ENCODING KNOWLEDGE GRAPH WITH GRAPH CNN FOR QUESTION ANSWERING

Leo Laugier*, **Anran Wang***, **Chuan-Sheng Foo, Vijay Chandrasekhar**
Institute for Infocomm Research, A*STAR, Singapore
leojlaugier@gmail.com
{wang_anran,foo_chuan_sheng,vijay}@i2r.a-star.edu.sg

ABSTRACT

Question answering remains a challenge for machines, partly due to the ambiguity of natural language and implied context, which often requires external knowledge to resolve. In this work, we present a general framework for incorporating such external knowledge (encoded as knowledge graphs) into question answering systems using graph convolutional neural networks. We applied our framework on top of Stochastic Answer Networks (SAN), a state-of-the-art method for question answering, and evaluated the system on the Stanford Question Answering Dataset 2.0. Our results show that leveraging knowledge brings significant improvements in terms of EM and F1 scores, validating the importance of incorporating external knowledge in understanding textual context.

1 INTRODUCTION

Question answering with text or machine reading comprehension entails answering questions according to the given textual context. By exploring various attention mechanisms and model structures, existing methods (Xiong et al., 2017; Seo et al., 2017; Devlin et al., 2018) have achieved a substantial leap of performance in this task. However, question answering remains a challenging task. This is partially because natural language understanding involves ambiguity and implied context. Answering questions sometimes requires essential common sense knowledge or related background facts aside from the given context.

Several methods have been proposed to leverage knowledge graphs for question answering tasks (Mihaylov & Frank, 2018; Shen et al., 2018). However, they either only utilize entity embeddings pre-trained on the knowledge graph (Shen et al., 2018), or incorporate fact-triplets extracted from the knowledge graph (Mihaylov & Frank, 2018). Neither of these approaches can adequately exploit the graph structure of the knowledge graph, which is essential for modelling relationships between the concepts.

In this work, we focus on incorporating external knowledge in the question answering task. In particular, we utilize Graph Convolutional Neural Networks (Graph CNNs) to represent knowledge from ConceptNet (Speer et al., 2016). To model knowledge related to the context associated with a question, we build a knowledge sub-graph that includes relevant concepts and relations. A Graph CNN is then used to encode the knowledge sub-graph and generate a context-related knowledge vector. Finally, the knowledge vector is incorporated into an existing question answering method to derive a knowledge-aware context representation. Our formulation to produce knowledge-aware context representations is general and can be plugged into any question answering method. We chose the Stochastic Answer Networks (SAN) (Liu et al., 2018) as the base model in this work. Experimental results on the SQuAD2.0 dataset (Rajpurkar et al., 2018) show that our method brings a significant performance improvement to the base model, which verifies the importance of external knowledge for resolving ambiguity and supplying implied facts for question answering.

*Equal contribution

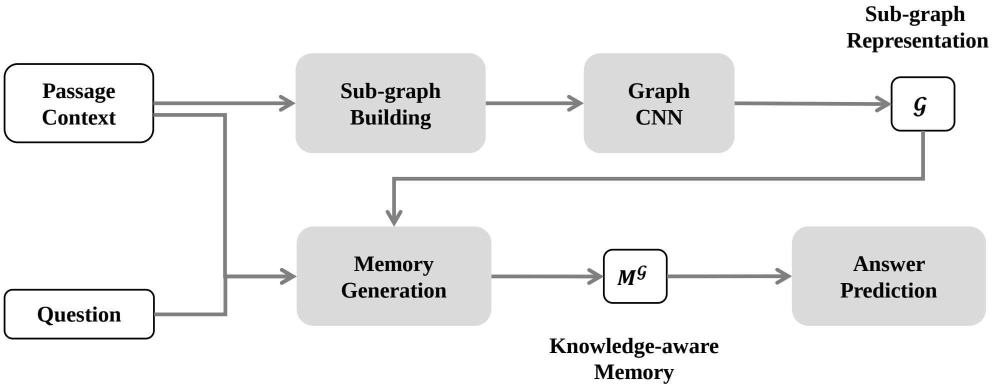

Figure 1: Illustration of our framework. Based on the ConceptNet, a knowledge sub-graph is built for the passage context associated with each question. A Graph CNN is utilized to capture knowledge from these sub-graphs and to generate knowledge-aware memory for answer prediction.

