Research on Entity Relation Extraction Methods Based on Recurrent Convolutional Neural Networks: Postprint

Authors: Wan Jing, Li Haoming, Yan Huanchun, Zhang Xuechao

Date: 2019-01-03T00:00:00+00:00

Abstract

Addressing the issue that in most current relation extraction approaches, longer entity co-occurrence sentences in text corpora often yield only local features while failing to capture long-distance dependency information, we propose an entity relation extraction model based on recurrent convolutional neural networks and attention mechanisms. The GRU recurrent neural network, which excels at modeling long-distance dependencies, is incorporated into the vector representation stage of the convolutional neural network. Word context information vectors are learned through a bidirectional GRU, while a piecewise max-pooling method is employed in the pooling layer of the convolutional neural network to extract finer-grained feature information while simultaneously capturing entity pair structural information. Additionally, sentence-level attention mechanisms are integrated into the model. Experiments conducted on the NYT dataset validate the approach, and results demonstrate that the proposed method effectively improves the accuracy and recall of entity relation extraction.

Full Text

Preamble

Vol. 37 No. 3
Application Research of Computers (ChinaXiv Cooperative Journal)
Accepted Paper

Relation Extraction Based on Recurrent Convolutional Neural Network

Wan Jing[1], Li Haoming[1], Yan Huanchun[1], Zhang Xuechao[2]†
[1] College of Information Science & Technology, Beijing University of Chemical

Technology, Beijing 100029, China
[2] College of Joint Logistics, National Defence University, Beijing 100091, China

Abstract: Most existing relation extraction approaches cannot learn long-distance dependency information from long sentences containing co-occurring entities. This paper proposes a novel entity relation extraction model based on recurrent convolutional neural networks and sentence-level attention mechanisms. The model incorporates Bi-GRU neural networks, which excel at handling long-distance dependencies, into the vector representation phase of convolutional neural networks. The Bi-GRU learns contextual information vectors for words, while the pooling layer employs a piecewise max pooling method to capture structural information between entity pairs and extract finer-grained features. Additionally, the model integrates a sentence-level attention mechanism. Experiments conducted on the NYT dataset demonstrate that the proposed method effectively improves both the precision and recall of entity relation extraction.

Keywords: GRU; recurrent convolutional neural networks; attention mechanism; relation extraction
Classification: TP391
DOI: 10.3969/j.issn.1001-3695.2018.09.0635

0 Introduction

The rapid development of the Internet has brought massive amounts of data while simultaneously increasing the importance of efficiently extracting valuable information. In recent years, the establishment of knowledge bases such as Freebase, DBpedia, and YAGO has enabled widespread application in NLP tasks. Against this backdrop, information extraction technology has emerged with the goal of extracting structured information from unstructured natural language text, primarily focusing on three types of content: named entities, relations, and events.

Entity relation extraction, as a subtask of information extraction, has remained a research hotspot in both academia and industry in recent years, holding significant importance for frontier fields such as information retrieval, question answering systems, and intelligent recommendation. Traditional entity relation extraction methods require manual feature engineering and corpus annotation, consuming substantial time and human resources. Feature selection directly impacts the final performance of relation classifiers, and the use of NLP annotation tools often leads to error propagation problems.

Deep neural network models that have emerged in recent years can automatically learn from large-scale text corpora through deep networks. Convolutional neural networks (CNNs) have gradually been applied to entity relation extraction tasks due to their excellent feature extraction capabilities. However, for

longer entity co-occurrence sentences in text corpora, CNNs cannot learn long-distance dependency information. Moreover, for entity relation extraction tasks, while ordinary max pooling can extract the most valuable feature information, it fails to capture structural information between two entities.

To address these limitations, this paper focuses on an entity relation extraction method based on recurrent convolutional neural networks and attention mechanisms, aiming to improve extraction performance by combining recurrent neural networks with convolutional neural networks and incorporating attention mechanisms. To tackle the problem of simple CNNs being unable to learn long-distance dependencies, we propose adding GRU (Gated Recurrent Unit), a recurrent neural network adept at handling long-distance dependencies, to the vector representation phase of CNNs. To address the inability of ordinary max pooling to capture structural information between entities, we propose adopting a piecewise max pooling method in the pooling layer of CNNs. During pooling, the entire sentence vector is divided into three segments using the two entities as dividing points, with max pooling performed on each segment separately before concatenating the three pooled vectors. Additionally, we propose incorporating a sentence-level attention mechanism into the relation extraction model to improve accuracy.

