Improving Question Answering over Incomplete KBs with Knowledge-Aware Reader

Wenhan Xiong[†], **Mo Yu**[*], **Shiyu Chang**[*], **Xiaoxiao Guo**[*], **William Yang Wang**[†]
[†] University of California, Santa Barbara
[*] IBM Research
{xwhan, william}@cs.ucsb.edu, yum@us.ibm.com, {shiyu.chang, xiaoxiao.guo}@ibm.com

Abstract

We propose a new end-to-end question answering model, which learns to aggregate answer evidence from an incomplete knowledge base (KB) and a set of retrieved text snippets. Under the assumptions that the structured KB is easier to query and the acquired knowledge can help the understanding of unstructured text, our model first accumulates knowledge of entities from a question-related KB subgraph; then reformulates the question in the latent space and reads the texts with the accumulated entity knowledge at hand. The evidence from KB and texts are finally aggregated to predict answers. On the widely-used KBQA benchmark WebQSP, our model achieves consistent improvements across settings with different extents of KB incompleteness.[1]

1 Introduction

Knowledge bases (KBs) are considered as an essential resource for answering factoid questions. However, accurately constructing KB with a well-designed and complicated schema requires lots of human efforts, which inevitably limits the coverage of KBs (Min et al., 2013). As a matter of fact, KBs are often incomplete and insufficient to cover full evidence required by open-domain questions.

On the other hand, the vast amount of unstructured text on the Internet can easily cover a wide range of evolving knowledge, which is commonly used for open-domain question answering (Chen et al., 2017; Wang et al., 2018). Therefore, to improve the coverage of KBs, it is straightforward to augment KB with text data. Recently, text-based QA models along (Seo et al., 2016; Xiong et al., 2017; Yu et al., 2018) have achieved remarkable performance when dealing with a single passage that is guaranteed to include the answer. However, they are still insufficient when multiple documents are presented. We hypothesize this is partially due to the lack of background knowledge while distinguishing relevant information from irrelevant ones (see Figure 1 for a real example).

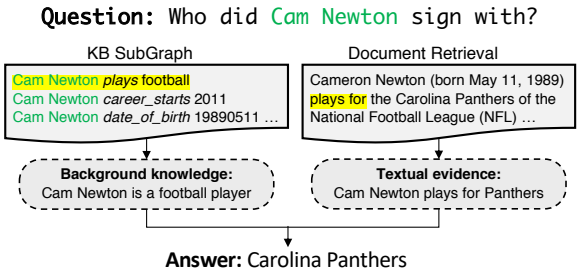

Figure 1: A real example from WebQSP. Here the answer cannot be directly found in the KB. But the knowledge provided by the KB, *i.e.*, *Cam Newton is a football player*, indicates he **signed with** *the team he plays for*. This knowledge can be essential for recognizing the relevant text piece.

To better utilize textual evidence for improving QA over incomplete KBs, this paper presents a new end-to-end model, which consists of (1) a simple yet effective subgraph reader that accumulates knowledge of each KB entity from a question-related KB subgraph; and (2) a knowledge-aware text reader that selectively incorporates the learned KB knowledge about entities with a novel conditional gating mechanism. With the specifically designed gate functions, our model has the ability to dynamically determine how much KB knowledge to incorporate while encoding questions and passages, thus is able to make the structured knowledge more compatible with the text information. Compared to the previous state-of-the-art (Sun et al., 2018), our model achieves consistent improvements with a much more efficient pipeline, which only requires a single pass of the evidence resources.

[1] https://github.com/xwhan/Knowledge-Aware-Reader.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4258–4264
Florence, Italy, July 28 - August 2, 2019.

