

# Improving open domain question answering with Knowledge Base and Wikipedia graph

Ruiyu Lin  
SUN YAT-SEN UNIVERSITY  
linry23@mail2.sysu.edu.cn

## ABSTRACT

A clear and well-documented L<sup>A</sup>T<sub>E</sub>X document is presented as an

## KEYWORDS

neural networks

## ACM Reference Format:

Ruiyu Lin. 2018. Improving open domain question answering with Knowledge Base and Wikipedia graph. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/1122445.1122456>

## 1 INTRODUCTION

Open-domain Question Answering mostly focus on factoid question answering, which require systems to return a short and concise answer to these questions. Most existing models, however, answer questions using a single information source, usually either text from an text corpus such as Wikipedia[4], or a single knowledge base (KB).

Large-scale factual knowledge bases such as WikiData[12], Freebase [3], Dbpedia[2], stores a large number of facts in an organized way. Namely, Freebase has 46m entities and 2.6b facts, WikiData contains 87m items. Each fact is made of two entities and a relation between them. Most knowledge bases are curated, ensuring the correctness of the information, common or "simple" questions can be answered easily if semantic parsing (question query) is done correctly. The advantage of graph structure also enables multi-hop question answering. Unfortunately, curated knowledge bases, which demands tremendous human labor, might not keep up with times, thereby some relations would be missing. Limited coverage of questions can be answered because the reasoning is based on the similarity over relationships and entities.

Wikipedia[4], a text source, was proposed for the first time to process Open Domain QA tasks, and a DRQA system was developed, including Document Retriever and Document Reader, which laid the pipe-line, two-stage approaches, of QA for successive work. We also follow this tradition, retrieve and then read. Text corpus provides a more completed coverage of facts, and it is easier to catch the time, however lacks the ability of multi-hop reasoning.

To combine the coverage of text evidence and reasoning ability of knowledge base, some recent work use both text and KB, to constructs graphs of nodes and edges[9, 10, 13]. These works basically augment the KB graph with the entities identified from the relevant text evidences, the task of answer determination is then reduced to classify the entity node is the answer or not. Another line[5, 8], inversely, augment retrieval passage with KB graph, and the task of answer determination is to do answer extraction from text.

## 2 RELATED WORK

### 2.1 QA using Text

### 2.2 QA using KB

### 2.3 QA using both Text and KB

In [9, 10, 13], the answers is restricted to be the KB entities. Meanwhile, their question subgraph is heterogeneous, which contains KB triples(entity, relation, entity), and entity-linked text. The task of QA then reduces to learning the representations of the nodes, and then performing a binary classification over these nodes to decide whether it is the answer or not. They both augmented knowledge bases with text from Wikipedia, which means KB dominates the whole process.

However, [8] construct the graph in a different way. Inversely, the knowledge base is used to better model relationships between different passages of text, which means the text corpus dominates instead. Its question subgraph is not heterogeneous, only contains entity-linked passage and relations. The task of QA switches to learning the representations of the passage. Not to classify the node is the answer or not, it extracts the most possible span as answer in the most possible passage as prior work did. Our work is consistent with it.

## 3 METHOD

Specifically, we use Wikipedia graph[1] as the text corpus  $\mathcal{G}$  and Wikidata[12] as the knowledge base  $\mathcal{K}$ , as there exists an alignment between the two resources. Knowledge Base is a multi-relational graphs, each edge has a label and direction associated with it, and each node in the graph is an entity. Wikipedia graph[1] is a directed graph, constructed by hyperlinks and within-document links, each node represents a single paragraph. The Wikipedia graph is densely connected and covers a wide range of topics that provide useful evidence for open-domain questions. Our goal is to fuse Knowledge Base knowledge into Wikipedia graph passages, learn the KB-aware representation of all the retrieved passage, as the input to a reader model to extract answer.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

Woodstock '18, June 03–05, 2018, Woodstock, NY

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/10.1145/1122445.1122456>

**Seed Nodes** Given a natural language question  $q$ , we use the top  $K_{TF-IDF}$  paragraphs returned by a TF-IDF based retrieval system as the start nodes.

**Subgraph** Next, we run Topic-sensitive pagerank[7] around these seed nodes to get the edges weight over their neighbouring, denoted as  $\mathcal{G}_q = (\mathcal{P}, \mathcal{E})$ , where  $\mathcal{P}$  is the set of paragraphs  $\{p_1, \dots, p_{|p|}\}$  in the Wikipedia graph, and the edge  $\mathcal{E}$  are links between them with a pagerank score  $pr$  over them. Each paragraphs is a sequence of words  $p_i = (w_1^{p_i}, \dots, w_{|p_i|}^{p_i})$

**Node Initialization** Note that we use the same encoder BERT during the whole process.

(1) Question initialization

Given a natural language question  $q = (w_1^q, \dots, w_q^q)$ , initialize question as:

$$h_q^{(0)} = BERT(w_1^q, \dots, w_q^q)$$

We perform an entity linking system[6] to identify the *topic entity* in the question, denoted as  $E_{topic} = \{e_1, \dots, e_{|topic|}\}$ .

