SCIENCE CHINA
Information Sciences

• LETTER •

November 2019, Vol. 62 219102:1–219102:3
https://doi.org/10.1007/s11432-018-9528-8

Implicit discourse relation detection using concatenated word embeddings and a gated relevance network

Jinlan FU, Qi ZHANG*, Jifan CHEN, Minlong PENG, Tao GUI, Xipeng QIU & Xuanjing HUANG

Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai 201203*, China*

Received 3 March 2018/Revised 14 June 2018/Accepted 20 July 2018/Published online 18 September 2019

Citation Fu J L, Zhang Q, Chen J F, et al. Implicit discourse relation detection using concatenated word embeddings and a gated relevance network. Sci China Inf Sci, 2019, 62(11): 219102, https://doi.org/10.1007/s11432-018-9528-8

Dear editor,

Discourse relation detection involves recognizing the relationships between pairs of discourse fragments (e.g., clauses or sentences). As compared with explicit detection, implicit discourse relation detection is much more challenging, owing to connective words, such as "so" or "because", being absent. In such cases, the relationships between the fragments pairs cannot be via simple frequency-based mapping; instead, the relationships must be inferred from potential semantic and logical connections between the two arguments.

Many methods have been proposed to solve the implicit discourse relation detection task [1–5]. Some of these studies require heavy data preprocessing and specific hand-crafted features, such as linguistically informed [1] or part-of-speech (POS) [2]. Liu et al. [3] proposed a multi-task learning system that combines similar tasks by learning both shared and unique representations. Zhang et al. [4] proposed a generative model that generates both the discourse and the relationship between the two arguments. Qin et al. [5] used context-aware character-enhanced embeddings as input to a convolutional neural network (CNN); however, they captured semantic interactions at sentence level. In addition, most previous studies [1–4] have regarded words as the smallest units for feature extraction and ignored character level features, including morphological ones, even though, character-level features can handle the rare word problem.

In this study, similar to majority of the previous study related to character-level features, we utilize a stacked CNN to capture the character-level features, then combine these features with ordinary word embeddings, concatenated word-embeddings. Unlike Qin et al. [5], we capture the semantic interactions between arguments using word pairs. Specifically, we propose to use a gated relevance network (GRN) that combines a bilinear model and single layer network with a gate to compute the arguments' relevance scores from the word pairs.

Character-level embeddings. In our proposed approach, character-level embeddings are captured by a stacked CNN. The aim of using character-level embeddings is to handle the rare word problem, as these embeddings can capture rich morphological information, such as prefixes, suffixes, genders, and tenses.

Consider the word $k \in V$, where k comprises the character sequence $C = [c_1, c_2, \ldots, c_n]$, V is a fixed-size word, and n is the word length. In addition, let $c_i \in \mathbb{R}^{d_c}$ be the character vector for the i-th character in the word, where d_c is the dimensionality of the character vector. we apply a convolution operation between the character matrix C

* Corresponding author (email: qz@fudan.edu.cn)

© Science China Press and Springer-Verlag GmbH Germany, part of Springer Nature 2019

and a filter $H \in \mathbb{R}^{d_c * w}$ of width w. Following the convolution we use the nonlinear function relu to capture the character feature map $f^k \in \mathbb{R}^{n-w+1}$. The i-th element of f^k is as follows:

$$f^k[i] = [\ldots; \text{relu}(H_i \times C^k[i : i+w-1] + b); \ldots]. \quad (1)$$

Next, we apply max-pooling to retain only the highest value y^k of the given filter:

$$y^k = \max(f^k[i]). \quad (2)$$

In this study, the character embeddings were pre-trained on the Penn Treebank (PTB) and Penn Discourse Treebank (PDTB) using the method proposed by Kim [6]. The alphabet used in this study contains the following 69 characters:

abcdefghijklmnopqrstuvwxyz0123456789
-,;.!?:'"/ _@#$%&*+−= _⟨⟩()[]{}| ∼ ∧ \

The character-level word embeddings produced by the CNN are independent of each other and lack any contextual information; hence, a single-layer bidirectional long short-term memory (LSTM) [7] is used to capture the contextual information. Assume that the contextual representation from left to right is h_t^f, and from right to left is h_t^b. Then we concatenate them to get rich contextual representation $h_t = [h_t^f, h_t^b]$.