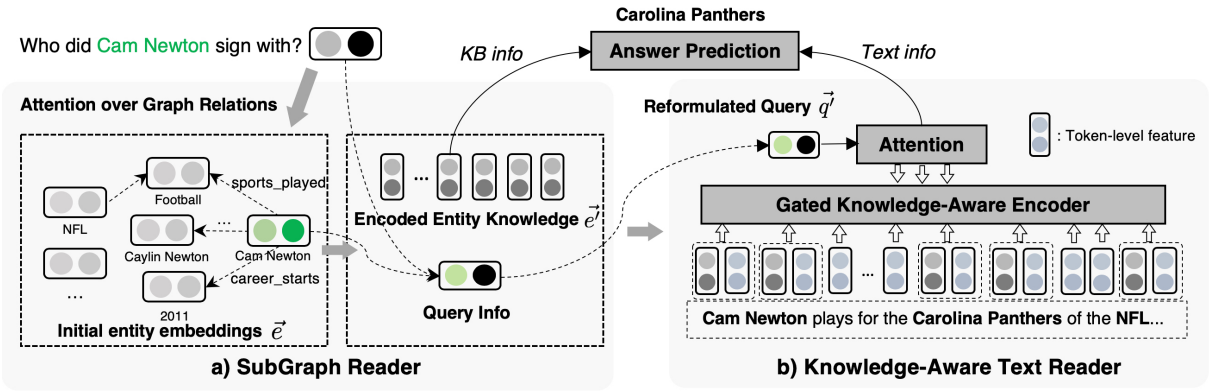

Figure 2: Model Overview. The subgraph reader **a**) first utilizes graph attention networks (Veličković et al., 2017) to collect information for each entity in the question-related subgraph. The learned knowledge of each entity ($\vec{e'}$) is then passed to the text reader **b)** to reformulate the question representation ($\vec{q'}$) and encode the passage in a knowledge-aware manner. Finally, the information from the text and the KB subgraph is aggregated for answer entity prediction.

2 Task Definition

The QA task we consider here requires answering questions by reading knowledge base tuples $\mathcal{K} = \{(e_s, r, e_o)\}$ and retrieved Wikipedia documents $\mathcal{D}$. To build a scalable system, we follow Sun et al. (2018) and only consider a subgraph for each question. The subgraph is retrieved by running Personalized PageRank (Haveliwala, 2002) from the topic entities[2] (entities mentioned by the question: $\mathcal{E}_0 = \{e | e \in Q\}$). The documents $\mathcal{D}$ are retrieved by an existing document retriever (Chen et al., 2017) and further ranked by Lucene index. The entities in documents are also annotated and linked to KB entities. For each question, the model tries to retrieve answer entities from a candidate set including all KB and document entities.

3 Model

The core components of our model consist of a graph-attention based KB reader (§3.1) and a knowledge-aware text reader (§3.2). The interaction between the modules is shown in Figure 2.

3.1 SubGraph Reader

This section describes the KB subgraph reader (SGREADER), which employs graph-attention techniques to accumulate knowledge of each subgraph entity (e) from its linked neighbors (N_e). The graph attention mechanism is particularly designed to take into account two important aspects: (1) whether the neighbor relation is relevant to the question; (2) whether the neighbor entity is a topic entity mentioned by the question. After the propagation, the SGREADER finally outputs a vectorized representation for each entity, encoding the knowledge indicated by its linked neighbors.

Question-Relation Matching To match the question and KB relation in an isomorphic latent space, we apply a shared LSTM to encode the question $\{w_1^q, w_2^q, ..., w_{l_q}^q\}$ and the tokenized relation $\{w_1^r, w_2^r, ..., w_{l_r}^r\}$. With the derived hidden states $\mathbf{h^q} \in \mathbb{R}^{l_q \times d_h}$ and $\mathbf{h^r} \in \mathbb{R}^{l_r \times d_h}$ for each word, we first compute the representation of relations with a self-attentive encoder:

$$\vec{r} = \sum_i \alpha_i \vec{h}_i^r, \ \alpha_i \propto \exp(\vec{w_r} \cdot \vec{h}_i^r),$$

where $\vec{h}_i^r$ is the i-th row of $\mathbf{h^r}$ and $\vec{w_r}$ is a trainable vector. Since a question needs to be matched with different relations and each relation is only described by part of the question, instead of matching the relations with a single question vector, we calculate the matching score in a more fine-grained way. Specifically, we first use $\vec{r}$ to attend each question token and then model the matching s_r by a dot product as follows:

$$s_r = \vec{r} \cdot \sum_j \beta_j \vec{h}_j^q, \beta_j \propto \exp(\vec{r} \cdot \vec{h}_j^q).$$

