

# Exploiting Explicit Paths for Multi-hop Reading Comprehension

Souvik Kundu<sup>+,\*</sup> and Tushar Khot<sup>†</sup> and Ashish Sabharwal<sup>†</sup>

<sup>+</sup>National University of Singapore, Singapore

<sup>†</sup> Allen Institute for Artificial Intelligence, Seattle, U.S.A.

souvik@comp.nus.edu.sg, tushark@allenai.org, ashishs@allenai.org

## Abstract

We focus on the task of multi-hop reading comprehension where a system is required to reason over a chain of multiple facts, distributed across multiple passages, to answer a question. Inspired by graph-based reasoning, we present a path-based reasoning approach for textual reading comprehension. It operates by generating potential paths across multiple passages, extracting implicit relations along this path, and composing them to encode each path. The proposed model achieves a 2.3% gain on the WikiHop Dev set as compared to previous state-of-the-art and, as a side-effect, is also able to explain its reasoning through explicit paths of sentences.

## 1 Introduction

Many Reading Comprehension (RC) datasets (Rajpurkar et al., 2016; Trischler et al., 2017; Joshi et al., 2017) have been proposed recently to evaluate a system’s ability to understand language based on its ability to answer a question given a passage. However, most of the questions in these datasets can be answered by using only a single sentence or passage. As a result, systems designed for these tasks may not acquire the capability to compose knowledge from multiple passages, a key aspect of natural language understanding. To remedy this, new datasets (Weston et al., 2015; Welbl et al., 2018; Khashabi et al., 2018a; Mihaylov et al., 2018) have been proposed recently that require a system to combine information from multiple sentences to arrive at the answer, often referred to as multi-hop reasoning.

Multi-hop reasoning has been studied for question answering over structured knowledge graphs (Lao et al., 2011; Guu et al., 2015; Das et al., 2017) where many of the successful models

**Question:** (always breaking my heart, record label, ?)  
**Supporting Passages:**  
 (p1) “Always Breaking My Heart” is the second single from Belinda Carlisle’s A Woman and a Man album, released in 1996 (see 1996 in music). It made ...  
 (p2) A Woman and a Man is the sixth studio album by American singer Belinda Carlisle, released in the United Kingdom on September 23, 1996 by Chrysalis Records (then part of the EMI Group, like Carlisle’s former ...  
**Candidates:** chrysalis Records, emi group, virgin records, ...  
**Answer:** chrysalis records  
**Path:**  
 (“Always Breaking My Heart” ... single from ... A Woman and a Man)  
 (A Woman and a Man ... released ... by ... Chrysalis Records)

Figure 1: Example illustrating our proposed path extraction and reasoning approach.

explicitly identify paths in the knowledge graph that led to the answer. While these models can be highly interpretable due to explicit path-based reasoning, they can not be directly applied to question answering in the absence of such structure. As a result, most multi-hop RC models (Dhingra et al., 2017; Hu et al., 2018) over text extend standard attention-based models from RC by iteratively updating the attention to “hop” over different parts of the text. Recently, graph-based models (Song et al., 2018; Cao et al., 2018) have been proposed for WikiHop, but these models still only implicitly combine knowledge from all the passages, and hence are unable to provide explicit reasoning paths for the selected answer.

We propose a model that explicitly extracts po-

<sup>\*</sup>This work was done during an internship at AI2

tential paths from text and encodes the knowledge captured by each path. Figure 1 shows how to apply this approach to an example in the WikiHop dataset (Welbl et al., 2018). In this example, we show two sample paths connecting an entity in the question, the song “Always Breaking My Heart”, to a candidate answer, “Chrysalis Records”, through a singer, “Belinda Carlisle”, and an album, “A Woman and a Man”.

Our model extracts implicit (latent) relations between entity pairs in a passage based on the contextual representation of the first sentence. For example, our model would try to extract the implicit relation capturing the *single from* relation between the song and the name of the album in the first passage. Similarly, it extracts the relation capturing the *released by* between the album and the record label in the second passage.

