BAG: Bi-directional Attention Entity Graph Convolutional Network for Multi-hop Reasoning Question Answering

Yu Cao[1] **Meng Fang**[2] **Dacheng Tao**[1]
[1]UBTECH Sydney AI Center, School of Computer Science, FEIT, The University of Sydney
[2]Tencent Robotics X, China
ycao8647@uni.sydney.edu.au, mfang@tencent.com,
dacheng.tao@sydney.edu.au

Abstract

Multi-hop reasoning question answering requires deep comprehension of relationships between various documents and queries. We propose a Bi-directional Attention Entity Graph Convolutional Network (BAG), leveraging relationships between nodes in an entity graph and attention information between a query and the entity graph, to solve this task. Graph convolutional networks are used to obtain a relation-aware representation of nodes for entity graphs built from documents with multi-level features. Bidirectional attention is then applied on graphs and queries to generate a query-aware nodes representation, which will be used for the final prediction. Experimental evaluation shows BAG achieves state-of-the-art accuracy performance on the QAngaroo WIKIHOP dataset.

1 Introduction

Question Answering (QA) and Machine Comprehension (MC) tasks have drawn significant attention during the past years. The proposal of large-scale single-document-based QA/MC datasets, such as SQuAD (Rajpurkar et al., 2016), CNN/Daily mail (Hermann et al., 2015), makes training available for end-to-end deep neural models, such as BiDAF (Seo et al., 2016), DCN (Xiong et al., 2016) and SAN (Liu et al., 2017). However, gaps still exist between these datasets and real-world applications. For example, reasoning is constrained to a single paragraph, or even part of it. Extended work was done to meet practical demand, such as DrQA (Chen et al., 2017) answering a SQuAD question based on the whole Wikipedia instead of single paragraph. Besides, latest large-scale datasets, e.g. TriviaQA (Joshi et al., 2017) and NarrativeQA (Kočiskỳ et al., 2018), address this limitation by introducing multiple documents, ensuring reasoning cannot be done within local information. Although those datasets are fairly challenging, reasoning are within one document.

In many scenarios, we need to comprehend the relationships of entities across documents before answering questions. Therefore, reading comprehension tasks with multiple hops were proposed to make it available for machine to tackle such problems, e.g. QAngaroo task (Welbl et al., 2018). Each sample in QAngaroo contains multiple supporting documents, and the goal is selecting the correct answer from a set of candidates for a query. Most queries cannot be answered depending on a single document, and multi-step reasoning chains across documents are needed. Therefore, it is possible that understanding a part of paragraphs loses effectiveness for multi-hop inference, which posts a huge challenge for previous models. Some baseline models, e.g. BiDAF (Seo et al., 2016) and FastQA (Weissenborn et al., 2017), which are popular for single-document QA, suffer dramatical accuracy decline in this task.

In this paper, we propose a new graph-based QA model, named Bi-directional Attention Entity Graph convolutional network (BAG). Documents are transformed into a graph in which nodes are entities and edges are relationships between them. The graph is then imported into graph convolutional networks (GCNs) to learn relation-aware representation of nodes. Furthermore, we introduce a new bi-directional attention between the graph and a query with multi-level features to derive the mutual information for final prediction.

Experimental results demonstrate that BAG achieves state-of-the-art performance on the WIKIHOP dataset. Ablation test also shows BAG benefits from the bi-directional attention, multi-level features and graph convolutional networks.

Our contributions can be summarized as:

- Applying a bi-directional attention between graphs and queries to learn query-aware representation for reading comprehension.
- Multi-level features are involved to gain comprehensive relationship representation for graph nodes during processing of GCNs.

Proceedings of NAACL-HLT 2019, pages 357–362
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

