Understanding Dataset Design Choices for Multi-hop Reasoning

Jifan Chen and **Greg Durrett**
The University of Texas at Austin
{jfchen,gdurrett}@cs.utexas.edu

Abstract

Learning multi-hop reasoning has been a key challenge for reading comprehension models, leading to the design of datasets that explicitly focus on it. Ideally, a model should not be able to perform well on a multi-hop question answering task without doing multi-hop reasoning. In this paper, we investigate two recently proposed datasets, WikiHop (Welbl et al., 2018) and HotpotQA (Yang et al., 2018). First, we explore sentence-factored models for these tasks; by design, these models cannot do multi-hop reasoning, but they are still able to solve a large number of examples in both datasets. Furthermore, we find spurious correlations in the unmasked version of WikiHop, which make it easy to achieve high performance considering only the questions and answers. Finally, we investigate one key difference between these datasets, namely span-based vs. multiple-choice formulations of the QA task. Multiple-choice versions of both datasets can be easily gamed, and two models we examine only marginally exceed a baseline in this setting. Overall, while these datasets are useful testbeds, high-performing models may not be learning as much multi-hop reasoning as previously thought.

1 Introduction

Question answering from text (Richardson et al., 2013; Hill et al., 2015; Hermann et al., 2015; Rajpurkar et al., 2016) is a key challenge problem for NLP that tests whether models can extract information based on a query. However, even sophisticated models that perform well on QA benchmarks (Seo et al., 2017; Shen et al., 2017; Yu et al., 2018) may only be doing shallow pattern matching of the question against the supporting passage (Weissenborn et al., 2017). More recent work (Kumar et al., 2016; Joshi et al., 2017; Welbl et al., 2018) has emphasized gathering information from different parts of a passage to answer the question, leading to a number of models designed to do *multi-hop reasoning*. Two recent large-scale datasets have been specifically designed to test multi-hop reasoning: WikiHop (Welbl et al., 2018) and HotpotQA (Yang et al., 2018).

In this paper, we seek to answer two main questions. First, although the two datasets are explicitly constructed for multi-hop reasoning, do models really need to do multi-hop reasoning to do well on them? Recent work has shown that large-scale QA datasets often do not exhibit their advertised properties (Chen et al., 2016; Kaushik and Lipton, 2018). We devise a test setting to see whether multi-hop reasoning is necessary: can a model which treats each sentence independently select the sentence containing the answer? This provides a rough estimate of the fraction of questions solvable by a non-multi-hop system. Our results show that more than half of the questions in WikiHop and HotpotQA do not require multi-hop reasoning to solve. Surprisingly, we find that a simple baseline which ignores the passage and only uses the question and answer can achieve strong results on WikiHop and a modified version of HotpotQA, further confirming this view.

Second, we study the nature of the supervision on the two datasets. One critical difference is that HotpotQA is span-based (the answer is a span of the passage) while WikiHop is multiple-choice. How does this difference affect learning and evaluation of multi-hop reasoning systems? We show that a multiple-choice version of HotpotQA is vulnerable to the same baseline that performs well on WikiHop, showing that this distinction may be important from an evaluation standpoint. Furthermore, we show that a state-of-the-art model, BiDAF++, trained on span-based HotpotQA and adapted to the multiple-choice setting outperforms the same model trained natively on

the multiple-choice setting. However, even in the span-based setting, the high performance of the sentence-factored models raises questions about whether multi-hop reasoning is being learned.

Our conclusions are as follows: (1) Many examples in both WikiHop and HotpotQA do not require multi-hop reasoning to solve, as the sentence-factored model can find the answers. (2) On WikiHop and a multiple-choice version of HotpotQA, a no context baseline does very well. (3) Span-based supervision provides a harder testbed than multiple choice by having more answers to choose from, but given the strong performance of the sentence-factored models, it is unclear whether any of the proposed models are doing a good job at multi-hop reasoning in any setting.

2 Datasets

WikiHop Welbl et al. (2018) introduced this English dataset specially designed for text understanding across multiple documents. The dataset consists of 40k+ questions, answers, and passages, where each passage consists of several documents collected from Wikipedia. Questions are posed as a query of a relation r followed by a head entity h, with the task being to find the tail entity t from a set of entity candidates E. Annotators followed links between documents and were required to use multiple documents to get the answer.

HotpotQA Yang et al. (2018) proposed a new dataset with 113k English Wikipedia-based question-answer pairs. The questions are diverse, falling into several categories, but all require finding and reasoning over multiple supporting documents to answer. Models should choose answers by selecting variable-length spans from these documents. Sentences relevant to finding the answer are annotated in the dataset as "supporting facts" so models can use these at training time as well.

3 Probing Multi-hop Datasets

In this section, we seek to answer whether multi-hop reasoning is really needed to solve these two multi-hop datasets.

3.1 Sentence-Factored Model Test

If a question requires a multi-hop model, then we should not be able to figure out the answer by only looking at the question and each sentence separately. Based on this idea, we propose a sentence-factored modeling setting, where a model must predict which sentence contains the answer but must score each sentence independently, i.e., without using information from other sentences in this process. Identifying the presence of the answer is generally easier than predicting the answer directly, particularly if a sentence is complicated, and is still sufficient to provide a bound on how strongly multi-hop reasoning is required. Figure 1 shows a typical example from these datasets, where identifying the answer (*Delhi*) requires bridging to an entity not mentioned in the question.

