1}, w_{e2}]$. Correspondingly, input representation consists of word embedding and position embedding.

1) Word embedding: Word embedding [21] performs well in reflecting word-level information. Given a word embedding matrix $W^{emb} \in \mathcal{R}^{d_e \times |V|}$, i-th word in sentence S can be encoded as w_i by looking up this matrix. $|V|$ is the fixed-size vocabulary size and d_e denotes the dimension of word embedding.

2) Position embedding: Observation revealed that crucial information for identifying relation tends to be concentrated on the words close to targeted entity pair. Therefore, position representation is utilized to highlight this part. Position feature [8] transforms the relative distance from context to entity pair into distributed representation. For word w_i, corresponding to head entity w_{e1} and tail entity w_{e2}, there are two distance vectors p_i^1 and p_i^2. p_i^1 and p_i^2 have the same dimension of d_p and are modified via system training process. Position feature of word that occurs on the left of entity is encoded as the vector of negative number, otherwise positive number. The overall input representation for i-th word is $w_i^* = [(w_i)^\top, (p_i^1)^\top, (p_i^2)^\top]^\top$

B. Bidirectional GRU Sequence Encoder

RNN provides good performance in sequence learning. To alleviate the dilemma of vanishing and exploding, LSTM [22] adopts an adaptive gating mechanism, which leverages a memory cell to remember related information and forget irrelevant content. GRU [23] is a simplified version of LSTM. It inherits the merits of gating mechanism and greatly reduces the number of parameters. So we select GRU to model sequence data for our relation classification system.

The hidden layer operations of GRU are primarily controlled by two gates: the reset gate r_t and the update gate z_t. At t step, the new hidden state h_t is calculated as

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

Simply put, new hidden state h_t is generated by a linear interpolation between the previous hidden state h_{t-1} and new hidden state $\tilde{h}_t$. The update gate z_t simultaneously defines

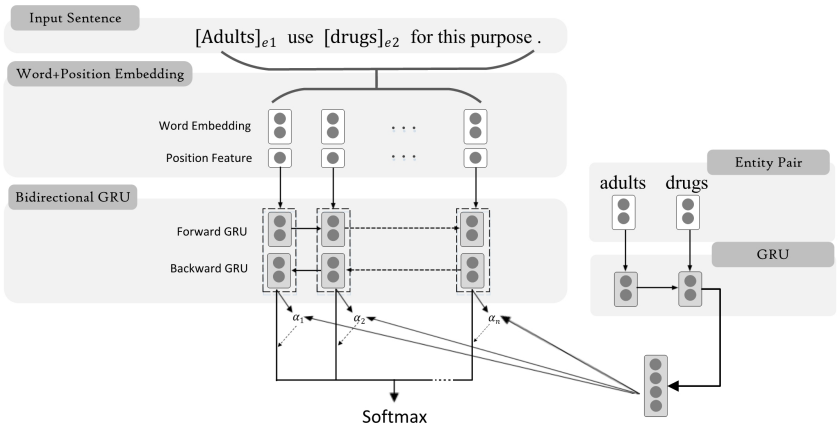


Fig. 1. The framework of EAtt-BiGRU. Taking an annotated sentence from SemEval-2010 Task 8 dataset as example, left part include input embedding layer, Bi-GRU layer and output layer; right part represents the Entity-pair-based Attention layer.

how much previous memory is retained and how much new information is added. z_t is computed by

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

where x_t denotes the input in time t, σ is the logistic sigmoid function.

