Machine Reading Comprehension Using Structural Knowledge Graph-aware Network

Delai Qiu1, Yuanzhe Zhang2, Xinwei Feng3, Xiangwen Liao4, Wenbin Jiang4, Yajuan Lyu1, Kang Liu2,3, Jun Zhao2,3

1 College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China
2 Institute of Automation, Chinese Academy of Sciences, Beijing, China
3 University of Chinese Academy of Sciences, Beijing, China
4 Baidu Inc., Beijing, China

noneqdl@gmail.com, {yzzhang, kliu, jzhao}@nlpr.ia.ac.cn
lxw@fzu.edu.cn, {fengxinwei, jiangwenbin, lvyajuan}@baidu.com

Abstract

Leveraging external knowledge is an emerging trend in machine comprehension task. Previous work usually utilizes knowledge graphs such as ConceptNet as external knowledge, and extracts triples from them to enhance the initial representation of the machine comprehension context. However, such method cannot capture the structural information in the knowledge graph. To this end, we propose a Structural Knowledge Graph-aware Network (SKG) model, constructing sub-graphs for entities in the machine comprehension context. Our method dynamically updates the representation of the knowledge according to the structural information of the constructed sub-graph. Experiments show that SKG achieves state-of-the-art performance on the ReCoRD dataset.

1 Introduction

Machine reading comprehension (MRC) is an important subtask in natural language processing, which requires a system to read a given passage and answer questions about it. The ability of utilizing external knowledge is of great significance in an MRC system (Rajpurkar et al., 2016; Trischler et al., 2017). Latest large-scale datasets, e.g. ReCoRD (Zhang et al., 2018) specify that external knowledge is required to answer questions.

Many previous studies have introduced external knowledge in machine comprehension (Weissenborn, 2017; Mihaylov and Frank, 2018; Bauer et al., 2018). They often acquire external knowledge from structural knowledge graphs, such as ConceptNet (Speer et al., 2017) and Freebase (Tanon et al., 2016), in which knowledge is organized by triples like “(shortage, related_to, lack)” and “(need, related_to, lack)”. However, most of them fail to make full use of the structural information in the knowledge graph, and use sequence modeling methods like recurrent neural networks (RNNs) to generate the representation of knowledge, rather than based on graph structure.

In this paper, we present a Structural Knowledge Graph-aware Network (SKG) model to leverage the structural information of external knowledge. The advantages of our proposed method are two folds: a) the constructed sub-graphs contain both MRC context and external knowledge nodes, reserving the knowledge structure; b) the graph neural network is capable of updating the nodes dynamically according to the structure of sub-graphs, instead of using external knowledge as pre-trained representations. Concretely, we first construct sub-graphs from external knowledge such as ConceptNet based on the context, and initialize the representation of nodes via knowledge graph embedding method (Yang et al., 2015). Then we employ graph attention networks to dynamically update the representation of nodes on sub-graphs. Finally, we utilize the final representation of nodes to augment the representation of context via gate mechanisms.

Our contributions can be summarized as:

- We present a simple but effective method to construct sub-graphs from a knowledge graph, which can reserve the structure of knowledge;
- Graph attention networks are employed to dynamically update the representation of knowledge based on sub-graph structure, which can make full advantage of structural information of external knowledge;
- Experiments demonstrate that SKG model is able to effectively leverage external knowledge in MRC task, and achieves state-of-the-art performance on the ReCoRD dataset.
2 Related work

External Knowledge Enhanced MRC Models
There are several models that use knowledge for machine comprehension (Yang and Mitchell-I, 2017; Mihaylov and Frank, 2018; Weissenborn, 2017; Bauer et al., 2018; Pan et al., 2019). Mihaylov and Frank (2018) relies on the ability of the attention mechanism to retrieve relevant pieces of knowledge, and Bauer et al. (2018) employs multi-hop commonsense paths to help multi-hop reasoning. They treat retrieved knowledge triples as sequences and use sequence modeling methods to compress the representation of knowledge, which are not based on graph structure. On the contrary, we organize knowledge as sub-graphs, then update the representation of nodes on sub-graphs with graph neural network.