Extra Attention over Topic Entity Neighbors In addition to the question-relation similarities, we find another binary indicator feature derived from the topic entity is very useful. This indicator is defined as $I[e_i \in \mathcal{E}_0]$ for a neighbor (r_i, e_i) of an arbitrary entity e. Intuitively, if one neighbor links to a topic entity that appear in the question then

[2] Annotated by STAGG (Yih et al., 2014).

the corresponding tuple (e, r_i, e_i) could be more relevant than other non-topic neighbors for question answering. Formally, the final attention score $\tilde{s}_{(r_i,e_i)}$ over each neighbor (r_i, e_i) is defined as:

$$\tilde{s}_{(r_i,e_i)} \propto \exp(I[e_i \in \mathcal{E}_0] + s_{r_i}).$$

Information Propagation from Neighbors To accumulate the knowledge from the linked tuples, we define the propagation rule for each entity e:

$$\vec{e'} = \gamma^e \vec{e} + (1-\gamma^e) \sum_{(e_i,r_i)\in N_e} \tilde{s}_{(r_i,e_i)} \sigma(\mathbf{W_e}[\vec{r_i}; \vec{e_i}]),$$

where $\vec{e}$ and $\vec{e_i}$ are pre-computed knowledge graph embeddings, $\mathbf{W_e} \in \mathbb{R}^{h_d \times 2h_d}$ is a trainable transformation matrix and $\sigma(\cdot)$ is an activation function. In addition, γ_e is a trade-off parameter calculated by a linear gate function as $\gamma^e = g(\vec{e}, \sum_{(e_i,r_i)\in N_e} \tilde{s}_{(r_i,e_i)} \sigma(\mathbf{W_e}[\vec{r_i}; \vec{e_i}]))$[3] , which controls how much information in the original entity representation should be retained.[4]

3.2 Knowledge-Aware Text Reader

With the learned KB embeddings, our model enhances text reading with KAReader. Briefly, we use an existing reading comprehension model (Chen et al., 2017) and improve it by learning more knowledge-aware representations for both question and documents.

Query Reformulation in Latent Space First, we update the question representation in a way that the KB knowledge of the topic entity can be incorporated. This allows the reader to discriminate relevant information beyond text matching.

Formally, we first take the original question encoding $\mathbf{h^q}$ and apply a self-attentive encoder to get a stand-alone question representation: $\vec{q} = \sum_i b_i \vec{h_i^q}$. We collect the topic entity knowledge of the question by $\vec{e^q} = \sum_{e\in\mathcal{E}_0} \vec{e'}/|\mathcal{E}_0|$. Then we apply a gating mechanism to fuse the original question representation and the KB knowledge:

$$\vec{q'} = \gamma^q \vec{q} + (1-\gamma^q) \tanh(\mathbf{W^q}[\vec{q}, \vec{e^q}, \vec{q} - \vec{e^q}]),$$

where $\mathbf{W^q} \in \mathbb{R}^{h_d \times 3h_d}$, and $\gamma^q = \text{sigmoid}(\mathbf{W^{gq}}[\vec{q}, \vec{e^q}, \vec{q} - \vec{e^q}])$ is a linear gate.

[3] $g(x, y) = \text{sigmoid}(\mathbf{W}[x; y]) \in (0, 1)$.

[4] The above step can be viewed as a gated version of the graph encoding techniques in NLP, e.g., (Song et al., 2018; Xu et al., 2018). These general graph-encoders and graph-attention techniques may help when the questions require more hops and we leave the investigation to future work.