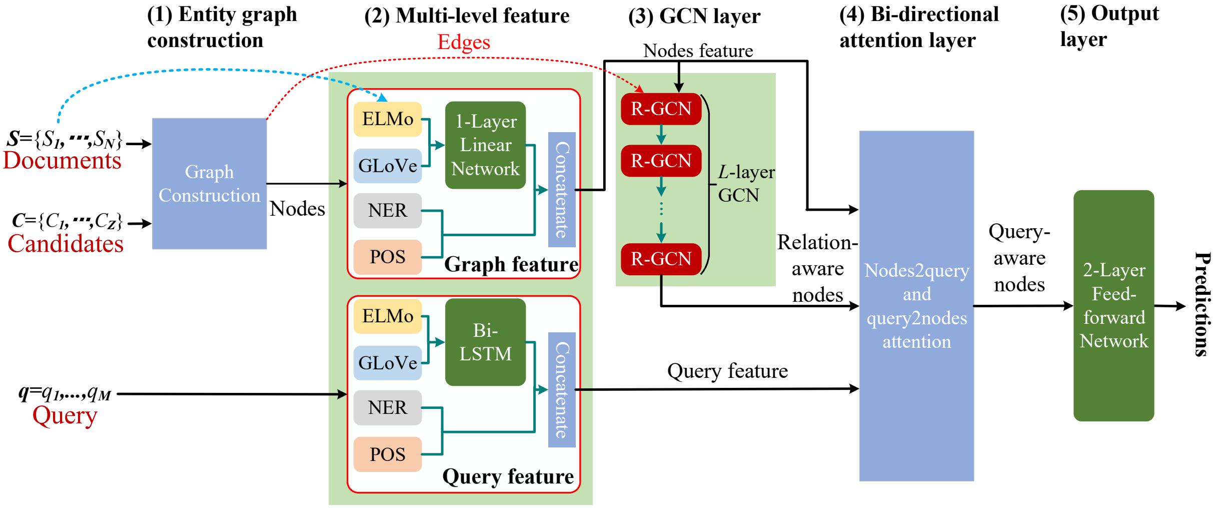

Figure 1: Framework of BAG model.

2 Related Work

Recently coreference and graph-based models are studied for multi-hop QA (Dhingra et al., 2018; Santoro et al., 2017). Coref-GRU (Dhingra et al., 2018) uses coreferences among tokens in documents. However, it is still limited by the long-distance relation propagation capability of RNNs. Besides, graph is proved to be an efficient way to represent complex relationships among objects and derive relational information (Santoro et al., 2017). MHQA-GRN (Song et al., 2018) and Entity-GCN (De Cao et al., 2018) construct entity graphs based on documents to learn more compact representation for multi-hop reasoning and derive answers from graph networks. However, both of them care less about input features and the attention between queries and graph nodes.

Attention has been proven to be an essential mechanism to promote the performance of NLP tasks in previous work (Bahdanau et al., 2014; Sukhbaatar et al., 2015). In addition, bi-directional attention (Seo et al., 2016) shows its superiority to vanilla mutual attention because it provides complementary information to each other for both contexts and queries. However, little work exploits the attention between graphs and queries.

3 BAG Model

We first formally define the multiple-hop QA task, taking QAngaroo (Welbl et al., 2018) WIKIHOP data as an example, There is a set $\boldsymbol{S}$ containing N supporting documents, a query $\boldsymbol{q}$ with M tokens and a set of answer candidates $\boldsymbol{C}$. Our goal is to find the correct answer index a. Giving a triple-style query $\boldsymbol{q} = (\textit{country, kepahiang})$, it means *which country does kepahiang belongs to*. Then answer candidates are provided, e.g. $C = \{Indonesia, Malaysia\}$. There are multiple supporting documents but not all of them are related to reasoning, e.g. *Kephiang is a regency in Bengkulu*, *Bengkulu is one of provinces of Indonesia*, *Jambi is a province of Indonesia*. We can derive the correct candidate is *Indonesia*, i.e. $a = 0$, based on reasoning hops in former two documents.

We show the proposed BAG model in Figure 1. It contains five modules: (1) entity graph construction, (2) multi-level feature layer, (3) GCN layer, (4) bi-directional attention and (5) output layer.

3.1 Entity Graph Construction

We construct an entity graph based on Entity-GCN (De Cao et al., 2018), which means all mentions of candidates found out in documents are used as nodes in the graph. Undirected edges are defined according to positional properties of every node pair. There are two kinds of edges included: 1) cross-document edge, for every node pair with the same entity string located in different documents; 2) within-document edge, for every node pair located in the same document.

Nodes in an entity graph can be found out via simple string matching. This approach can simplify calculation as well as make sure all relevant entities are included in the graph. Picked out along possible reasoning chains during dataset generating (Welbl et al., 2018), answer candidates have contained all related entities for answering. Finally, We can obtain a set of T nodes $\{n_i\}, 1 \leq i \leq T$ and corresponding edges among these nodes via above procedures.

3.2 Multi-level Features

We represent both nodes and queries using multi-level features as shown in Figure 1(2). We first use pretrained word embeddings to represent tokens, such as GLoVe (Pennington et al., 2014) because nodes and queries are composed of tokens. Then contextual-level feature is used to offset the deficiency of GLoVe. Note that only part of tokens are remained during graph construction because we only extract entities as nodes. Thus contextual information around these entities in original document becomes essential for indicating relations between tokens and we use higher-level information for nodes except for token-level feature.