This paper treats entity relation extraction as a multi-classification problem, investigating the combined use of recurrent and convolutional neural networks to more accurately classify relations between entities, and enhances extraction effectiveness through the addition of an attention mechanism.

---

1 Model Based on GRU_PCNN and Attention Mechanism

The proposed relation extraction method based on recurrent convolutional neural networks and attention mechanisms consists of three stages: the vector representation stage based on bidirectional GRU, the feature learning stage based on PCNN, and the attention weight learning stage.

1.1 Vector Representation Based on GRU

The vector representation stage based on GRU is primarily responsible for converting raw input sentences into vector forms suitable for neural network processing to enable subsequent feature extraction and learning operations. In current entity relation extraction tasks, word vectors (distributed representations) are commonly used as neural network inputs. This paper employs word vectors while additionally incorporating position vectors to capture relative position information between words and entities, providing more features for the model. Furthermore, we propose adding recurrent neural networks to the vector representation stage, leveraging their "memory" capability to encode contextual feature vectors and provide richer features for the relation extraction model.

The term "word" in this context includes not only single words but also entity phrases, such as "Tiger Woods," which is a person name entity.

1.1.1 Word Distributed Representation Word distributed representation is based on the distributional hypothesis that "the context of a word determines its semantics." It constructs semantic information-bearing vectors by modeling word contexts, also known as "word vectors" or "word embeddings." The fundamental approach maps each word in text to a new space represented by high-density, low-dimensional continuous real-valued vectors. For an input sentence S containing t words, each word w_i is converted into a d_w-dimensional real-valued vector.

1.1.2 Position Feature Representation In relation extraction tasks, words closer to the two entities contribute more to target relation extraction. Therefore, this paper introduces position features during the vector representation stage to provide the model with positional structural information that helps determine the location and type of relation words. For input sentence S, position features are obtained by calculating the relative distance from the current word w_i to the two entities (head entity and tail entity). For example, in the sentence "···spread that Earl Woods, the father of Tiger Woods, had···" , the entity phrases "Earl Woods" and "Tiger Woods" are the head and tail entities respectively, with the word "father" having distances of 3 and -2 to these entities. In our model, the relative distances from word w_i to the two entities are mapped into d_p-dimensional vectors and combined to form the word' s position feature representation.

1.1.3 Contextual Information Vector Representation Based on Bi-GRU Recurrent Neural Networks (RNNs) have been used in many NLP tasks due to their ability to model text sequences, achieving promising results. However, gradient vanishing makes it difficult for RNNs to learn long-distance dependencies. To address this, Long Short-Term Memory (LSTM) networks were proposed, which use three gates to control memory and forgetting of learned sequence information. The forget gate determines how much information from the previous time step is retained; the input gate controls how much of the current input is preserved; and the output gate regulates how much information is output. GRU (Gated Recurrent Unit), proposed by Cho et al. in 2014, is a simplified variant of LSTM that merges the forget and input gates into a single update gate. Research has shown that GRU can achieve better results than LSTM with fewer parameters. Therefore, this paper uses GRU to learn contextual information for words.

The GRU model contains two gates: the reset gate r and the update gate z, as shown in [Figure 2: see original paper]. For word w_i in a sentence, learning its preceding context vector requires the distributed representation vector of the previous word and the preceding context vector. When calculating the preceding context vector for the first word, we initialize them as zero vectors.

The weight matrices W_r, W_z, and W_c are separate for each component due to vector concatenation.

The reset gate helps discard irrelevant information when its sigmoid function output is close to 0, resetting current input information to obtain a more concise and valuable vector representation. The update gate controls the amount of preceding information participating in current learning, helping the model memorize long-distance dependencies.

The forward propagation GRU model shown in [Figure 2: see original paper] helps memorize preceding information to learn long-distance dependencies from earlier text. However, our vector representation learning module must memorize both preceding and following contextual information. Therefore, we employ a bidirectional GRU (Bi-GRU) model, where the forward GRU learns the preceding context vector c_l and the backward GRU learns the following context vector c_r.

For example, in the sentence "···spread that Earl Woods, the father of Tiger Woods, had···" , the word "father' s" preceding context vector c_l results from learning the left text "···spread that Earl Woods, the," while its following context vector c_r learns from the right text "of Tiger Woods, had···" . The following context vector learning process is similar but processes input in reverse order.