2 Related work

Incorporating external knowledge has been shown to be beneficial for applications in computer vision and natural language processing (Wu et al., 2018; Fang et al., 2017). For question answering tasks based on text, several methods have been proposed to incorporate external knowledge. For example, Shen et al. (2018) presented a method for ranking question answer pairs with external knowledge from a knowledge graph, where entities are extracted from the question answer pair and represented by embeddings pre-trained on the knowledge graph. Then a knowledge-based context representation is derived with an attention mechanism between entity representations from knowledge graph and original question answer pair. Mihaylov & Frank (2018) proposed to use external knowledge from a knowledge graph for the cloze-style reading comprehension task. They collect a set of facts from the knowledge graph as the memory, where facts are represented as triples (subject, relation, object). Key knowledge is retrieved for answer inference. In these methods, the graph structure among the concepts is not well exploited.

3 Method

In this section, we introduce our method that encodes external knowledge from a knowledge graph for question answering. We first describe the process of building knowledge sub-graphs from ConceptNet for the passage context associated with each question. Then we explain the Graph CNN model to encode knowledge from the sub-graphs. After that, we describe how to incorporate knowledge-aware context representation in the base model. An overview of the framework is shown in Figure 1.

3.1 Building and Encoding Knolwedge Sub-graphs

It is not feasible to incorporate the entire ConceptNet into the question answering model, as ConceptNet has millions of vertices and edges. We instead build a directed knowledge sub-graph for the passage context associated with a question. We extract initial concepts from the context and retrieve their one-hop neighborhood in ConceptNet as external concepts; relevant edges are also collected. The resulting sub-graph covers concepts and their neighborhood in the knowledge graph that are related to the passage context.

With knowledge sub-graphs built for the passage context, we can use a Graph CNN to encode this knowledge. A sub-graph is denoted by $G = (V, E)$, where V is the set of concepts/vertices and E is the set of edges. Concept representations are initialized with pre-trained word embeddings and we randomly initialize the edge embedding for each relationship.

Let h_i^l denote the feature vector for the i-th vertex at layer l, and e_{ij}^l be the feature vector for the edge from the j-th vertex to the i-the vertex at layer l. We follow (Bresson & Laurent, 2017) to add gating and non-linearity in the Graph CNN. All W_k^l and b_k^l are parameters to be learned. The vertex feature vector is computed as:

$$h_i^{l+1} = ReLU\Big(W_1^l h_i^l + \sum_{j \to i} \sigma\left(W_2^l e_{ij}^l + b_1^l\right) \odot \left(W_3^l h_j^l\right) + b_2^l\Big) \tag{1}$$

where $\{h_j^l : j \to i\}$ denotes the neighborhood of the i-th vertex at layer l. σ is the sigmoid function. $\odot$ is the Hadamard point-wise multiplication operator. $\sigma\left(W_2^l e_{ij}^l + b_1^l\right)$ acts as a gate which regulates the flow of information from different neighboring vertices. The edge feature vector is defined as:

$$e_{ij}^{l+1} = ReLU\Big(W_4^l h_i^l + W_5^l h_j^l + W_6^l e_{i,j}^l + b_3^l\Big) \tag{2}$$

Following (Bresson & Laurent, 2017), we add residual connections every 2 Graph CNN layers and perform batch normalization (Ioffe & Szegedy, 2015) over vertices and edges after each layer.

At the final layer L, we generate a single vector representing the knowledge sub-graph using average pooling over feature vectors of all vertices:

$$\mathcal{G} = \frac{1}{|V|} \sum_{i=1}^{|V|} h_i^L \tag{3}$$

where $|V|$ indicates the number of vertices in the entire sub-graph.