Knowledge-aware Passage Enhancement To encode the retrieved passages, we use a standard bi-LSTM, which takes several token-level features[5]. With the entity linking annotations in passages, we fuse the entity knowledge with the token-level features in a similar fashion as the query reformulation process. However, instead of applying a standard gating mechanism (Yang and Mitchell, 2017; Mihaylov and Frank, 2018), we propose a new conditional gating function that explicitly conditions on the question $\vec{q'}$. This simple modification allows the reader to dynamically select the inputs according to their relevance to the question. Considering a passage token w_i^d with its token features $\vec{f}_{w_i}^d$ and its linked entity e_{w_i}[6], we define the conditional gating function as:

$$\vec{i}_{w_i}^d = \gamma^d \vec{e'}_{w_i} + (1-\gamma^d) \vec{f}_{w_i}^d, \text{ where}$$
$$\gamma^d = \text{sigmoid}(\mathbf{W^{gd}}[\vec{q} \cdot \vec{e'}_{w_i}; \vec{q} \cdot \vec{f}_{w_i}^d]).$$

$\vec{e'}_{w_i}$ denotes the entity embedding learned by our SGReader.

Entity Info Aggregation from Text Reading Finally we feed the knowledge-augmented inputs $\vec{i}_{w_i}^d$ into the biLSTM and use the output token-level hidden state $\vec{h}_{w_i}^d$ to calculate the attention scores $\lambda_i = \vec{q'}^T \vec{h}_{w_i}^d$. Afterwards, we get each document's representation as $\vec{d} = \sum_i \lambda_i \vec{h}_{w_i}^d$. For a certain entity e and all the documents containing e: $\mathcal{D}^e = \{d | e \in d\}$, we simply aggregate the information by averaging the representations of linked documents as $\vec{e}_d = \frac{1}{|\mathcal{D}^e|} \sum_{d\in\mathcal{D}^e} \vec{d}$.

3.3 Answer Prediction

With entities representations ($\vec{e'}$ and $\vec{e^d}$), we predict the probability of an entity being the answer by matching the query vectors and the entity representations: $s^e = \sigma_s(\vec{q'}^T \mathbf{W_s}[\vec{e'}; \vec{e^d}])$.

4 Experiment

4.1 Setup

Dataset Our experiments are based on the WebQSP dataset (Yih et al., 2016). To simulate the real-world scenarios, we test our models following the settings of (Sun et al., 2018), where the KB is

[5] We use the same set of features as in (Chen et al., 2017) except for the tagging labels.

[6] Non-entity tokens are encoded with token-level features only.

Model	10% KB		30% KB		50% KB		100% KB	
	Hit@1	F1	Hit@1	F1	Hit@1	F1	Hit@1	F1
KV-KB	12.5	4.3	25.8	13.8	33.3	21.3	46.7	38.6
GN-KB	15.5	6.5	34.9	**20.4**	47.7	**34.3**	**66.7**	**62.4**
SGREADER (Ours)	**17.1**	**7.0**	**35.9**	20.2	**49.2**	33.5	66.5	58.0
KV-KB+TEXT	24.6	14.4	27.0	17.7	32.5	23.6	40.5	30.9
GN-LF	29.8	17.0	39.1	25.9	46.2	35.6	65.4	56.8
GN-EF	31.5	17.7	40.7	25.2	49.9	34.7	67.8	60.4
SGREADER + KAREADER (Ours)	**33.6**	**18.9**	**42.6**	**27.1**	**52.7**	**36.1**	67.2	57.3
GN-LF+EF (ensemble)	33.3	19.3	42.5	26.7	52.3	37.4	68.7	62.3

Table 1: Comparisons with Key-Value Memory Networks and GRAFT-Nets under different KB settings.

downsampled to different extents. For a fair comparison, the retrieved document set is the same as the previous work.

Baselines and Evaluation Key-Value (KV) Memory Network (Miller et al., 2016) is a simple baseline that treats KB triples and documents as memory cells. Specifically, we consider its two variants, **KV-KB** and **KV-KB+Text**. The former is a KB-only model while the latter uses both KB and text. We also compare to the latest method **GraftNet** (GN) (Sun et al., 2018), which treats documents as a special genre of nodes in KBs and utilizes graph convolution (Kipf and Welling, 2016) to aggregate the information. Similar to the KV-based baselines, we denote **GN-KB** as the KB-only version. Further, both **GN-LF** (late fusion) and **GN-EF** (early fusion) consider both KB and text. The former one considers KB and texts as two separate graphs, and then ensembles the answer scores. GN-EF is the existing best single model, which considers KB and texts as a single heterogeneous graph and aggregate the evidence to predict a single answer score for each entity. F1 and His@1 are used for evaluation since multiple correct answers are possible.