We use ELMo (Peters et al., 2018) as contextualized word representations, modeling both complex word characteristics and contextual linguistic conditions. It should be noted that ELMo features for nodes are calculated based on original documents, then truncated according to the position indices of nodes. Token-level and context-level features will be concatenated and encoded to make a further comprehension. Since a node may contain more than one token, we average features among tokens to generate a feature vector for each node before encoding it. It will be transformed into the encoded node feature via a 1-layer linear network.

Different from nodes, we represent a query by directly using a bidirectional LSTM (Bi-LSTM) whose output in each step is used as encoded query features. And both linear network and LSTM have the same output dimension $\hat{d}$.

In addition, we add two manual features to reflect the semantic properties of tokens, which are named-entity recognition (NER) and part-of-speech (POS). The complete feature $f_n \in \mathbb{R}^{T\times d}$, $f_q \in \mathbb{R}^{M\times d}$ for both nodes and queries will be the concatenation of corresponding encoded features, NER embedding and POS embedding, where $d = \hat{d} + d_{POS} + d_{NER}$.

3.3 GCN Layer

In order to realize multi-hop reasoning, we use a Relational Graph Convolutional Network (R-GCN) (Schlichtkrull et al., 2018) that can propagate message across different entity nodes in graphs and generate transformed representation of original ones. The R-GCN is employed to handle high-relational data characteristics and make use of different edge types. At lth layer, given the hidden state $h_i^l \in \mathbb{R}^d$ of node i, the hidden states $h_j^l \in \mathbb{R}^d, j \in \{N_i\}$ and relations R_{N_i} of all its neighbors (d is the hidden state dimension), the hidden state in the next layer can be obtained via

$$h_i^{l+1} = \sigma(\sum_{r\in R_{N_i}} \sum_{j\in N_i} \frac{1}{c_{i,r}} W_r^l h_j^l + W_0^l h_i^l), \tag{1}$$

where $c_{i,r}$ is a normalization constant $|N_i|$, $W_r^l \in \mathbb{R}^{d\times d}$ stands a relation-specific weight matrix and $W_0^l \in \mathbb{R}^{d\times d}$ stands a general weight.

Similar to Entity-GCN (De Cao et al., 2018), we apply a gate on update vector u_i^l and hidden state h_i^l of current node by a linear transformation f_s,

$$w_i^l = \sigma(f_s(\text{concat}(u_i^l, h_i^l))), \tag{2}$$

in which u_i^l can be obtained via (1) without sigmoid function. Then it will be used for updating weights for the hidden state h_i^{l+1} of the same node in next layer,

$$h_i^{l+1} = w_i^l \odot \tanh(u_i^l) + (1 - w_i^l) \odot h_i^l. \tag{3}$$

We stack such networks for L layers in which all parameters are shared. The information of each node will be propagated up to L-node distance away, generating L-hop-reasoning relation-aware representation of nodes. The initial input will be mutli-level nodes features $\mathbf{f_n} = \{f_{n_i}\}, 0 \le i \le T$ and edges $\mathbf{e} = \{e_{ij}\}$ in the graph.

3.4 Bi-directional Attention Between a Graph and a Query

Bi-directional attention is responsible for generating the mutual information between a graph and a query. In BiDAF (Seo et al., 2016), attention is applied to sequence data in QA tasks such as supporting texts. However, we also find it works well between graph nodes and queries. It generates query-aware node representations that can provide more reasoning information for prediction.

What differs in BAG is that attention is applied for graphs as shown in Figure 1(4). The similarity matrix $\mathbf{S} \in \mathbb{R}^{\mathbf{T}\times\mathbf{M}}$ is calculated via

$$\mathbf{S}= \text{avg}_{-1} f_a(\text{concat}(h_n, f_q, h_n \circ f_q)), \tag{4}$$

in which $h_n \in \mathbb{R}^{T\times d}$ is all node representations obtained from the last GCN layer, $f_q \in \mathbb{R}^{M\times d}$ is the query feature matrix after encoding, d is the dimension for both query feature and transformed node representation, f_a is a linear transformation, avg_{-1} stands for the average operation in last dimension, and $\circ$ is element-wise multiplication.