Having extracted these implicit relations, the model learns to compose them such that they map to the main relation in the query, namely, *record\_label*. In essence, our goal is to train a model that learns to extract implicit relations from text and valid compositions of these relations, such as:  $(x, \textit{single from}, y), (y, \textit{released by}, z) \rightarrow (x, \textit{record_label}, z)$ . Apart from focusing on the specific entities in each passage, we also learn to compose the aggregate passage representations in a path to capture more global information i.e.  $\text{encode}(p1), \text{encode}(p2) \rightarrow (x, \textit{record_label}, z)$ .

We make 3 main contributions: (1) a novel path-based reasoning approach for multi-hop question answering over text. (2) A model that learns to extract implicit relations from text and compose them. (3) A competitive model<sup>1</sup> on the WikiHop dataset that also produces explanations via reasoning over explicit paths.

## 2 Related Work

We present related work from question answering over text, semi-structured knowledge and knowledge graphs.

### 2.1 Multi-hop RC

Recently multiple datasets such as bAbI (Weston et al., 2015), Multi-RC (Khashabi et al., 2018a), WikiHop (Welbl et al., 2018), and OpenBookQA (Mihaylov et al., 2018) have been proposed to encourage research in multi-hop ques-

tion answering over text. Multi-hop models for these tasks can be categorized into: state-based or graph-based reasoning models. The state-based reasoning models (Dhingra et al., 2017; Shen et al., 2017; Hu et al., 2018) are closer to standard attention-based reading comprehension model with an additional “state” representation that is iteratively updated. The changing state representation results in the model focusing on different parts of the passage during each iteration allowing the model to combine information from different parts of the passage. The graph-based reasoning models (Dhingra et al., 2018; Cao et al., 2018; Song et al., 2018), on the other hand, create graphs over entities within these passages and update the entity representations via recurrent/convolutional networks. Our approach explicitly identifies paths connecting the entities in the question to the answer choices.

### 2.2 Semi-structured QA

Our model is closer to Integer Linear Programming (ILP)-based methods (Khashabi et al., 2016; Khot et al., 2017; Khashabi et al., 2018b) developed for science question answering. These approaches define an ILP program to find optimal “support graphs” connecting words in the question to the choices through a semi-structured knowledge representation. However, these models require a manually authored and tuned ILP program and need to convert text into a semi-structured representation that can be noisy (such as Open IE tuples (Khot et al., 2017) or SRL frames (Khashabi et al., 2018b)). Our models are trained end-to-end on the target dataset and allow the model to discover the relevant relations from text.

### 2.3 Knowledge Graph QA

Question answering datasets on knowledge graphs such as Freebase (Bollacker et al., 2008) require systems to map queries to a single relation (Bordes et al., 2015), a path (Gua et al., 2015) or complex structured queries (Berant et al., 2013) over these graphs. While early models (Lao et al., 2011; Gardner and Mitchell, 2015) on this task focused on creating path-based features, recent neural path-based models (Gua et al., 2015; Das et al., 2017; Toutanova et al., 2016) encode the entities and relations along the path and compose them using recurrent networks. However, the input knowledge graphs have entities and relations shared across all examples that the model can ex-

<sup>1</sup>Best reported dev results with evaluations on the blind test set pending.

exploit during learning (e.g. through learned entity/relation embeddings). When reasoning with text, our model has to learn these representations purely based on their local context.

### 3 Approach Overview

In this work, we focus on the multiple-choice RC setting. Given a question and a set of passages as support, the task is to find the correct answer from a predefined set of candidate answer choices.

In the WikiHop dataset, a question  $Q$  is given in the form of a tuple  $(h_e, r, ?)$ , where  $h_e$  represents the head entity and  $r$  represents the relation between  $h_e$  and the unknown tail entity. The task is to select the unknown tail entity from a given set of answer candidates represented as  $(c_1, c_2, \dots, c_N)$ .

To perform multi-hop reasoning, we extract multiple paths  $\mathbb{P}$  (described in the next section) connecting  $h_e$  to each  $c_k$  from the supporting passages,  $\mathcal{P} = p_1, \dots, p_M$ . We represent the  $j$ -th path for candidate  $c_k$  as  $p_{kj}$ . For simplicity, we only consider two-hop paths, i.e.,  $p_{kj} = h_e \rightarrow e_1 \rightarrow c_k$  where  $e_1$  is called an *intermediate entity*. Note that, while we only consider up to two hops of reasoning<sup>2</sup>, our approach can be easily extended to more than two hops.