The implementation details of our model can be found in the Appendix.

4.2 Results and Analysis

We show the main results of different incomplete KB settings in Table 1. For reference, we also show the results under full KB settings (*i.e.*, 100%, all of the required evidence is covered by KB). The row of SGREADER shows the results of our model using only KB evidence. Compared to the previous KBQA methods (KV-KB and GN-KB), SGREADER achieves better results in incomplete KB settings and competitive performance with the full KB. Here we do not compare with existing methods that utilize semantic parsing annotations (Yih et al., 2016; Yu et al., 2017). It is worth noting that SGREADER only needs one hop of graph propagation while the compared methods typically require multiple hops.

Model	Hit@1	F1
Full Model	46.8	28.1
- w/o query reformulation	44.4	27.6
- w/o knowledge enhancement	45.2	27.0
- w/o conditional knowledge gate	44.4	27.0

Table 2: Ablation on dev under the 30% KB setting.

Augmenting the SGREADER with our knowledge-aware reader (KAREADER) results in consistent improvements in the settings with incomplete KBs. Compared to other baselines, although our model is built upon a stronger KB-QA base model, it achieves the largest absolute improvement. It is worth mentioning that our model is still a single model, but it achieves competitive results to the existing ensemble model (GN-LF+EF). The results demonstrate the advantage of our knowledge-aware text reader.

Ablation Study To study the effect of each KAREADER component, we conduct ablation analysis under the 30% KB setting (Table 2). We see that both query reformulation and knowledge enhancement are essential to the performance. Additionally, we find the conditional gating mechanism proposed in §3.2 is important. When replacing it with a standard gate function (see the row *w/o conditional knowledge gate*), the performance is even lower than the reader without knowledge enhancement, suggesting our proposed new gate function is crucial for the success of knowledge-aware text reading. The potential reason is that without the question information, the gating mechanism might introduce some irrelevant and misleading knowledge.

1)	Question: Which airport to fly into Rome? Groundtruth: **Leonardo da Vinci-Fiumicino Airport (fb:m.01ky5r), Ciampino-G. B. Pastine International Airport (fb:m.033_52)** SGREADER: Italian Met Office Airport (fb:m.04fngkc) SGREADER + KAREADER: **Leonardo da Vinci-Fiumicino Airport (fb:m.01ky5r)** Missing knowledge of the incomplete KB: No airport info about Rome.
1)	Question: Where did George Herbert Walker Bush go to college? Groundtruth: **Yale (fb:m.08815)** SGREADER: United States of America (fb:m.09c7w0) SGREADER + KAREADER: **Yale (fb:m.08815)** Missing knowledge of the incomplete KB: No college info about George Herbert Walker Bush.
2)	Question: When did Juventus win the champions league? Groundtruth: **1996 UEFA Champions League Final (fb:m.02pt_57)** SGREADER: 1996 UEFA Super Cup (fb:m.02rw0yt) SGREADER + KAREADER: **1996 UEFA Champions League Final (fb:m.02pt_57)** Missing knowledge of the incomplete KB: UEFA Super Cup is not UEFA Champions League Final (fb:m.05nblxt)
2)	Question: What college did Albert Einstein go to? Groundtruth: **ETH Zurich (fb:m.01dyk8), University of Zurich (fb:m.01tpvt)** SGREADER: Sri Krishnaswamy matriculation higher secondary school (fb:m.0127vh33) SGREADER + KAREADER: **ETH Zurich (fb:m.01dyk8)** Missing knowledge of the incomplete KB: the answer should be a college (fb:m.01y2hnl)
3)	Question: When is the last time the Denver Broncos won the Superbowl? Groundtruth: **Super Bowl XXXIII (fb:m.076y0)** SGREADER: **Super Bowl XXXIII (fb:m.076y0)** SGREADER + KAREADER: 1999 AFC Championship game (fb:m.0100z7bp)
3)	Question: What was Lebron James first team? Groundtruth: **Cleveland Cavaliers (fb:m.0jm7n)** SGREADER: **Cleveland Cavaliers (fb:m.0jm7n)** SGREADER + KAREADER: Toronto Raptors (fb:m.0jmcb)