Graph Neural Networks
Graph neural networks (Kipf and Welling, 2016; Schlichtkrull et al., 2018; Velickovic et al., 2017) have been shown successful on many Natural Language Processing (NLP) tasks (Cao et al., 2018; Zhou et al., 2018; Song et al., 2018; Cao et al., 2019). They are good at dealing with graph-structured data. In (Cao et al., 2018; Song et al., 2018; Cao et al., 2019), Graph Convolutional Networks (GCNs) have been applied in multi-document machine comprehension for multi-hop reasoning question answering, but they consider only the internal structure information in the MRC context without incorporating external knowledge. To the best of our knowledge, our work is the first to study graph attention networks in machine comprehension with external knowledge.

3 SKG Model

The architecture of SKG is shown in Figure 1. It contains four modules: (1) Question and paragraph modeling module, which acquires the contextual representation of question and paragraph with BERT; (2) Knowledge sub-graph construction module, which retrieves sub-graphs from knowledge graph based on the context; (3) Graph attention module, which updates the representation of nodes on graph; (4) Output layer module, which is employed to generate the final answer.

3.1 Question and Paragraph Modeling

We first encode the tokens with BERT (Devlin et al., 2018). BERT has become one of the most successful natural language representation models on various NLP datasets, including SQuAD (Rajpurkar et al., 2016). BERT’s model architecture is a multi-layer bidirectional Transformer (Vaswani et al., 2017) encoder, which is pre-trained on large-scale corpus. We represent the input question and paragraph as a single packed sequence as follows:

$$[CLS]Question[SEP]Paragraph[SEP]$$

where $[CLS]$ is a specific classifier token and $[SEP]$ is a sentence separator which are defined in BERT. And we use BERT to generate the contextual representation of question and paragraph. The final hidden output from BERT for the $i^{th}$ input token is denoted as $t_{b_i} \in \mathbb{R}^H$, and $H$ is the output hidden size of BERT model.

3.2 Knowledge Sub-Graph Construction

We first use knowledge graph embedding approach to generate the initial representation of nodes and edges in terms of the whole knowledge graph.

We consider a knowledge triple in a knowledge graph as $(head, relation, tail)$. For the $i^{th}$ token $t_i$ in paragraph, we retrieve all triples whose head or tail contains lemmas of the token. Take token “shortage” as an example, we retrieve triples like “(shortage, related_to, lack)”. Then, we retrieve the neighbor triple of them, and reserve ones that contain lemmas of any token of the question. Thus we can acquire triples like “(need, related_to, lack)”. We reorganize these triples into a sub-graph via connecting identical entities and reserving the relations as edges in these triples. So
we can construct a simple sub-graph like “(shortage, related to, lack, related to, need)”, where “lack” is the identical entity. The sub-graph can be denoted as \( g_i \), and the nodes and edges of it are initiated by the embeddings above.

### 3.3 Graph Attention

Our graph attention network is designed to update the representation of the nodes in a constructed sub-graph, which is inspired by (Veličković et al., 2018; Zhou et al., 2018). For the \( i \)-th token \( t_i \) in paragraph, its sub-graph is \( g_i = \{n_1, n_2, ..., n_k\} \), where \( k \) is the number of nodes. And \( N_j \) is the set of the \( j \)-th node neighbors.

The representation of nodes is updated \( L \) times. At the \( l \)-th update, the updating rules are as follows, which are designed to model the interaction between the \( j \)-th node and its neighbor nodes. In this way, the nodes are dynamically updated according to the structure of sub-graphs.

\[
\begin{align*}
h_j^{l+1} &= \sum_{n=1}^{N_j} \alpha_n t_n^l \\
\alpha_n &= \frac{\exp(\beta_n)}{\sum_{j=1}^{N_j} \exp(\beta_j)} \\
\beta_n &= (W_r^j r_n^l) \cdot \tanh(W_b^j h_n^l + W_t^j t_n^l)
\end{align*}
\] (1)

(2)

(3)

where \( h_j^l \in \mathbb{R}^d \) is hidden state of the \( j \)-th node, and its neighbor’s hidden state is \( t_n^l \), and the hidden state of relation is \( r_n^l \), and \( d \) is the hidden state dimension. \( W_r^j, W_b^j \) and \( W_t^j \) are trainable weight matrices for the node, its neighbors and relations respectively. After \( L \) updates, we can get the final hidden state of the central node as the final representation, which can be denoted as \( t_k \).