Concatenated word embedding using a bidirectional LSTM. To obtain both contextual and morphological information, we concatenate the ordinary word embedding with the above character-based word embedding, then utilize a bidirectional LSTM again. Specifically, we feed the concatenated word embeddings into a bidirectional LSTM.

Gated relevance network. In order to get the semantic interaction information between the two arguments, we use a GRN, which combines a bilinear model [8] and single layer network using a gate mechanism. The GRN computes interaction values for word pairs from different arguments.

Let x_{h_i} and y_{h_j} be the representations of words from the two arguments, X and Y, respectively.

Bilinear model can capture the linear interactions between these two vectors, but cannot handle nonlinear interactions. The bilinear model is defined as follows:

$$s(x_{h_i}, y_{h_j}) = x_{h_i}^{\mathrm{T}} M y_{h_j}, \quad (3)$$

where $M \in \mathbb{R}^{d_h \times d_h}$ is a parameter matrix and d_h is the dimension of word representation after bidirectional LSTM.

The single layer network could capture nonlinear interaction, but the interaction between the vectors is weak. The single layer network is defined as follows:

$$s(x_{h_i}, y_{h_j}) = u^{\mathrm{T}} f\left(V \begin{bmatrix} x_{h_i} \\ y_{h_j} \end{bmatrix} + b\right), \quad (4)$$

where $V \in \mathbb{R}^{k \times 2d_h}$, $b \in \mathbb{R}^k$, and $u \in \mathbb{R}^k$, and f is a nonlinear function that is applied element-wise.

Clearly, both models have their own advantages, and combining can enable us to inherit the benefits of both. In particular, we utilize a gate mechanism to incorporate both models, thereby enabling our model to capture semantic interactions more robustly by adaptively taking a linear or nonlinear approach as appropriate. The gate g is defined as

$$g = \sigma\left(W_g \begin{bmatrix} x_{h_i} \\ y_{h_j} \end{bmatrix} + b_g\right), \quad (5)$$

where σ is the sigmoid function, and $W_g \in \mathbb{R}^{r \times 2d_h}$ and $b \in \mathbb{R}^r$ are parameter and bias, respectively.

The GRN can now be defined as

$$s(x_{h_i}, y_{h_j}) = u^{\mathrm{T}}\Bigg(g \odot x_{h_i}^{\mathrm{T}} M^{[1:r]} y_{h_j} + (1-g) \odot f\left(V \begin{bmatrix} x_{h_i} \\ y_{h_j} \end{bmatrix}\right) + b\Bigg), \quad (6)$$

where g is the gate, f is a nonlinear function. $M^{[1:r]} \in \mathbb{R}^{r \times d_h \times d_h}$ is a bilinear tensor, and the bilinear process $x_{h_i}^{\mathrm{T}} M^{[1:r]} y_{h_j}$ produces $m \in \mathbb{R}^r$, where each slice $l = 1, 2, \ldots, r$ of the tensor is used to compute one entry of the bilinear result m. In addition, $V \in \mathbb{R}^{r \times 2d_h}$, $b \in \mathbb{R}^r$, and $u \in \mathbb{R}^r$.

In summary, the GRN computes semantic interaction scores for each pair of word representations, creating a matrix of semantic interaction scores for the two arguments.

Experiment. Our experiment utilized the PDTB 2.0 dataset, and considered the following relationship categories: comparison (Comp.), contingency (Cont.), expansion (Expa.), and temporal (Temp.). For comparison with previous studies [1–5], we treated these four relationships as four separate classification sub-tasks and accordingly trained four binary classifiers. In our model, we utilized truncation or zero-padding operations to fix the lengths of the segments and words as 50 and 20, respectively. To initialize the word embeddings, we utilized 300-dimensional pre-trained embeddings using word2vec[1)]. The character embedding dimensionality d_c was set to 15. The dimension of bidirectional LSTMs' intermediate rep-

1) http://www.code.google.com/p/word2vec.

resentations were 50-dimensional in the character-level module and 100-dimensional in the concatenated word embedding module. In the character-level module, we used three groups of 32 filters with window sizes of 2, 3, and 4. We used the AdaGrad optimizer, and set the learning rate to 0.05. The number of tensor slices was set to 2.