Unlike the context-to-query attention in BiDAF, we introduce a node-to-query attention $\widetilde{a}_{n2q} \in \mathbb{R}^{T\times d}$, which signifies the query tokens that have the highest relevancy for each node using

$$\widetilde{a}_{n2q} = \mathrm{softmax}_{\mathrm{col}}(\mathbf{S}) \cdot f_q, \tag{5}$$

where $\mathrm{softmax}_{\mathrm{col}}$ means performing softmax function across the column, and $\cdot$ stands for matrix multiplication.

At the same time, we also design query-to-node attention $\widetilde{a}_{q2n} \in \mathbb{R}^{M\times d}$ which signifies the nodes that are most related to each token in the query via

$$\widetilde{a}_{q2n} = \mathrm{dup}(\mathrm{softmax}(\mathrm{max}_{\mathrm{col}}(\mathbf{S})))^{\top} \cdot f_n\,, \tag{6}$$

in which $\mathrm{max}_{\mathrm{col}}$ is the maximum function applied on across column of a matrix, which will transform $\mathbf{S}$ into $\mathbb{R}^{1\times M}$. Then function dup will duplicate it for T times into shape $\mathbb{R}^{T\times M}$. $f_n \in \mathbb{R}^{T\times d}$ is the original node feature before GCN layer.

Our bi-directional attention layer is the concatenation of the original nodes feature, nodes-to-query attention, the element-wise multiplication of nodes feature and nodes-to-query attention, and multiplication of nodes feature and query-to-nodes attention. It should be noted that the relation-aware nodes representation from GCN layer is just used to calculate the similarity matrix, and original node feature is used in rest calculation to obtain more general complementary information between graph and query. Edges are not taken in account because they are discrete and combined with nodes in GCN layer. The output is defined as

$$\widetilde{a} = \mathrm{concat}(f_n, \widetilde{a}_{n2q}, f_n \circ \widetilde{a}_{n2q}, f_n \circ \widetilde{a}_{q2n}). \tag{7}$$

3.5 Output layer

A 2-layer fully connect feed-forward network is employed to generate the final prediction, with $\tanh$ as the activation function in each layer. Softmax will be applied among the output. It uses query-aware representation of nodes from the attention layer as input, and its output is regarded as the probability of each node becoming answer. Since each candidate may appear several times in the graph, the probability of each candidate is the sum of all corresponding nodes. The loss function is defined as the cross entropy between the gold answer and its predicted probability.

4 Experiment

We used both unmasked and masked versions of the QAngaroo WIKIHOP dataset (Welbl et al., 2018) and followed its basic setting, in which masked version used specific tokens such as *__MASK1__* to replace original candidates tokens in documents. There are 43,738, 5,129 and 2,451 examples in the training set, the development set and the test set respectively, and test set is not public.

In the implementation[1], we used standard ELMo with a 1024 dimension representation. Besides, 300-dimension GLoVe pre-trained embeddings from 840B Web crawl data were used as token-level features. We used spaCy to provide additional 8-dimension NER and POS features. The dimension of the 1-layer linear network for nodes in multi-level feature module was 512 with $\tanh$ as activation function. A 2-layer Bi-LSTM was employed for queries whose hidden state size is 256. Then the feature dimension is $d = 512 + 8 + 8 = 528$. The GCN layer number L was set as 5. And the unit number of intermediate layers in output layer was 256.

In addition, the number of nodes and the query length were truncated as 500 and 25 respectively for normalized computation. Dropout with rate 0.2 was applied before GCN layer. Adam optimizer is employed with initial learning rate 2×10^{-4}, which will be halved for every 5 epochs, With batch size 32. It took about 14 hours for 50-epoch training on two GTX1080Ti GPUs using pre-built and pre-processed graph data generated from original corpus, which can significantly decrease the training time.

We consider the following baseline models: FastQA (Weissenborn et al., 2017), BiDAF (Seo et al., 2016), Coref-GRU (Dhingra et al., 2018), MHQA-GRN (Song et al., 2018), Entity-GCN (De Cao et al., 2018). Former three models are RNN-based models, while coreference relationship is involved in Coref-GRU. The last two models are graph-based models specially designed for multi-hop QA tasks.

As shown in Table 1, we collected three kinds of results. The dev and test results stand for the original validation and test sets respectively, noting that the test set is not public. In addition, we divide the original validation set of masked version into two parts evenly, one as a split validation set for tuning model and the other one as a split test set. The test[1] results are for the split test set.