### 4 Path Extraction

The first step in our approach is extracting paths from text passages for each candidate given a question. Consider the example in Figure 1. Overall, there are four steps in our path extraction approach:

- We find a passage  $p_1$  that contains  $h_e$  of the question  $Q$ . In our example, we would find the first supporting passage that contains *always breaking my heart*.
- Then, we find all the named entities and noun phrases that appear in the same sentence as  $h_e$  or in the subsequent sentence. For instance, we would collect *Belinda Carlisle*, *A Woman and a Man*, or *album* as intermediate entity  $e_1$ .
- Now, we find a passage  $p_2$  that contains any of the intermediate entities found in the previous step. For instance, we find the second passage that contains both *Belinda Carlisle* and *A Woman and a Man*.

<sup>2</sup>We observed that most of WikiHop questions can be answered with 2 hops of reasoning.

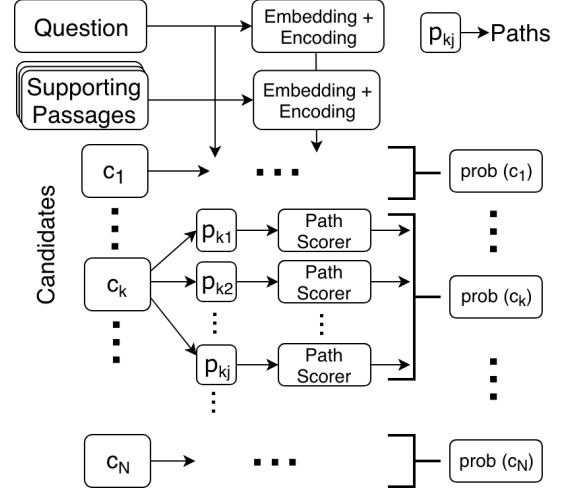


Figure 2: Architecture of the proposed model.

- Finally, we check if  $p_2$  contains any of the candidate answer choices. For instance,  $p_2$  contains *chrysalis records* and *emi group* in our example.

The extracted paths for candidate *chrysalis records* can be summarized as a set of entity sequences: (*always breaking my heart*, *Belinda Carlisle*, *chrysalis records*), (*always breaking my heart*, *A Woman and a Man*, *chrysalis records*). Similarly, we can extract paths for the other candidate, *emi group*. Notably, our path extraction method can be easily extended for three or more hops by repeating step 2 and 3. Specifically, for  $m$  hop reasoning, step 2 and 3 need to be repeated  $(m - 1)$  times. For one hop reasoning, i.e., when a single passage is sufficient to answer a question, we construct the path with  $e_1$  as *null*. In this case, both  $h_e$  and answer candidate are found in a single passage.

### 5 Path-based Multi-Hop QA Model

Once we have all the paths collected for the questions and candidates, we feed them to our proposed model. An overview of our proposed model is depicted in Fig 2. The key component of our model is the path-scorer module that computes the score for each path  $p_{kj}$ . We normalize these scores across all paths and compute the probability of a candidate by summing the normalized scores of the paths associated with that candidate. In this way, the probability of candidate  $c_k$  being an answer can be given as:

$$\text{prob}(c_k) = \sum_j \text{score}(p_{kj}) \quad (1)$$

Next we describe three key model components, given the question  $Q$ , supporting passages  $p_1$  and  $p_2$ , the candidate  $c_k$ , and the locations of  $h_e$ ,  $e_1$ ,  $c_k$  in these passages as input.

- (1) Embedding and Encoding (Section 5.1)
- (2) Path Encoding (Section 5.2)
- (3) Path Scoring (Section 5.3)

## 5.1 Embedding and Encoding

Here, we will describe the text embedding and encoding approaches used in the model. We use the same embedding and contextual encoding for question, supporting passages, and candidate answer choices. For word embedding, we use pre-trained 300d embedding vectors from GloVe (Pennington et al., 2014). For out of vocabulary (OOV) words, we use randomly initialized vectors. For contextual encoding, we use bi-directional long short term memory (BiLSTM) networks (Hochreiter and Schmidhuber, 1997). Let  $\mathbf{e}_t^{s_p}$  be the  $t$ th word embedding vector of the  $p$ th supporting passage. To get the contextual representations, we use the concatenation of the forward and backward hidden states of the BiLSTM, i.e.  $s_{p,t} = \overrightarrow{\mathbf{h}}_t^{s_p} \parallel \overleftarrow{\mathbf{h}}_t^{s_p}$  where  $\parallel$  is used to indicate concatenation,  $\mathbf{h}_t^{s_p}$  is the LSTM hidden state representation, and  $s_{p,t}$  is the contextual representation of the  $t$ th token in  $p$ th passage.