The computation of new hidden state $\tilde{h_t}$ is analogous to traditional RNN:

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

The different is that, an reset gate r_t is used to determine how to combine the new input with the previous memory. When r_t is close to zero, it means the previous hidden state is compelled to be ignore. r_t is updated as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r). \tag{4}$$

Standard GRU is the unidirectional network, which means current hidden states only have access to the past context in temporal order. However, relation classification depends on the overall information of sentence. Future words equally have impact on the semantic of past content. To solve this problem, we adopt a bidirectional architecture as in Figure 1 to summarize information for words from both directions. Standard GRU is regarded as the forward GRU. Backward GRU is used to process sequences in opposite temporal order. In time t, $\overrightarrow{h_i} \in \mathcal{R}^{d_h}$ and $\overleftarrow{h_i} \in \mathcal{R}^{d_h}$ are the hidden states of forward and backward versions respectively. Based on this, the final hidden state h_i^* of i-th word is represented as the concatenation of $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$,

$$h_t^* = [\overrightarrow{h_i}, \overleftarrow{h_i}]. \tag{5}$$

which involves the entire information centered around w_i within sentence.

C. Entity-pair-based Attention Mechanism

Recently, attention mechanism is regarded as a successful technology for deep learning. When people read an article, they pay more attention to the words or segments that are valuable for the comprehension of text. Inspired by this, when computer processes NLP tasks, not all words contribute equally to the ultimate objective. Therefore, attention mechanism is introduced to let neural network automatically calculate a series of weight distribution for words in text sequence. It is initially proposed and applied in question answering, machine translations, speech recognition [24] and image captioning [25]. Obviously, the commonality of these application fields is that they belong to sequence-to-sequence problem. For different instances, there are different priori knowledge injected into attention layer. However, current attention mechanism for relation classification task cannot incorporate analogous adaptive priori knowledge into prediction.

Entity pair is the crucial ingredient for identifying relation. For various entity pairs in the same sentence, there exist various relation labels. Naturally, for different relation of interest, the contribution of the same word may be unequal. For this reason, we introduce a adaptive attention mechanism for relation classification based on entity pair information.

First, entity pair e_1 and e_2 are encoded into the corresponding word embedding w_{e1} and w_{e2}. In the case of multi-word entity, for instance, e_1 consists of 3 words; w_{e1} represents the word sequence $w_{e_1} = \{w_{e_1^1}, w_{e_1^2}, w_{e_1^3}\}$. The overall input of attention layer is the sequence $S_e = \{w_{e_1}, w_{e_2}\}$. Then, we employ an unidirectional GRU to extract features from entity pair. Because every actual relation has directions, such as *Component-Whole(e1,e2)* and *Component-Whole(e2,e1)*, the appearance order of entity pair plays an important role in the identification between this kind of relation pair. Therefore, we utilize GRU to highlight temporal order information.

Unlike the aforementioned BiGRU Encoder, for this part,

we only use the last hidden state as output.

$$u_a = GRU(S_e). \tag{6}$$

Let $H = [h_1^*, h_2^*, ..., h_n^*]$ be a matrix consisting of outputs of BiGRU. Subsequently, the contribution of w_i is calculated by dot product between entity-pair-level vector u_a and word-level vector u_i. Through a softmax operation, we get a normalized attention weight α_i.

$$u_i = tanh(h_i^*). \tag{7}$$

$$\alpha_i = \frac{exp(u_i^\top u_a)}{\sum_i exp(u_i^\top u_a)} \tag{8}$$

Then, we adopt two different schemes to calculate sentence representation.

1) Vector sum: BiGRU outputs are aggregated by vector sum operation weighted on attention weight α_i, and we get the final sentence representation s.

$$s = \sum_i \alpha_i h_i^*. \tag{9}$$

2) Max-pooling: The other scenario to produce sentence representation is max-pooling operation. Although widely used in CNN, it is also applicable to integrate the outputs of RNN [13], [15]. It is formulated as follows:

$$s_j = \max_i [\alpha_i h_{ij}^*], \quad \forall j = 1, ..., 2d_h \tag{10}$$

where s_j is the j-th dimension of s.