After these calculations, we obtain the word vector e_i, preceding context vector c_l, following context vector c_r, and position feature vector p_i for each word w_i. These are concatenated to form a final vector representation x_i that incorporates semantic, contextual, and positional features:

$$x_i = [e_i; p_i; c_l; c_r]$$

where d_l and d_r are the dimensions of the context vectors.

1.2 PCNN Feature Learning Stage

1.2.1 Convolution In relation extraction tasks, we predict the relationship between two target entities by learning from sentences where they co-occur, requiring feature extraction from the entire sentence. This paper employs CNNs, which excel at feature extraction, to learn from all features obtained in the representation layer. The convolution layer performs feature extraction by applying convolution operations within a sliding window. For sentence $S = \{x_1, x_2, ..., x_t\}$ where x_i is the vector representation from the previous stage, we define $x_{i:i+l-1}$ as the concatenation of the sequence with window length l, yielding convolution matrix $W \in \mathbb{R}^{d_c \times (d_w + 2d_p + d_l + d_r)}$.

To learn diverse features, multiple convolution kernels are typically used. If the model employs n convolution kernels, the convolution operation can be expressed as:

$$c_{i,j} = W_j \cdot x_{i:i+l-1} + b_j$$

where j indexes the convolution kernel. For positions near sentence boundaries, we pad vectors beyond the sentence range with zero values. After convolution, we obtain the convolution layer output vector $c = \{c_1, c_2, ..., c_n\}$.

1.2.2 Piecewise Max Pooling Strategy The pooling layer follows the convolution layer to further extract features, reducing network complexity while capturing primary characteristics. While standard max pooling selects the maximum value from each convolution kernel' s feature sequence, it cannot capture structural information between entities or extract fine-grained features for entity relation extraction. Therefore, this paper adopts piecewise max pooling, which divides the entire sentence into three segments using the two target entities as boundaries and applies max pooling to each segment separately.

Each convolution kernel' s result vector c_i is split into three segments $\{c_{i1}, c_{i2}, c_{i3}\}$. The piecewise max pooling operation is:

$$p_{i,j} = \max(c_{i,j}) \quad \text{for } j = 1, 2, 3$$

The three pooled vectors are then concatenated to form $p_i = [p_{i,1}, p_{i,2}, p_{i,3}]$. Connecting all p_i and applying a non-linear function yields the final pooling layer output vector $s \in \mathbb{R}^{3n}$. This transforms the sentence feature vector into a fixed-length representation independent of the original sentence length.

1.3 Attention Mechanism

1.3.1 Attention Mechanism Layer The sentence-level attention mechanism layer learns an attention value (weight) for each entity co-occurrence sentence. Sentences that correctly express the target relation receive higher weights, while incorrectly labeled sentences receive very low weights.

For an entity pair (e_1, e_2), all their co-occurring sentences form set $T = \{s_1, s_2, ..., s_q\}$. The attention layer computes corresponding weights $\{\alpha_1, \alpha_2, ..., \alpha_q\}$, allowing the aggregated representation of set T to be calculated as:

$$T = \sum_{i=1}^{q} \alpha_i s_i$$

We compute sentence weights by measuring similarity between sentence feature vectors and target relation vectors. Inspired by TransE, which represents knowledge graph relations as translation operations from head to tail entities, we hypothesize that when a sentence' s feature representation vector s_i is more

similar to the target relation vector r, the sentence is more likely to correctly express that relation, deserving higher attention weight α_i. The weight calculation is defined as:

$$\alpha_i = \frac{\exp(\omega_i)}{\sum_{j=1}^{q} \exp(\omega_j)}$$

where ω_i is the matching score between sentence feature vector s_i and predicted relation vector r, computed as:

$$\omega_i = \tanh(W_a \cdot [s_i; r] + b_a)$$

with W_a as an intermediate matrix and b_a as a bias vector.

1.3.2 Softmax Layer Softmax is a multi-class classification model that generalizes logistic regression. In relation multi-classification, relation category labels correspond to multiple values. The sentence feature representation weighted by attention is fed into the Softmax layer to compute matching scores for all relation types:

$$o = W_s \cdot T + b_s$$

where W_s is the weight matrix and b_s is the bias vector. The conditional probability for the i-th relation type is:

$$p(r_i|T) = \frac{\exp(o_i)}{\sum_{j=1}^{n_r} \exp(o_j)}$$

where n_r is the total number of relation types.

The model parameters are learned by minimizing the negative log-likelihood of the training data. Assuming the training data contains m sentence sets, each representing one relation type, the objective function is:

$$J(\theta) = -\sum_{i=1}^{m} \log p(r_i|T_i; \theta)$$

where θ represents all model parameters.