The final encoded representation for the  $p$ th supporting passage can be obtained by stacking these vectors into  $S_p \in \mathbb{R}^{T \times H}$  where  $H$  is the number of hidden units for the BiLSTMs. Similarly, the sequence level encoding for a question  $Q \in \mathbb{R}^{U \times H}$  and any candidate answer choice  $C_k \in \mathbb{R}^{V \times H}$ .  $T$ ,  $U$ , and  $V$  represent the number of tokens in the  $p$ th supporting passage, question, and  $k$ th candidate answer choice respectively.

## 5.2 Path Encoding

This is the core component of the proposed model architecture. After extracting the paths as discussed in Section 4, they are encoded inside an end-to-end neural network architecture. Path encoder consists of two components: context-based path encoder and passage-based path encoder.

### Context-based Path Encoding

In context-based path encoding, we aim to implicitly encode the relation between  $(h_e, e_1)$ , and  $(e_1, c_k)$ . These implicit relation representations are

further composed to encode a path representation for  $(h_e, e_1, c_k)$ . Note that,  $(h_e, e_1)$  and  $(e_1, c_k)$  are located in different passages, say  $p_1$  and  $p_2$  respectively. For clarity, we denote  $(e_1, c_k)$  as  $(e_1', c_k)$  from now onwards. First, we extract the contextual representations for each  $h_e$ ,  $e_1$ ,  $e_1'$ , and  $c_k$ . Based on the locations of these entities in the corresponding passages, we extract the boundary vectors from the passage encoding representation. For instance, if  $h_e$  appears in the  $p$ th supporting passage from token  $i_1$  to  $i_2$  ( $i_1 \leq i_2$ ), then the contextual encoding of  $h_e$ ,  $\mathbf{g}_{h_e} \in \mathbb{R}^{2H}$  can be given as:

$$\mathbf{g}_{h_e} = s_{p_1, i_1} \parallel s_{p_1, i_2} \quad (2)$$

Similarly, we obtain the location encoding vectors  $\mathbf{g}_{e_1}$ ,  $\mathbf{g}_{e_1'}$ , and  $\mathbf{g}_{c_k}$ . Note that, if they appear multiple locations, we use the mean vector representation for all the locations.

Now, we extract the implicit relations between  $h_e$  and  $e_1$  as  $\mathbf{r}_{h_e, e_1} \in \mathbb{R}^H$  with a simple feed forward layer:

$$\mathbf{r}_{h_e, e_1} = \text{FFL}(\mathbf{g}_{h_e}, \mathbf{g}_{e_1}) \quad , \quad (3)$$

where FFL can be described as:

$$\text{FFL}(\mathbf{a}, \mathbf{b}) = \tanh(\mathbf{a}\mathbf{W}_a + \mathbf{b}\mathbf{W}_b) \quad , \quad (4)$$

where  $\mathbf{a} \in \mathbb{R}^{H'}$  and  $\mathbf{b} \in \mathbb{R}^{H''}$  are input vectors.  $\mathbf{W}_a \in \mathbb{R}^{H' \times H}$  and  $\mathbf{W}_b \in \mathbb{R}^{H'' \times H}$  are trainable weight matrices. The bias vectors are not shown here for simplicity.