### 3.4 Output Layer

In the output layer, we combine this knowledge representation \( t_k \) with the textual representation \( t_{\text{h}} \) via a sigmoid gate, since external knowledge is not always necessary for reasoning.

\[
w_i = \sigma((W^t [t_{\text{h}}; t_k]))
\] (4)

\[
t_i' = w_i \odot t_{\text{h}} + (1 - w_i) \odot t_k
\] (5)

We denote \( T = \{t_1', t_2', ..., t_n'\} \) as the final representation, where \( t_i' \in \mathbb{R}^H \). And we study a start vector \( S \in \mathbb{R}^H \) and an end vector \( E \in \mathbb{R}^H \) like (Devlin et al., 2018), which take \( T \) as input. Then, the probability of the \( i \)-th token being the start of the answer span is computed as a dot product between \( T_i \) and \( S \) followed by a softmax over all of the words in the paragraph:

\[
P_i^s = \frac{e^{S^T T_i}}{\sum_j e^{S^T T_j}}
\] (6)

Let \( P_i^e \) be the probability of the \( i \)-th token to be the end of an answer span, which can be calculated by the same above formula, and the maximum scoring span is used as the answer. The training objective is the loglikelihood of the correct start and end positions.

### 4 Experiments

#### 4.1 Dataset

**ReCoRD** We report results on ReCoRD dataset (Zhang et al., 2018), a large-scale dataset for machine comprehension requiring external knowledge. There are 100,730, 10,000 and 10,000 examples in the training set, the development set and the test set respectively. The test set is not public, which needs to submit the model to the organization\(^1\) to get the results.

**External Knowledge** We consider two knowledge sources as our external knowledge: WordNet and ConceptNet. For WordNet, we use the preprocessed data provided by Bordes et at. (2013), which contains 151,442 triples with 40,943 synsets and 18 relations. For ConceptNet, we use the preprocessed data provided by Bauer et al. (2018), which contains 2,808,998 triples with 978,672 entities and 46 relations.

#### 4.2 Implementation Details

Our model is implemented with pytorch\(^2\), and uses the framework\(^3\) for BERT model. We employ the open-source framework OpenKE (Han et al., 2018) to obtain the embedding of entities and relations with the BILINEAR model (Yang et al., 2015). The size of embedding of entities and relations is 100. The update times \( L \) of graph attention network is set to 5. We use Adam optimizer. The learning rate uses the linear schedule to decrease from 0.00003 to 0.

\(^1\)https://sheng-z.github.io/ReCoRD-explorer/

\(^2\)https://github.com/pytorch/pytorch

\(^3\)https://github.com/huggingface/pytorch-pretrained-BERT
Table 1: The performance of different models on ReCoRD dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>QANet (Yu et al., 2018)</td>
<td>35.38</td>
<td>36.51</td>
</tr>
<tr>
<td>SAN (Liu et al., 2018)</td>
<td>38.14</td>
<td>39.77</td>
</tr>
<tr>
<td>DocQA w/o ELMo (Clark and Gardner, 2018)</td>
<td>35.59</td>
<td>38.52</td>
</tr>
<tr>
<td>DocQA w/ ELMo (Clark and Gardner, 2018)</td>
<td>44.13</td>
<td>45.44</td>
</tr>
<tr>
<td>SKG+BERT-Large(ours)</td>
<td>70.94</td>
<td>72.24</td>
</tr>
</tbody>
</table>

Table 2: The effectiveness of introducing external knowledge.