2 Experiments

2.1 Experimental Setup

2.1.1 Experimental Data The knowledge graph commonly used in relation extraction research is Freebase, a large-scale knowledge graph containing over 40 million entities, tens of thousands of attribute relations, and more than 2.4 billion fact triples. Correspondingly, the New York Times (NYT) dataset serves as the text corpus, covering all NYT news from 1987-2007 and containing numerous Freebase entities. The dataset generated through distant supervision provides abundant entity co-occurrence sentences.

This paper uses the entity relation annotation dataset created by Riedel et al. in 2010, which aligns NYT corpus with Freebase. The annotated NYT corpus is divided into two parts: news from 2005-2006 containing 281,270 entity pairs and 522,611 co-occurrence sentences for training, and news from 2007 containing 96,678 entity pairs and 172,448 sentences for testing. The corpus contains 53 relation types, with "NA" indicating no relation between entities, as shown in [Figure 3: see original paper]. presents example instances from the dataset.

2.1.2 Evaluation Metrics Precision (P), recall (R), and F-measure (F1) are standard evaluation metrics for entity relation extraction, calculated as:

$$P = \frac{\text{Number of correctly extracted relations}}{\text{Total number of extracted relations}}$$

$$R = \frac{\text{Number of correctly extracted relations}}{\text{Total number of relations in samples}}$$

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

Additionally, we employ Precision-Recall Curves (PRC) for more intuitive comparison with other algorithms.

2.1.3 Word Vectors In our entity relation extraction model, word vectors constitute an essential part of the PCNN input and serve as input to the GRU network for learning contextual information. Randomly initialized word vectors may lead to unstable training results. Therefore, we use word vectors trained by distributed representation models on large-scale corpora. Word2vec, open-sourced by Google in 2013, implements Skip-gram and CBOW models, with Skip-gram demonstrating superior performance in capturing semantic relationships. We employ word2vec with the Skip-gram model to train English word vectors on the NYT text dataset.

2.1.4 Experimental Parameters We use cross-validation on the training set to optimize model parameters. Parameter ranges are set as follows: word vector dimension $\{50, 100, 200, 300\}$, position feature dimension $\{5, 10, 20\}$, number of convolution kernels $\{50, 100, 150, 200, 250\}$, learning rate $\{0.1, 0.01, 0.001\}$, convolution window size $\{3, 5, 7\}$, and batch size $\{50, 100, 150, 200\}$. The final parameter settings are shown in .

2.1.5 Impact of Bidirectional GRU We propose adding Bi-GRU, which excels at learning long-distance sequential dependencies with a simple network structure, to the vector representation stage. The forward and backward GRUs learn preceding and following context vectors respectively, providing rich feature information for subsequent convolutional learning. To validate our approach, we design comparative experiments between PCNN with attention (PCNN_ATT) and GRU-enhanced PCNN with attention (GRU_PCNN_ATT). For fair comparison, we add sentence-level attention to the standalone PCNN model. The PCNN_ATT input consists of word and position feature vectors, while GRU_PCNN_ATT additionally includes context vectors obtained through Bi-GRU.

2.1.6 Impact of Sentence-Level Attention Mechanism We incorporate sentence-level attention to compute attention weights for each sentence, increasing weights for sentences that correctly express target relations while decreasing weights for incorrectly labeled sentences to reduce their interference. We compare three approaches: GRU_PCNN_ONE (randomly selecting one sentence per entity pair), GRU_PCNN_AVG (using equal weights for all co-occurring sentences), and GRU_PCNN_ATT (our proposed attention-based model).

2.2 Results and Analysis

2.2.1 Bidirectional GRU Impact Analysis and [Figure 4: see original paper] demonstrate that the relation extraction model with bidirectional GRU outperforms standard PCNN, with GRU_PCNN_ATT achieving approximately 0.05 higher precision across the entire recall range. This is because GRU, as an RNN, possesses excellent sequential feature memorization capabilities and, like LSTM, excels at learning long-distance dependencies. The experimental corpus contains many long-dependency sentences that PCNN_ATT cannot capture with its sliding window-based local context, whereas the memory-enabled bidirectional GRU_PCNN_ATT can learn rich contextual features, yielding superior results.