Similarly, we compute the implicit relation between  $e_1'$  and  $c_k$  as  $\mathbf{r}_{e_1', c_k} \in \mathbb{R}^H$ , using their location encoding vectors  $\mathbf{g}_{e_1'}$  and  $\mathbf{g}_{c_k}$ . Finally, we compose the two implicit relation vectors with a feed forward layer to obtain a context-based path representation  $\mathbf{x}_{ctx} \in \mathbb{R}^H$ :

$$\mathbf{x}_{ctx} = \text{FFL}(\mathbf{r}_{h_e, e_1}, \mathbf{r}_{e_1', c_k}) \quad (5)$$

### Passage-based Path Encoding

In passage-based path encoder, we use the whole passages to compute the path representation. Let us consider that,  $(h_e, e_1)$  and  $(e_1', c_k)$  appear in the supporting passages  $p_1$  and  $p_2$  respectively. We encode both  $p_1$  and  $p_2$  into single vectors based on the interaction with question encoding representation  $Q$ . We first compute a question-weighted representation for the tokens and then compute an aggregate vector for each passage.



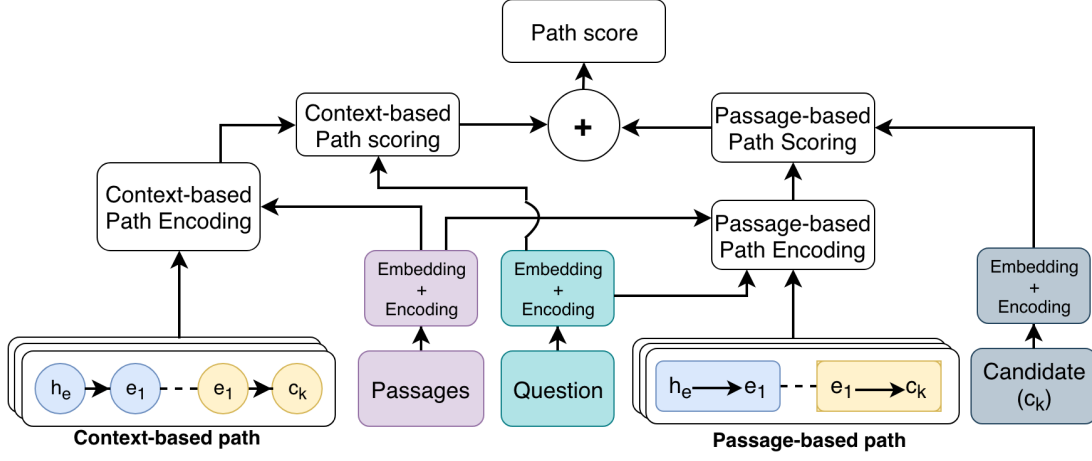


Figure 3: Architecture of the proposed path scoring module.

### Question-weighted Passage Representation:

For any  $p$ th passage, we first compute the attention matrix  $\mathbf{A}_p \in \mathbb{R}^{T \times U}$  which captures the similarity between the passage and question words. Then, we calculate a question-aware passage representation  $\mathbf{S}_p^{q_1} \in \mathbb{R}^{T \times H}$ , where  $\mathbf{S}_p^{q_1} = \mathbf{A}\mathbf{Q}$ . Similarly, a passage-aware question representation,  $\mathbf{Q}_p \in \mathbb{R}^{U \times H}$  is computed, where  $\mathbf{Q}_p = \mathbf{A}^\top \mathbf{S}_p$ .

Now based on this updated question representation, we compute another passage representation  $\mathbf{S}_p^{q_2} \in \mathbb{R}^{T \times H}$  from  $\mathbf{Q}_p$ , where  $\mathbf{S}_p^{q_2} = \mathbf{A}\mathbf{Q}_p$ . Intuitively,  $\mathbf{S}_p^{q_1}$  captures the important passage words based on the question whereas  $\mathbf{S}_p^{q_2}$  focuses on the passage-relevant question words. The idea of encoding a passage after interacting with the question multiple times is inspired from the Gated Attention Reader model (Dhingra et al., 2017).

**Aggregate Passage Representation:** To get a single passage vector, we first concatenate the two passage representations for each token,  $\mathbf{S}_p^q \in \mathbb{R}^{T \times 2H} = \mathbf{S}_p^{q_1} \parallel \mathbf{S}_p^{q_2}$ . We then use an attentive pooling mechanism for aggregating the token representations. The aggregated vector  $\tilde{\mathbf{s}}_p \in \mathbb{R}^{2H}$  for  $p$ th passage can be obtained as:

$$\begin{aligned} a_t^p &\propto \exp(\mathbf{s}_{p,t}^q \mathbf{w}^\top) \\ \tilde{\mathbf{s}}_p &= \mathbf{a}^p \mathbf{S}_p^q, \end{aligned} \quad (6)$$

where  $\mathbf{w} \in \mathbb{R}^{2H}$  is a learned vector. In this way, we obtain the aggregated vector representations for both supporting passages  $p_1$  and  $p_2$  as  $\tilde{\mathbf{s}}_{p_1} \in \mathbb{R}^{2H}$  and  $\tilde{\mathbf{s}}_{p_2} \in \mathbb{R}^{2H}$  respectively.