<table>
<thead>
<tr>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-Base( 2018)</td>
<td>54.03</td>
<td>55.99</td>
</tr>
<tr>
<td>SKG+BERT-Base(ours)</td>
<td>60.26</td>
<td>60.79</td>
</tr>
<tr>
<td>BERT-Large( 2018)</td>
<td>64.28</td>
<td>66.60</td>
</tr>
<tr>
<td>SKG+BERT-Large(ours)</td>
<td>70.94</td>
<td>71.55</td>
</tr>
</tbody>
</table>

Table 3: The results of different ways of introducing external knowledge.

<table>
<thead>
<tr>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHPGM+NOIC( 2018)</td>
<td>28.36</td>
<td>29.29</td>
</tr>
<tr>
<td>KG+LSTM+Bert-Base</td>
<td>58.01</td>
<td>58.49</td>
</tr>
<tr>
<td>SKG+BERT-Base(ours)</td>
<td>60.26</td>
<td>60.79</td>
</tr>
</tbody>
</table>

4.3 Results and Analysis

Results on ReCoRD

We choose several baselines: (1) QANet (Yu et al., 2018) is one of the top MRC models, which is different from many other MRC models due to the use of transformer (Vaswani et al., 2017). (2) SAN (Liu et al., 2018) is a top-rank MRC model, which employs a stochastic answer module. (3) DocQA (Clark and Gardner, 2018) is a strong baseline model, which consists of bidirectional attention flow and self-attention.

The results of different models are shown in Table 1. Our SKG+BERT-Large\(^4\) model achieves better performance than all previous published models, which is 26.13% higher in value than the state-of-art model DocQA with ELMo.

The Effectiveness of External Knowledge

Moreover, ablation experimental results on the dev set\(^5\) are given in Table 2. Once we remove the module incorporating external knowledge, the model degenerates into the fine-tuning BERT model, and the results show significant performance drop with 4.80% and 4.95% on different pre-trained model sizes respectively. The results demonstrate that our model can effectively utilize external knowledge.

Different Ways to Use External Knowledge

In the original paper of ReCoRD (Zhang et al., 2018), there is no research on existing models that use external knowledge to improve MRC task. So we study the recent model MHPGM+NOIC (Bauer et al., 2018), which utilizes multi-hop relational knowledge paths from ConceptNet. As shown in Table 3, our SKG model is more proper for introducing external knowledge. In addition, to investigate the impact of the structural information on performance, we replace our sub-graph construction module with KG+LSTM, which retrieves knowledge triples without reconstructing the structure of them, and considers paths among them as sequences. We employ Long Short-Term Memory (LSTM) model to generate the representation of knowledge. As shown in Table 3, the performance drops 2.3% in F1, which means that the incorporation of structural information in the knowledge graph is able to make better use of external knowledge.

5 Conclusion

We propose SKG model for improving machine comprehension. Rather than treating triples from knowledge graph independently and separately, we construct sub-graphs from external knowledge. Then we generate the representation of knowledge with graph attention networks to improve the representation of context. Experimental results indicate that our model achieves the best performance in the challenging ReCoRD dataset.

---

\(^4\)Our first module is based on different BERT model sizes. BERT-Base contains 110M parameters and BERT-Large contains 340M parameters.

\(^5\)Due to several weeks evaluation wait-time on the non-public test set, we just report results of our supplementary experiments on the dev set. In previous submissions, the results in dev set are consistent with test set.
Acknowledgements

This work was supported by National Key R&D Program of China under Grant 2018YFB1005100, the National Natural Science Foundation of China (No.61533018, No.U1605251). This work was also supported by the Open Project of National Laboratory of Pattern Recognition at the Institute of Automation of the Chinese Academy of Sciences (201900041) and the independent research project of National Laboratory of Pattern Recognition.

References


Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Li, Maosong Sun, and Jianzi Li. 2018. Openke: An open toolkit for knowledge embedding. In Proceedings of EMNLP, pages 139–144.


Xiaoman Pan, Kai Sun, Dian Yu, Heng Ji, and Dong Yu. 2019. Improving question answering with external knowledge.