3.2 Incorporating the Graph Representation

The question answering task we consider in this paper is defined as follows. For a question $Q = \{q_0, q_1, ..., q_{m-1}\}$ and the related passage context $P = \{p_0, p_1, ..., p_{n-1}\}$ the goal is to find the answer span $A = \{a_{start}, a_{end}\}$. In this paper, we choose the Stochastic Answer Network (SAN) (Liu et al., 2018) as the base model, but our method is general and is compatible with any question answering method. The SAN method mainly consists of two modules: (1) the memory generation module, where Q and P are processed to generate the memory; (2) the answer module where the answer span is predicted based on the memory.

A shared Bidirectional Long Short-Term Memory (BiLSTM) network to derive contextual representations of Q and P as $H^q \in \mathbb{R}^{2d \times m}$ and $H^p \in \mathbb{R}^{2d \times n}$. d denotes the hidden size of the BiLSTM. The memory generation module of SAN takes H_q and H_p as input and produces memory M for answer prediction:

$$M = Mem(H_p, Hq) \tag{4}$$

Together with M, SAN also produces intermediate representations: question-aware context representations U^p, and $\hat{U}^p$ that are generated by performing self-attention on U^p.

To incorporate the knowledge captured in the sub-graph representation $\mathcal{G}$ into the SAN base model, we learn the knowledge-aware memory based on $\mathcal{G}$. We first compute the attention vector $A^{\mathcal{G}}$ between the learned vector $\mathcal{G}$ representing the knowledge sub-graph and the memory M derived by the SAN method :

$$A^{\mathcal{G}} = f_{att}(\bar{M}, \bar{\mathcal{G}}) \in \mathbb{R}^n \tag{5}$$

where graph representation $\mathcal{G}$ and memory M are first projected to a lower dimension and processed by $ReLU$ as $\bar{\mathcal{G}}$ and $\bar{M}$ before the dot product. $A^{\mathcal{G}}$ indicates the relevance or importance of the i-th memory slot with respect to $\mathcal{G}$. Then we can derive the knowledge-aware context representation $u^{\mathcal{G}}$ as:

$$u^{\mathcal{G}} = concat(U^p A^{\mathcal{G}}, \hat{U}^p A^{\mathcal{G}}, M A^{\mathcal{G}}) \tag{6}$$

where we incorporate memory M and intermediate context representations U^p and $\hat{U}^p$. Then we tile the knowledge-aware context $u^{\mathcal{G}}$ for n times as $U^{\mathcal{G}}$ and concatenate it to the memory. The final knowledge-aware memory $M^{\mathcal{G}}$ is:

$$M^{\mathcal{G}} = concat(M, U^{\mathcal{G}}) \tag{7}$$

The resulting $M^{\mathcal{G}}$ is a fusion of the question, the passage context, and the sub-graph representation, which enables the framework to be aware of hidden background knowledge beyond the context. Finally, $M^{\mathcal{G}}$ is used instead of M to be passed to the answer module.

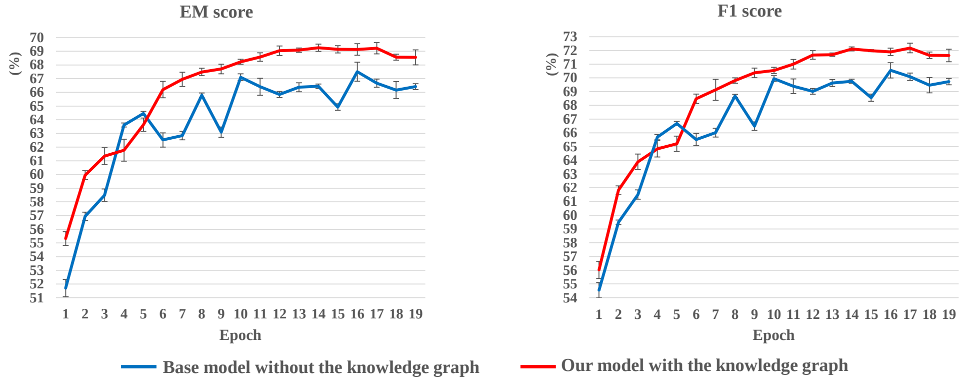

Figure 2: Comparison of EM and F1 scores for the base model and our framework incorporating external knowledge.