**Path Encoding:** Now, we compose the aggregated passage vectors to obtain the passage-based

path representation  $\mathbf{x}_{psg} \in \mathbb{R}^H$  by using a simple feed forward network:

$$\mathbf{x}_{psg} = \text{FFL}(\tilde{\mathbf{s}}_{p_1}, \tilde{\mathbf{s}}_{p_2}) \quad (7)$$

### 5.3 Path Scoring

**Context-based Path Scoring:** We score the context-based paths based on the interaction between question encoding and context-based path encoding. First, we aggregate the question into a single vector. As the question is in the form  $(h_e, r)$ , we take the first and last hidden state representation from the question encoding  $\mathbf{Q}$  to explicitly cover both head entity and relation. The aggregated question vector  $\tilde{\mathbf{q}} \in \mathbb{R}^H$  can be given as:

$$\tilde{\mathbf{q}} = (\mathbf{q}_0 \parallel \mathbf{q}_U) \mathbf{W}_q, \quad (8)$$

where  $\mathbf{W}_q \in \mathbb{R}^{2H \times H}$  is a learnable weight matrix. The combined representation  $\mathbf{y}_{ctx,q} \in \mathbb{R}^H$  of the question and a context-based path can be given as:

$$\mathbf{y}_{ctx,q} = \text{FFL}(\mathbf{x}_{ctx}, \tilde{\mathbf{q}}) \quad (9)$$

Finally, the scores for context-based paths are derived:

$$z_{ctx} = \mathbf{y}_{ctx,q} \mathbf{w}_{ctx}^\top, \quad (10)$$

where  $\mathbf{w}_{ctx} \in \mathbb{R}^H$  is a learnable vector.

**Passage-based Path Scoring:** On the other hand, we capture the interaction between passage-based path encoding vector and candidate encoding to score the passage-based paths<sup>3</sup>. We aggregate a candidate encoding representation  $\mathbf{C}_k$  into a single vector  $\tilde{\mathbf{c}}_k \in \mathbb{R}^H$  by applying an attentive

<sup>3</sup>Since the passage based encoding already uses the question representation to compute attention

pooling operation similar to Eq. 6. Now, the score for passage-based path is computed as follows:

$$z_{psg} = \tilde{\mathbf{c}}_k \mathbf{x}_{psg}^\top \quad (11)$$

Finally, the unnormalized score for a path  $p_{kj}$  is:

$$z = z_{ctx} + z_{psg} \quad (12)$$

and the normalized  $score(p_{kj})$  is calculated by applying a softmax over all the paths and candidates.

## 6 Experiments

We start by describing the experimental setup, which includes the dataset and experimental configuration. Then, we present the results and analysis of our model.

### 6.1 Settings

For experimentation, we used the recently proposed WikiHop dataset (Welbl et al., 2018). In this work, we considered the unmasked version of the dataset. WikiHop is a large scale multi-hop QA dataset consisting of about 51K questions. Each question is associated with an average of 13.7 supporting passages collected from Wikipedia. Each passage consists of 36.4 tokens on average.

We use Spacy for tokenization. For word embedding, we use the 840B 300-dimensional pre-trained word vectors from GloVe and we do not update them during training. For simplicity, we do not use any character embedding in our model. The number of hidden units in all LSTMs is 50 ( $H = 100$ ). We use dropout (Srivastava et al., 2014) with probability 0.25 for every learnable layer. During training, the minibatch size is fixed at 8. We use the Adam optimizer (Kingma and Ba, 2015) with learning rate of 0.001 and clipnorm of 5. We use cross entropy loss for training. This being a multiple-choice QA task, we use accuracy as the evaluation metric.