D. Classification

Through elaborated layers above, the obtained sentence vector is a high-level sentence representation, which can be directly used for predict label $\hat{y}$. This classification process consists of a softmax classifier:

$$p(y|S) = softmax(W_c s + b_c) \tag{11}$$

The relation label with highest probability value is identified as ultimate result,

$$\hat{y} = \underset{y}{argmax}\ p(y|S) \tag{12}$$

E. Implementation Details

Dropout is a stochastic regularization technique [26]. Input embedding layer is exerted by the dropout rate ρ_w. In addition, we adopt dropout rate ρ_a on entity-pair-based attention layer and ρ_c on the output layer. Max-norm is also used to prevent the blow-up of hidden parameters. After a gradient descent step, the neural network is optimized under the constraint $||W||_2 \leq \varepsilon$, where ε is a tunable hyperparameter that decided by validation set.

IV. Datasets and Experimental Setup

The proposed method is evaluated on the commonly used SemEval-2010 Task 8 benchmark[1] [27]. This dataset has 10717 sentences that consist of the predefined training set of 8000 examples and test set of 2717 examples. Each sentence has been annotated with two target entities and an unique relation label. There are 9 actual relation classes, together with an artificial class *Other*. Particularly, as mentioned above, each actual relation class has a reversed version, such as *Content-Container(e1,e2)* and *Content-Container(e2,e1)*. That is to say, for a specific actual relation, the sentence with entity pair in opposite order belongs to different relation class. However, *Other* does not have a reverse version. So, the total number of relation classes is 19. The official evaluation metric is adopted to evaluate our models. The final performance is based on macro-averaged F1-score over 9 actual relation classes (excluding *Other*) that takes directionality into account.

We use the released word embedding set *GoogleNews-vectors-negative300.bin* to initialize our embedding layer, which is trained by Mikolov's word2vec tool[2]. Beyond that, other parameter matrices in our model are initialized randomly following a Gaussian distribution. We utilize AdaDelta [28] with a mini-batch size to learn network parameters. Within training set, we randomly select 800 examples as validation set. The optimal hyperparameters are determined by a cross-validation procedure. Detailed settings are presented in Table I.

TABLE I
Hyperparameter settings

Hyperparameter	Value
d_e, d_p	300,10
d_h	100
ρ_w, ρ_a, ρ_c	$0.7, 0.2, 0.5$
ε	3
$Learning\ rate$	1.0
$Batch\ size$	20

V. Results and Discussion

Table II illustrates the comparison between EAtt-BiGRU and other state-of-the-art relation classification models. For SemEval-2010 Task 8 benchmark, the SVM [12] presented in the first entry is a top performing traditional feature-based method. This model combines with a rich set of costly handcrafted features to generate sentence-level feature, and achieves an F1-score of 82.2%.

Subsequently, more progresses are made by neural networks. MVRNN [29] pioneers the use of end-to-end neural network for relation classification, which constructs a recursive neural network based on the syntactic parsing tree and simultaneously trains additional matrices to modify the meanings of neighboring words. It raises the F1 to 82.4%. CNN [8] leverages the raw word sequence as input and exploits Position Feature to identify the position of entity pair. The extra lexical

[1] https://docs.google.com/document/d/1QO_CnmvNRnYwNWu1-QCAeR5ToQYkXUqFeAJbdEhsq7w/preview