Table 3: Human analysis on test samples in the 30% KB settingo. 1) and 2) show some typical examples of the case (83.2% of all test samples) where the KAREADER improves upon our SGREADER. 3) shows some examples where using KB alone is better than using both KB and Text (16.8%). The Freebase IDs of the entities are also included for reference.

Qualitative Analysis In Table 3, there are two major categories of questions that can be better answered using our full model. In the first category, indicated by 1), the answer fact is missing in the KB, mainly because there are no links from the question entities to the answer entity. In these cases, the SGREADER sometimes can predict an answer with a correct type, but the answers are mostly irrelevant to the question.

The second category, denoted as 2), indicates examples where the KB provides relevant information but does not cover some of the constraints on answers' properties (*e.g.*, answers' entity types). In the two examples shown above, we can see that SGREADER is able to give some reasonable answers but the answers do not satisfy the constraints indicated by the question.

Finally, when the KB is sufficient to answer a question, there are some cases where the KAREADER introduces wrong answers into the top-ranked answer list. We list two examples at the bottom of the Table 3. These newly included incorrect answers are usually relevant to the original questions but come from the noises in machine reading. These cases suggest that our concatenation-based knowledge aggregation still has some room for improvement, which we leave for future work.

5 Conclusion

We present a new QA model that operates over incomplete KB and text documents to answer open-domain questions, which yields consistent improvements over previous methods on the WebQSP benchmark with incomplete KBs. The results show that (1) with the graph attention technique, we can efficiently and accurately accumulate question-related knowledge for each KB entity in one-pass of the KB sub-graph; (2) our designed gating mechanisms could successfully incorporate the encoded entity knowledge while processing the text documents. In future work, we will extend the proposed idea to other QA tasks with evidence of multimodality, e.g. combining with symbolic approaches for visual QA (Gan et al., 2017; Mao et al., 2019; Hu et al., 2019).

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1870–1879.

Chuang Gan, Yandong Li, Haoxiang Li, Chen Sun, and Boqing Gong. 2017. Vqs: Linking segmentations to questions and answers for supervised attention in vqa and question-focused semantic segmentation. In *ICCV*, pages 1811–1820.

Taher H Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM.

Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. 2019. Language-conditioned graph networks for relational reasoning. *arXiv preprint arXiv:1905.04405*.

Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *ACL 2018*, pages 821–832.

Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1400–1409.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for amr-to-text generation. *arXiv preprint arXiv:1805.02473*.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W. Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4231–4242. Association for Computational Linguistics.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018. R 3: Reinforced ranker-reader for open-domain question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Caiming Xiong, Victor Zhong, and Richard Socher. 2017. DCN+: mixed objective and deep residual coattention for question answering. *CoRR*, abs/1711.00106.

Kun Xu, Lingfei Wu, Zhiguo Wang, and Vadim Sheinin. 2018. Graph2seq: Graph to sequence learning with attention-based neural networks. *arXiv preprint arXiv:1804.00823*.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1436–1446.

Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *ACL 2014*, volume 2, pages 643–648.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 201–206.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *CoRR*, abs/1804.09541.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 571–581.

Implementation Details Throughout our experiments, we use the 300-dimension GloVe embeddings trained on the Common Crawl corpus. The hidden dimension of LSTM and the dimension of entity embeddings are both 100. We use the same pre-trained entity embeddings as used by Sun et al. (2018). For graph attention over the KB subgraph, we limit the max number of neighbors for each entity to be 50. We use the norm for gradient clipping as 1.0. We apply dropout=0.2 on both word embeddings and LSTM hidden states. The max question length is set to 10 and the max document length is set to 50. For optimization, we apply label smoothing with a factor of 0.1 on the binary cross-entropy loss. During training, we use the Adam with a learning rate of 0.001.