Our BAG model achieves state-of-the art per-

[1] Source code is available on `https://github.com/caoyu1991/BAG`.

formance on both unmasked and masked data[2], with accuracy 69.0% on the test set, which is 1.4% higher in value than previous best model Entity-GCN. It is significant superior than FastQA and BiDAF due to leveraging of relationship information given by the graph and abandoning some distracting context in multiple documents. Although Coref-GRU extends GRU with coreference relationships, it is still not enough for multi-hop because hop relationships are not limited to coreference, entities with the same strings also existed across documents which can be used for reasoning. Both MHQA-GRN and Entity-GCN utilize graph networks to resolve relations among entities in documents. However, the lack of attention and complementary features limits their performance. Therefore our BAG model achieves the best performance under all data configurations. It is noticed that BAG only gets a small promotion on masked data. We argue that the reason is the attention between masks and queries generating less useful information compared to unmasked ones.

Moreover, ablation experimental results on unmasked version of the WIKIHOP dev set are given in Table 2. Once we remove the bi-directional attention and put the concatenation of nodes and queries directly into the output layer, it shows significant performance drop with more than 3%, proving the necessity of attention for reasoning in multi-hop QA. If we use linear-transformation-based single attention $a = h_n \mathbf{W_a} f_q$ given in (Luong et al., 2015) instead of our bi-directional attention, the accuracy drops with 2%, which means attention bi-directionality also contributes to the performance improvement. The similar condition will appear if we remove GCN, but use raw nodes as input for the attention layer.

In addition, if edge types are no longer considered, which makes R-GCN degraded to vanilla GCN, noticeable accuracy loss about 2% appears. The absence of multi-level features will also cause degradation. The removal of semantic-level features causes slight decline on the performance, including NER and POS features. Further removal of ELMo feature will causes a dramatical drop, which reflects the insufficiency of only using word embeddings as features for nodes and that contextual information is very important.

Models	Unmasked		Masked	
	dev	test	dev	test[1]
FastQA	27.2*	25.7	38.0*	48.3
BiDAF	49.7*	42.9	59.8*	57.5
Coref-GRU†	56.0*	59.3	-	-
MHQA-GRN‡	62.8*	65.4	-	-
Entity-GCN	64.8*	67.6	70.5*	68.1
BAG	**66.5**	**69.0**	**70.9**	**68.9**

Table 1: The performance of different models on both masked and unmasked version of WIKIHOP dataset. ([*] Results reported in original papers, others are obtained by official code. [†] Masked data is not suitable for coreference parsing. [‡] Some results are missing due to unavailability of source code.)

Models	Unmasked
Without Attention	63.1
Using Single Attention	64.5
Without GCN	63.3
Without edge type	63.9
Without NER, POS	66.0
+Without ELMo	60.5
Full Model	**66.5**

Table 2: Ablation test results of BAG model on the unmasked validation set of the WIKIHOP dataset.

5 Conclusion

We propose a Bi-directional Attention entity Graph convolutional network (BAG) for multi-hop reasoning QA tasks. Regarding task characteristics, graph convolutional networks (GCNs) are efficient to handle relationships among entities in documents. We demonstrate that both bi-directional attention between nodes and queries and multi-level features are necessary for such tasks. The former one aims to obtain query-aware node representation for answering, while the latter one provides contextual comprehension of isolated nodes in graphs. Our experimental results not only demonstrate the effectiveness of two proposed modules, but also show BAG achieves state-of-the-art performance on the WIKIHOP dataset.

Our future work will be making use of more complex relations between entities and building graphs in more general way without candidates.

Acknowledgements

This work was supported by Australian Research Council Projects under grants FL-170100117, DP-280103424.

[2] The paper was written on early Dec. 2018, during that time Entity-GCN is the best public model, and only one anonymous model is better than it.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2018. Question answering by reasoning across documents with graph convolutional networks. *arXiv preprint arXiv:1808.09920*.

Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2018. Neural models for reasoning over multiple mentions using coreference. *arXiv preprint arXiv:1804.05922*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gáabor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association of Computational Linguistics*, 6:317–328.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2017. Stochastic answer networks for machine reading comprehension. *arXiv preprint arXiv:1712.03556*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.

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

Linfeng Song, Zhiguo Wang, Mo Yu, Yue Zhang, Radu Florian, and Daniel Gildea. 2018. Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks. *arXiv preprint arXiv:1809.02040*.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Fastqa: A simple and efficient neural architecture for question answering. *CoRR, abs/1703.04816*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association of Computational Linguistics*, 6:287–302.

Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*.