[2] http://code.google.com/p/word2vec/

TABLE II
COMPARISON WITH PREVIOUS RELATION CLASSIFICATION SYSTEMS

Model	Additional Information	F1
SVM (Rink and Harabagiu 2010)	POS, prefixes, morphological, WordNet, dependency parse, Levin classed, ProBank, FrameNet, NomLex-Plus, Google n-gram, paraphrases, TextRunner	82.2
MVRNN (Socher et al. 2012)	word embedding, syntactic parsing tree	79.1
	+POS, NER, WordNet	82.4
CNN (Zeng et al. 2014)	Word embeddings+PF[1]	78.9
	+WordNet, words around nominal	82.7
BRNN (Zhang and Wang 2015)	Word embeddings	82.5
CR-CNN (dos Santos et al. 2015)	Word embeddings	82.8
	+ PF	84.1
SDP-LSTM (Xu et al. 2015)	Word embeddings	82.4
	+POS+GR+WordNet embeddings	83.7
BLSTM (Zhang et al. 2015)	Word embeddings	82.7
	+PF+POS+NER+WNSYN+DEP	84.3
Att-BLSTM (Zhou et al. 2016)	Word embeddings+PI[2]	84.0
Att-BLSTM[3]	Word embeddings+PF	83.5
EAtt-BiGRU	Word embeddings+PF	**84.7**

[1] Position Feature.
[2] Position Indicator [13], which is to insert four position indicators into input sentence to specify the starting and ending of target entity pair.
[3] The experimental result of this entry is obtained through reproducing Att-BLSTM method; however, PI is substituted by PF.

features are transformed into distributed representations and concatenated with sentence-level vector to predict relation class. F1-score is elevated to 82.7%. SDP-LSTM [14] treats the shortest dependency path between entity pair as input to pick up heterogeneous information. The external linguistic features are integrated via multichannel LSTM networks. Based on this, it achieves an F1-score of 83.7%. For the purpose of better performance, works mentioned above involve the external lexical knowledge. However, our proposed model, without complicated human-designed features, still achieves the superior F1-score of 84.7%.

CR-CNN [9] focus more on the influence of class *Other*, which proposes a new pairwise ranking function to substitute softmax. This targeted modification obtains F1-score of 84.1%. Our model simply uses basic softmax operation to finish classification.

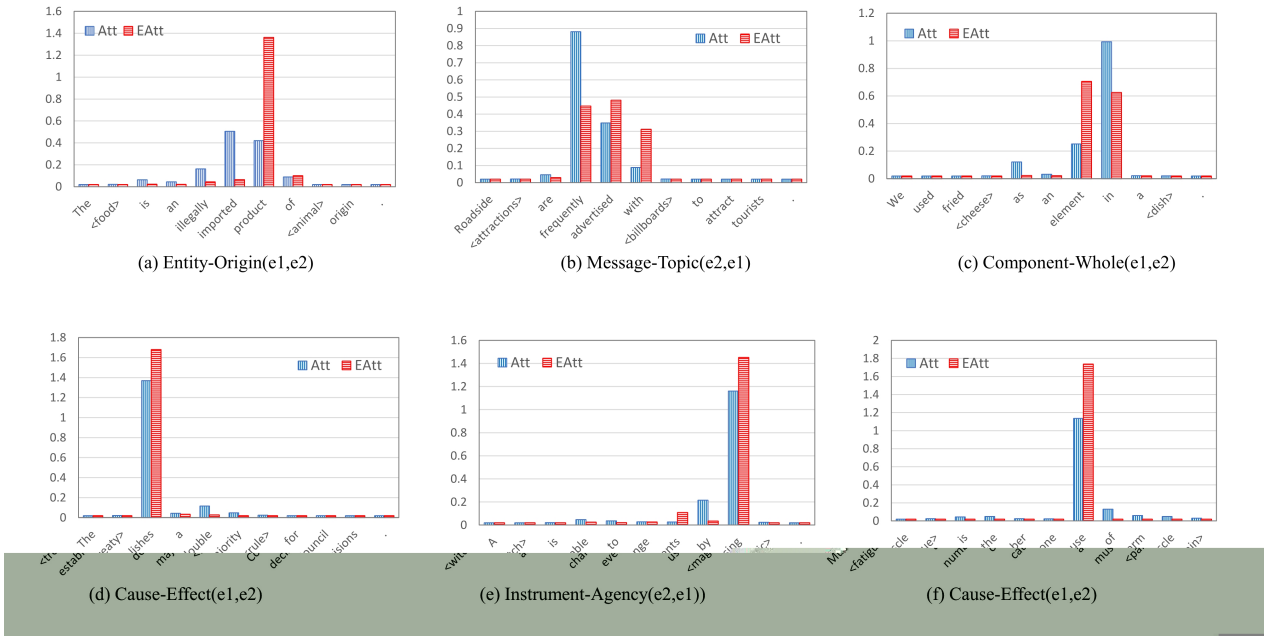
BRNN, BLSTM and Att-BLSTM are three more relevant works to our model. Similar to our model, they all utilize the Bidirectional RNN architecture. BRNN [13] leverages the original RNN and max-pooling operation to extract sentence-level feature, which achieves F1-score of 82.5% with Position Indicator. BLSTM [15] adopts bidirectional LSTM to modeling sequence, and employs a piece-wise max pooling to generate sentence level representation. With the assistance of NLP tools and lexical resources, this model achieves the performance of 84.3%. However, our result is yielded without any extra features. Att-BLSTM [20] equally adopts attention mechanism to tackle relation classification. The difference is that, we give attention layer reasonable priori knowledge. The generation of attention weights depends on corresponding entity pair information. Such specific design brings us better performance than Att-BLSTM.