To provide a clearer illustration of our model's performance, we divided our baseline models into three groups, namely character-level (Char.), word-level (Word.), and concatenated modules (Con.). Table 1 shows the results thus obtained, together with those of previous studies [1–5]. And the Word Bi-LSTM+GRN model was first proposed by Chen et al. [9].

Table 1 Experimental results for the PDTB dataset

	Comp.	Cont.	Expa.	Temp.
Pitler et al. (2009) [1]	21.96	47.13	76.42	16.76
Rutherford and Xue. (2014) [2]	39.70	54.42	80.44	28.69
Liu and Li. (2016) [3]	36.70	54.76	–	31.32
Zhang et al. (2016) [4]	35.88	50.56	–	29.54
Qin et al. (2016) [5]	38.67	54.91	80.66	**32.76**
Char. CNN	32.76	49.53	76.80	22.12
Char. CNN+Bi-LSTM	33.63	50.42	77.99	22.58
Char. CNN+Bi-LSTM+GRN	34.14	51.44	78.31	23.22
Word. LSTM	35.48	52.11	77.36	27.62
Word. Bi-LSTM	37.35	52.27	78.33	29.36
Word. Bi-LSTM+GRN	40.17	54.76	80.62	31.32
Con. LSTM	36.86	52.56	78.28	28.89
Con. Bi-LSTM	38.11	53.38	79.22	31.37
Con. LSTM+GRN	38.32	53.52	78.34	29.02
Con. Bi-LSTM+GRN	**41.02**	**54.94**	**80.78**	31.76

Results and analysis. First, the results indicate that our model outperforms most of the previous models. Because the Temporal (Temp.) only contains 826 examples (the numbers of samples for training, development, and testing are 665, 93, and 68, respectively), it is unstable to use the data-hungry deep learning method for classification. Therefore, the method proposed by Qin et al. [5] outperform our method in Temporal (Temp.) slightly.

Second, on comparing methods using the same feature extraction method (e.g, LSTM, Bi-LSTM, and Bi-LSTM+GRN) with different word embedding techniques (e.g, character-level, word-level, or concatenated), the models based on concatenated word embeddings exhibited the best performance. This was because concatenated word embeddings have the advantages of both character- and word-level embeddings; hence, they can not only handle the rare word problem and include rich morphological information, but also capture more semantic information like word level embeddings do.

Third, on comparing methods that use the same word embedding technique with different feature extraction methods, models that used GRN to compute the semantic interaction scores achieved better performance, because GRNs capture both linear and nonlinear interactions.

In summary, our model exhibited better performance as compared with previous approaches and baseline models indicate that combining a GRN with concatenated word embedding is effective.

Acknowledgements This work was partially funded by National Natural Science Foundation of China (Grant Nos. 61532011, 61473092, 61472088) and Science and Technology Commission Shanghai Municipality (Grant Nos. 16JC1420401, 17JC1420200).

References

1 Pitler E, Louis A, Nenkova A. Automatic sense prediction for implicit discourse relations in text. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language, Suntec, 2009. 683–691

2 Rutherford A, Xue N. Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, 2014. 645–654

3 Liu Y, Li S J, Zhang X D, et al. Implicit discourse relation classification via multi-task neural networks. 2016. ArXiv: 1603.02776

4 Zhang B, Xiong D, Su J, et al. Variational neural discourse relation recognizer. 2016. ArXiv: 1603.03876

5 Qin L, Zhang Z, Zhao H. Implicit discourse relation recognition with context-aware character-enhanced embeddings. In: Proceedings of COLING, Osaka, 2016. 1914–1924

6 Kim Y, Jernite Y, Sontag D, et al. Character-Aware Neural Language Models. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2016. 2741–2749

7 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput, 1997, 9: 1735–1780

8 Sutskever I, Tenenbaum J B, Salakhutdinov R R. Modelling relational data using Bayesian clustered tensor factorization. Adv Neural Inf Process Syst, 2009, 2009: 1821–1828

9 Chen J, Zhang Q, Liu P, et al. Implicit discourse relation detection via a deep architecture with gated relevance network. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016. 1: 1726–1735