2.2.2 Attention Mechanism Impact Analysis presents experimental results showing that our attention-based method (GRU_PCNN_ATT) outperforms random selection (GRU_PCNN_ONE) and equal weighting (GRU_PCNN_AVG) by 3-4% across different test set sizes, demonstrating that sentence-level attention improves extraction accuracy. Notably, the equal-weight model (GRU_PCNN_AVG) performs significantly worse, particularly

on the full test set, because while it learns from more sentences, it also assigns equal weight to incorrectly labeled sentences, introducing noise that degrades model performance.

2.2.3 GRU_PCNN_ATT Method Validation and [Figure 5: see original paper] compare our method with three classic distant supervision approaches: Mintz, MultiR, and MIML. The results show that our method substantially outperforms these traditional methods in both precision and recall. The PRC curves demonstrate that GRU_PCNN_ATT maintains higher precision across all recall ranges. When recall exceeds 0.1, feature-based methods experience sharp precision drops, whereas our method maintains strong performance. This is because manually designed features cannot accurately capture semantic information, and NLP annotation tools inevitably introduce errors. In contrast, our neural network-based approach avoids such tool errors and learns semantic information more accurately.

---

3 Conclusion

This paper proposes an entity relation extraction method based on recurrent convolutional neural networks and attention mechanisms. By employing bidirectional GRU to learn word contextual information, using recurrent convolutional neural networks to obtain finer-grained features and extract inter-entity structural information, and incorporating sentence-level attention mechanisms, our approach effectively addresses limitations in existing methods. Through comprehensive experimental comparisons, we demonstrate that the proposed method significantly improves entity relation extraction performance.

References

[1] Huang Xun, You Hongliang, Yu Yang. A review of relation extraction [J]. New Technology of Library & Information Service, 2013(11).

[2] Kurt B, Colin E, Praveen P, et al. Freebase: a collaboratively created graph database for structuring human knowledge [C]// Proc of KDD. 2008: 1247-1250.

[3] Lehmann J. DBpedia: a nucleus for a web of open data [C]// Proc of Semantic Web, International Semantic Web Conference & Asian Semantic Web Conference. 2007: 11-15.

[4] Liu Chunyang, Sun Wenbo, Chao Wenhan, et al. Convolution neural network for relation extraction [C]// Proc of International Conference on Advanced Data Mining and Applications. Berlin: Springer, 2013.

[5] Lin Yankai, Shen Shiqi, Liu Zhiyuan, et al. Neural relation extraction with selective attention over instances [C]// Proc of the 54th Annual Meeting of

the Association for Computational Linguistics, Volume 1: Long Papers. 2016: 2124-2133.

[6] Zeng Daojian, Liu Kang, Lai Siwei, et al. Relation classification via convolutional deep neural network [C]// Proc of the 25th International Conference on Computational Linguistics: Technical Papers. 2014.

[7] Zhou Peng, Shi Wei, Tian Jun, et al. Attention-based bidirectional long short-term memory networks for relation classification [C]// Proc of the 54th Annual Meeting of the Association for Computational Linguistics, Volume 2: Short Papers. 2016: 207-212.

[8] Nguyen T H, Grishman R. Relation extraction: perspective from convolutional neural networks [C]// Proc of the 1st Workshop on Vector Space Modeling for Natural Language Processing. 2015: 39-48.

[9] Zeng Daojian, Liu Kang, Chen Yubo, et al. Distant supervision for relation extraction via piecewise convolutional neural networks [C]// Empirical Methods in Natural Language Processing. 2015: 1753-1762.

[10] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735-1780.

[11] Cho K, Van Merrienboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation [C]// Proc of Empirical Methods in Natural Language Processing. 2014: 1724-1734.

[12] Shen Yatian, Huang Xuanjing. Attention-based convolutional neural network for semantic relation extraction [C]// Proc of the 26th International Conference on Computational Linguistics: Technical Papers. 2016: 2526-2536.

[13] Mikolov T, Sutskever I, Chen Kai, et al. Distributed representations of words and phrases and their compositionality [C]// Advances in Neural Information Processing Systems. 2013: 3111-3119.

[14] Jozefowicz R, Zaremba W, Sutskever I. An empirical exploration of recurrent network architectures [C]// Proc of International Conference on Machine Learning. 2015: 2342-2350.

[15] Mnih V, Heess N, Graves A. Recurrent models of visual attention [C]// Advances in Neural Information Processing Systems. 2014: 2204-2212.

[16] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate [C]// Proc of ICLR. 2015.

[17] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data [C]// Neural Information Processing Systems. 2013: 2787-2795.

Note: Figure translations are in progress. See original paper for figures.