A. Detailed Analysis

In order to further analyze the effectiveness of proposed model, we give some detailed comparison results in Table III. For fair comparison, hyperparameters are set based on validation dataset. Firstly and obviously, in contrast to models without attention layer, attention mechanism indeed enhances the performance of relation classification. Particularly, compared with previous attention strategy, our entity-pair-based attention mechanism gives a top performance. From this point of view, it can be concluded that our attention strategy is more suitable for relation classification task.

In above descriptions, we introduce two different solutions, vector sum and max-pooling, to generate sentence-level representation. Table III demonstrates that, without attention layer, two solutions achieve similar results. However, with the integration of attention mechanism, max-pooling just obtains suboptimal result. Max-pooling operation is to select the maximum value in specific feature dimension, which can be interpreted as a special attention mechanism. It gives weight "1" for the position of maximum value, and the rest are assigned "0". On this base, if we exert an attention mechanism again, the attention weights may impair the extracted information. Above all, the best performance is yielded by combining entity-pair-based attention mechanism with vector sum operation.

Figure 2 presents a more intuitive analysis of entity-pair-based attention mechanism. Generally, two types of attention mechanism certainly work reasonably that assign larger weights for words with crucial information. However, by comparison, entity-pair-based attention performs more rationally. For instance, histogram(a) shows the weight distribution over



(a) Entity-Origin(e1,e2) (b) Message-Topic(e2,e1) (c) Component-Whole(e1,e2)

(d) Cause-Effect(e1,e2) (e) Instrument-Agency(e2,e1)) (f) Cause-Effect(e1,e2)

Fig. 2. Attention visualization comparison between original attention mechanism and entity-pair-based attention mechanism. Histograms (a)(b)(c) are in the cases that two attention strategies give the different highest weighted words. Histograms (d)(e)(f) show the cases that the highest weighted words are the same, but weight degrees are different. In order to give a better visual feeling, the value of the y-coordinate is computed as $exp(a_i) - 0.98$. In horizontal coordinate, words in angle brackets represent target entities.

TABLE III
COMPARISON BETWEEN THE MAIN MODEL AND VARIANTS

Model	F1
BiGRU+MaxPooling	82.8
BiGRU+Sum	82.7
BiGRU+Att+MaxPooling	83.2
BiGRU+Att+Sum	83.6
BiGRU+EAtt+MaxPooling	83.5
BiGRU+EAtt+Sum	**84.7**

an annotated sentence of *Entity-Origin(e1,e2)*. According to the human's understanding, word "product" plays a more important role in identifying corresponding relation type. As expected, entity-pair-based attention assigns a conspicuous weight for "product". In contrast, original attention strategy irrationally gives word "imported" the highest weight. Moreover, with a view to the cases in histogram (d)-(f), although the key words are correctly recognized by both methods, entity-pair-based attention mechanism calculates higher weight values, which indicates that our model possesses a greater identification capability for relation classification.