(2) Passage initialization

And the passages in  $\mathcal{G}_q$  are initialized as:

$$h_{p_i}^{(0)} = BERT(w_1^{p_i}, \dots, w_{|p_i|}^{p_i})$$

We denote the embedding of *pos-th* word in the passage  $p_i$  as  $h_{p_i}^{(l)}[pos]$ .

**Node Update** Assuming that  $(p_a, p_b) \in \mathcal{G}_q$ , and the edge weight is  $pr_{a,b}$ , we also perform an entity linking system[6] to identify the entity in  $p_a$  and  $p_b$ . Suppose that  $p_a$  contains  $n$  entities  $(e_{a1}, \dots, e_{an})$ ,  $p_b$  contains  $m$  entities  $(e_{b1}, \dots, e_{bm})$ . In each layer  $l$ , we update the information as follow:

(1) Corporate entity relations into passage .

We update passage representation at token level. If the triple  $(e_{ai}, r, e_{bj}) \in \mathcal{K}$ , and  $e_{ai}$  is at the position  $pos_a$  in  $p_a$ ,  $e_{bj}$  is at  $pos_b$  in  $p_b$ , let  $L = (p_a, pos_a)$  be the set of all the entities appearing at position  $pos_a$  in paragraph  $p_a$ , they are all linked to the entities at position  $pos_b$  in paragraph  $p_b$ . We aggregate over the token hidden states coming in at each position separately:

$$h_{p_b}^{(l)}[pos_b] = FFN \left( \sum_{(p_a, pos_a) \in L} pr_{a,b} * s(h_r, h_q^{(l)}) * \phi(h_{p_a}^{(l-1)}[pos_a], h_r) \right)$$

$s(h_r, h_q^{(l)})$  is the relevant score between relation and question, we simply computes it through the dot product between their hidden state.  $h_r$  is embedding of relations, which are pretrained and can be looked up from an embedding table.

$$s(h_r, h_q^{(l)}) = h_r h_q^{(l)}$$

$\phi$  is a transform operation, which is

$$\phi(h, r) = FFN(h; r)$$

(2) Update passage representation

Next we aggregate states within the paragraph using BERT again:

$$h_p^{(l)} = BERT(h_p^{(l)}[1], \dots, h_p^{(l)}[|p|])$$

(3) Update topic entities representation

We have  $E_{topic} = \{e_1, \dots, e_{|topic|}\}$  from above step. Let  $Q = (e, p, pos)$  denotes the topic entity  $e$  appear at the position  $pos$  in passage  $p$ , we update the topic entity representation by aggregate the information from the updated passages.

$$h_e^{(l)} = \frac{1}{|Q|} \sum_{(e, p, pos) \in Q} h_p^{(l)}[pos]$$

(4) Update question representation

Next we aggregate states within the question using BERT again:

$$h_q^{(l)} = BERT(h_q^{(l)}[1], \dots, h_q^{(l)}[pos])$$

*s.t.*

$$h_q^{(l)}[i] = \begin{cases} h_e^{(l)}, & \Phi(e, i) = 1 \\ h_q^{(l-1)}[i], & \Phi(e, i) = 0 \end{cases}$$

$\Phi(e, i) = 1$  means the token at position  $i$  in question is a topic entity.

**Answer Extraction**

Denote the passage score as  $Pr(P_i|Q, P)$ , which reranks all retrieved passages

$$Pr(P_i|Q, P) = \text{softmax}(W[h_{p_i}^T; h_q])$$

$W$  is a trainable parameter.

The score of an answer span from passage  $P_i$  will be

$$Pr(a|Q, P) = Pr(P_i|Q, P)P_s(a_s|Q, P)P_e(a_e|Q, P).$$

## REFERENCES

- [1] Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. ICLR 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. (ICLR 2020).
- [2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer, 722–735.
- [3] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 1247–1250.
- [4] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051* (2017).
- [5] Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. ACL 2019. Cognitive graph for multi-hop reading comprehension at scale. (ACL 2019).
- [6] Paolo Ferragina and Ugo Scaiella. 2011. Fast and accurate annotation of short texts with wikipedia pages. *IEEE software* 29, 1 (2011), 70–75.
- [7] Taher H Haveliwal. 2003. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE transactions on knowledge and data engineering* 15, 4 (2003), 784–796.
- [8] Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Knowledge guided text retrieval and reading for open domain question answering. *arXiv preprint arXiv:1911.03868* (2019).
- [9] Haitian Sun, Tania Bedrax-Weiss, and William W Cohen. 2019. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2380–2390, 7.
- [10] 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*, pages 4231–4242.
- [11] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar. ICLR 2020. Composition-based Multi-Relational Graph Convolutional Networks. (ICLR 2020).
- [12] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

- [13] Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Improving question answering over incomplete kbs with knowledge-aware reader. *arXiv preprint arXiv:1905.07098* (2019).