### 6.2 Results

Table 1 presents our results<sup>4</sup> in comparison with several recently proposed models for multi-hop QA. We show the best dev results obtained from each of the competing entries.

Welbl et al. (2018) presented the results of BiDAF (Seo et al., 2017) on the WikiHop dataset. Dhingra et al. (2018) incorporated coreference

<sup>4</sup>We are in the process of obtaining the results on the hidden Test set.

Model	Accuracy (%)	
	Dev	Test
Welbl et al. (2018)	-	42.9
Dhingra et al. (2018)	56.0	59.3
Song et al. (2018)	62.8	65.4
Cao et al. (2018)	64.8	67.6
Our model	<b>67.1</b>	-

Table 1: Accuracy on the WikiHop Dev set. Our proposed approach achieves a significant 2.3% gain over the previous best.

Model	Accuracy (%)	
	Dev	$\Delta$
Full model	<b>67.1</b>	-
- context-based path	64.7	2.4
- passage-based path	63.2	3.9

Table 2: Ablation results on the WikiHop dev set. Both context-based and passage-based paths significantly impact the results.

connections inside GRU network to capture coreference links while obtaining the contextual representation. Recently, Cao et al. (2018) and Song et al. (2018) proposed graph neural network approaches for multi-hop reading comprehension. While the high level idea is similar for these work, Cao et al. (2018) used ELMO (Peters et al., 2018) for embedding which has proven to be very useful in the recent past in many natural language processing tasks. Table 1 clearly shows that our proposed model significantly outperforms the prior models by significant margin<sup>5</sup>. Additionally, in contrast to our model, the competing models do not possess the capability to identify which particular entity chains are leading to the final predicted answer.

Table 2 shows the ablation results on the WikiHop development set. When we do not consider the context-based paths in the model, only the passage-based paths are considered and vice versa. As we can see that, performance of the model degrades significantly when we ablate any of the two path encoding modules. Also the information captured by context-based and passage-based paths are complementary to some extent, as evidenced by the larger drop in the model with no paths. Intuitively, in context-based paths, lim-

<sup>5</sup>With 5129 dev questions, any gain above 1.3% would be significant at  $p = 0.05$  based on the Wilson score interval (Wilson, 1927).

ited and more fine-grained context is considered due to the use of syntactically matched location-based encoding representations of the entities that are used to construct a path. On the contrary, the passage-based path encoder computes the path representation with semantic similarity-based aggregation of the entire passages.

One key aspect of our proposed model is that it can indicate the paths that contribute the most towards predicting the answer choice. Table 3 illustrates the top two paths for two example questions which lead to correctly predicted final answer choice. In the first question, the top-2 paths are formed by connecting *Zoo Lake* to *Gauteng* through the intermediate entities *Johannesburg* and *South Africa* respectively. In the second example, the science fiction novel *This Day All Gods Die* is connected to the publisher *Bantam Books* through the author *Stephen R. Donaldson* in the first path, and through the collection *Gap Cycle* in the second path.

## 7 Conclusion

We present a novel path-based multi-hop reading comprehension model that achieves state-of-the-art results on the WikiHop Dev set. We also show that our model can explain its reasoning through paths across multiple passages. Our approach can potentially be generalized to longer chains (more than 2 hops) and longer natural language questions, which we will explore further.

## References

- Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of EMNLP*.
- 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 ACM SIGMOD international conference on Management of data*.
- Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. In *NIPS*.
- Nicola De Cao, Wilker Aziz, and Ivan Titov. 2018. Question answering by reasoning across documents with graph convolutional networks. *CoRR*, abs/1808.09920.
- Rajarshi Das, Arvind Neelakantan, David Belanger, and Andrew McCallum. 2017. Chains of reasoning over entities, relations, and text using recurrent neural networks. In *Proceedings of EACL*.
- Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2018. Neural models for reasoning over multiple mentions using coreference. In *Proceedings of NAACL*.
- Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2017. Gated-attention readers for text comprehension. In *Proceedings of ACL*.
- Matt Gardner and Tom M. Mitchell. 2015. Efficient and expressive knowledge base completion using subgraph feature extraction. In *Proceedings of EMNLP*.
- Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. In *Proceedings of EMNLP*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Minghao Hu, Yuxing Peng, Zhen Huang, Xipeng Qiu, Furu Wei, and Ming Zhou. 2018. Reinforced mnemonic reader for machine reading comprehension. In *Proceedings of IJCAI*.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of ACL*.
- Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018a. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of NAACL*.
- Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Peter Clark, Oren Etzioni, and Dan Roth. 2016. Question answering via integer programming over semi-structured knowledge. In *Proceedings of IJCAI*.
- Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2018b. Question answering as global reasoning over semantic abstractions. In *Proceedings of AAAI*.
- Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. Answering complex questions using open information extraction. In *Proceedings of ACL*.
- Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.
- Ni Lao, Tom M. Mitchell, and William W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of EMNLP*.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of EMNLP*.