VI. CONCLUSION

In this paper, we propose an Entity-pair-based attention mechanism specially for solving relation classification, which utilizes entity pair information as priori knowledge to adaptively generate attention weights. Based on this mechanism, we construct a novel neural network, named EAtt-BiGRU. The experimental results on SemEval-2010 relation classification task demonstrate that, without extra linguistic features, our model still achieves state-of-the-art performance. Moreover, the visualization result shows that entity-pair-attention mechanism evidently yields more robust attention weight distribution.

ACKNOWLEDGMENT

This work was supported by 111 Project of China under Grant no. B08004, the National Natural Science Foundation of China (61273217, 61300080), the Ph.D. Programs Foundation of Ministry of Education of China (20130005110004).

REFERENCES

[1] Y. Ke and M. Hagiwara, "A natural language processing neural network comprehending english," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–7.
[2] Z. Luo, M. Osborne, S. Petrovic, and T. Wang, "Improving twitter retrieval by exploiting structural information." in *AAAI*, 2012.
[3] H. Fang, F. Wu, Z. Zhao, X. Duan, Y. Zhuang, and M. Ester, "Community-based question answering via heterogeneous social network learning," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
[4] Y. Zhang, S. He, K. Liu, and J. Zhao, "A joint model for question answering over multiple knowledge bases," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
[5] S. Xiang, W. Rong, Y. Shen, Y. Ouyang, and Z. Xiong, "Multidimensional scaling based knowledge provision for new questions in community question answering systems," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 115–122.
[6] J. Urbani, C. Jacobs, and M. Krötzsch, "Column-oriented datalog materialization for large knowledge graphs," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
[7] M. Bienvenu, C. Bourgaux, and F. Goasdoué, "Explaining inconsistency-tolerant query answering over description logic knowledge bases," in *AAAI Conference on Artificial Intelligence*, 2016.

[8] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao *et al.*, "Relation classification via convolutional deep neural network." in *COLING*, 2014, pp. 2335–2344.

[9] C. N. dos Santos, B. Xiang, and B. Zhou, "Classifying relations by ranking with convolutional neural networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 1, 2015, pp. 626–634.

[10] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 1003–1011.

[11] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal*, 2015, pp. 17–21.

[12] B. Rink and S. Harabagiu, "Utd: Classifying semantic relations by combining lexical and semantic resources," in *Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2010, pp. 256–259.

[13] D. Zhang and D. Wang, "Relation classification via recurrent neural network," *arXiv preprint arXiv:1508.01006*, 2015.

[14] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, "Classifying relations via long short term memory networks along shortest dependency paths," in *Proceedings of Conference on Empirical Methods in Natural Language Processing (to appear)*, 2015.

[15] S. Zhang, D. Zheng, X. Hu, and M. Yang, "Bidirectional long short-term memory networks for relation classification," 2015.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Computer Science*, 2014.

[17] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language," *Eprint Arxiv*, 2014.

[18] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," *Computer Science*, 2015.

[19] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

[20] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *The 54th Annual Meeting of the Association for Computational Linguistics*, 2016, p. 207.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[22] D. Gusfield, *Algorithms on Strings, Trees and Sequences*. Cambridge, UK: Cambridge University Press, 1997.

[23] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[24] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems*, 2015, pp. 577–585.

[25] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *arXiv preprint arXiv:1502.03044*, vol. 2, no. 3, p. 5, 2015.

[26] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *Computer Science*, vol. 3, no. 4, pp. pgs. 212–223, 2012.

[27] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz, "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," in *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*. Association for Computational Linguistics, 2009, pp. 94–99.

[28] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[29] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 1201–1211.