Table 1: EM and F1 scores of Graph CNN with different depths and concept embeddings.

Depth	EM (%)	F1 (%)
2	68.74	71.45
4	**69.70**	**72.60**

Concept Embedding	EM (%)	F1 (%)
Random	67.86	70.89
GloVe 100	68.60	71.52
GloVe 300	69.39	72.44
NumberBatch	**69.70**	**72.60**

4 Experiments

We evaluate our method on the Stanford Question Answering Dataset (SQuAD) 2.0 (Rajpurkar et al., 2018). SQuAD 2.0 has 151,054 questions answers pairs from 505 articles, where 53,775 of the questions are unanswerable. Each question is accompanied with its “answerability” which indicates whether the question is answerable, and the ground truth answers if the answer exists in the context. Following settings in SAN (Liu et al., 2018), results on the official development set are shown.

We measure the standard Exact Match (EM) and F1 scores to evaluate the accuracy. EM is the ratio that predicted answers matches with one of ground truth answers exactly. F1 indicates the average overlap between the prediction and the ground truth.

We compare our knowledge-aware framework with the base model SAN. The results are shown in Figure 2. We show the average EM and F1 scores of four runs. Results over 19 epochs are shown and after that the performances of both methods decrease. It can be observed that incorporating the knowledge graph brings a significant performance improvement compared to the base model, which indicates the importance of external knowledge in question answering.

We analyze the effect of choosing different depths for Graph CNN. As we add residual connections after every 2 Graph CNN layers, 2 layers are treated as one unit. Table 1 shows that 4-layer Graph CNN performs better than a 2-layer framework, which indicates that a deeper graph CNN structure can encode knowledge of sub-graphs better. We have no results with more layers due to the limitation of GPU memory. We also show the effect of choosing different concepts embedding initializations: (1) random initialization; (2) GloVe 100 (Pennington et al., 2014); (3) GloVe 300; (4) NumberBatch (Speer et al., 2016). NumberBatch that is pre-trained on ConceptNet outperforms other embeddings, which confirms the importance of appropriate parameter initialization for vertices in Graph CNN.

5 Conclusion

This paper presents a framework which incorporates external knowledge encoded in knowledge graphs for the question answering task. We utilize Graph Convolutional Neural Networks to encode the knowledge sub-graph built for the passage context, and generate knowledge-aware memory for answer prediction. Our method is general and is compatible with any base method for question answering. We evaluate our approach on the SQuAD 2.0 dataset and show that the knowledge-aware method outperforms the base model, which verifies the importance of background knowledge for language understanding.

Acknowledgements

We are grateful to Xavier Bresson for helpful discussions on Graph CNN.

References

Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Yuan Fang, Kingsley Kuan, Jie Lin, Cheston Tan, and Vijay Chandrasekhar. Object detection meets knowledge graphs. In *IJCAI*, pp. 1661–1667, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pp. 448–456, 2015. URL http://dl.acm.org/citation.cfm?id=3045118.3045167.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. Stochastic answer networks for machine reading comprehension. In *ACL*, pp. 1694–1704, 2018. URL http://aclweb.org/anthology/P18-1157.

Todor Mihaylov and Anette Frank. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *ACL*, pp. 821–832, 2018. URL http://aclweb.org/anthology/P18-1076.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pp. 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In *ACL*, pp. 784–789, 2018. URL http://aclweb.org/anthology/P18-2124.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *ICLR*, 2017.

Ying Shen, Yang Deng, Min Yang, Yaliang Li, Nan Du, Wei Fan, and Kai Lei. Knowledge-aware attentive neural network for ranking question answer pairs. In *SIGIR*, 2018.

Robert Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, 2016.

Qi Wu, Chunhua Shen, Peng Wang, Anthony Dick, and Anton van den Hengel. Image captioning and visual question answering based on attributes and external knowledge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1367–1381, 2018.

Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. *ICLR*, 2017.