<p><b>Question:</b> (zoo lake, located in the administrative territorial entity, ?)</p> <p><b>Answer:</b> gauteng</p> <p><b>Rank-1 Path:</b> (zoo lake, Johannesburg, gauteng)</p> <p>Passage1: ... Zoo Lake is a popular lake and public park in Johannesburg , South Africa . It is part of the Hermann Eckstein Park and is ...</p> <p>Passage2: ... Johannesburg ( also known as Jozi , Joburg and eGoli ) is the largest city in South Africa and is one of the 50 largest urban areas in the world . It is the provincial capital of Gauteng , which is ...</p> <p><b>Rank-2 Path:</b> (zoo lake, South Africa, gauteng)</p> <p>Passage1: ... Zoo Lake is a popular lake and public park in Johannesburg , South Africa . It is ...</p> <p>Passage2: ... aka The Reef , is a 56-kilometre - long north - facing scarp in the Gauteng Province of South Africa . It consists of a ...</p>	
<p><b>Question:</b> (this day all gods die, publisher, ?)</p> <p><b>Answer:</b> bantam books</p> <p><b>Rank-1 Path:</b> (this day all gods die, Stephen R. Donaldson, bantam books)</p> <p>Passage1: ... All Gods Die , officially The Gap into Ruin : This Day All Gods Die , is a science fiction novel by Stephen R. Donaldson , being the final book of The Gap Cycle ...</p> <p>Passage2: ... The Gap Cycle ( published 19911996 by Bantam Books and reprinted by Gollancz in 2008 ) is a science fiction story , told in a series of 5 books , written by Stephen R. Donaldson . It is an ...</p> <p><b>Rank-2 Path:</b> (this day all gods die, Gap Cycle, bantam books)</p> <p>Passage1: ... All Gods Die , officially The Gap into Ruin : This Day All Gods Die , is a science fiction novel by Stephen R. Donaldson , being the final book of The Gap Cycle ...</p> <p>Passage2: ... The Gap Cycle ( published 19911996 by Bantam Books and reprinted by Gollancz in 2008 ) is a science fiction story ...</p>	

Table 3: Top two scoring paths for two questions in the dev set. For example, in the Rank-1 path for the first question, the model learns to compose the implicit *located in* relations between (Zoo lake, Johannesburg) and (Johannesburg, Gauteng).

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In <i>Proceedings of EMNLP</i> .	graph-structured passage representation for multi-hop reading comprehension with graph neural networks. <i>CoRR</i> , abs/1809.02040.
Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In <i>Proceedings of NAACL</i> .	Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. <i>JMLR</i> , 15(1):1929–1958.
Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In <i>Proceedings of EMNLP</i> .	Kristina Toutanova, Victoria Lin, Wen tau Yih, Hoi-fung Poon, and Chris Quirk. 2016. Compositional learning of embeddings for relation paths in knowledge base and text. In <i>Proceedings of ACL</i> .
Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In <i>Proceedings of ICLR</i> .	Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordani, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In <i>Proceedings of the 2nd Workshop on Representation Learning for NLP</i> .
Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2017. Reasonet: Learning to stop reading in machine comprehension. In <i>Proceedings of KDD</i> .	Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. <i>TACL</i> , 6:287–302.
Linfeng Song, Zhiguo Wang, Mo Yu, Yue Zhang, Radu Florian, and Daniel Gildea. 2018. Exploring	Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards ai-complete ques-



tion answering: A set of prerequisite toy tasks.  
*CoRR*, abs/1502.05698.

Edwin B. Wilson. 1927. Probable inference, the law of succession, and statistical inference. *JASA*, 22(158):209–212.