Method	Random	Factored	Factored BiDAF
WikiHop	6.5	60.9	66.1
HotpotQA	5.4	45.4	57.2
SQuAD	22.1	70.0	88.0

Table 1: The accuracy of our proposed sentence-factored models on identifying answer location in the development sets of WikiHop, HotpotQA and SQuAD. *Random*: we randomly pick a sentence in the passage to see whether it contains the answer. *Factored* and *Factored BiDAF* refer to the models of Section 3.1. As expected, these models perform better on SQuAD than the other two datasets, but the model can nevertheless find many answers in WikiHop especially.

Simple Factored Model We encode each passage sentence s_i and the question q into a contextual representation h_{s_i} and h_q using a bi-directional GRU (Chung et al., 2014). Then, $S_i = h_{s_i}^\top W h_q$; that is, compute a bilinear product of these representations with trainable weights W to get the score of the ith sentence. Finally, let $p_i = \text{softmax}_i(S_i)$; softmax over the sentences to get a probability distribution. We maximize the marginal log probability of picking a sentence containing the correct answer: $\log(\sum_{i:s_i \in \mathbf{s}^*} p_i)$, where $\mathbf{s}^*$ is the set of sentences containing the answer. During evaluation, we pick the sentence s with the highest score and treat it as correct if it contains the answer.

Factored BiDAF We encode the question and each sentence *separately* using bi-GRUs. Then, we generate the question-aware token representation for each token of sentence by using a co-attention layer (Seo et al., 2017). Finally, we max-pool over each sentence to get the sentence representation and feed those to a FFNN to compute the sentence score. Training and inference are the same as for the simple model.

We run this test on both datasets as well as

Question: **The Oberoi family** is part of a hotel company that has a head office in what city?

……

The Oberoi family is an Indian family that is famous for its involvement in hotels, namely through **The Oberoi Group**.

……

The Oberoi Group is a hotel company with its head office in **Delhi**.

Figure 1: An example from the HotpotQA dev set. Here, a model should have to form a reasoning chain *Oberoi family* → *Oberoi Group* → *Delhi* to arrive at the answer. However, the sentence containing *Delhi* has a substantial lexical overlap with the question, so strong QA systems can answer it directly.

SQuAD (Rajpurkar et al., 2016), where multi-hop reasoning is only needed in a few questions. Results in Table 1 indicate that although intentionally created for multi-hop reasoning, for more than half of questions in WikiHop and HotpotQA, we can figure out where the answer is without doing multi-hop reasoning. This result is initially surprising, but one reason it may be possible is suggested by the example from HotpotQA shown in Figure 1. We can see that the model could easily figure out the answer sentence without looking at the bridging entities using lexical cues alone. This observation is also in accordance with the work of Jansen (2018), which demonstrates that high performance for a simple baseline can be achieved in cases when passages have increasing lexical overlap with the question.

We note that this method potentially overestimates performance of a non-multi-hop model on HotpotQA, since there are some examples where many plausible answers are in the same sentence and require other context to resolve. However, these still form a minority in the dataset (see Table 3 of Yang et al. (2018)).

3.2 No Context Baseline

The results of the previous section show that a model can identify correspondences between questions and answer sentences. One other pair of correlations we can study is suggested in the work of Kaushik and Lipton (2018), namely examining question-answer correlations independent of the passage. We construct a "no context" baseline to verify whether it is possible to pick the correct answer without consulting the passage. In a similar fashion to the factored model, we encode the query q and each answer candidate c_i using a bi-GRU and once again compute a bilinear product between them to get the scores over candidates, making no reference to the document.

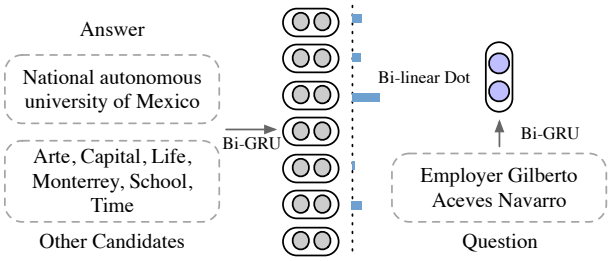

Figure 2: An example of question and candidates from WikiHop. Here we can see that among the candidates, only *National autonomous university of Mexico* is an organization which could be Navarro's employer; the model may pick up on this entity typing.

NoContext	Coref-GRU	MHQA-GRN	Entity-GCN
59.70	56.00	62.80	64.80

Table 2: The results of our no-context baseline compared with Coref-GRU (Dhingra et al., 2018), MHQA-GRN (Song et al., 2018), and Entity-GCN (De Cao et al., 2018) on the WikiHop dev set.

Results of this model on the multiple-choice WikiHop dataset are shown in Table 2. Surprisingly, the no-context baseline achieves high performance, comparable to some recently-published systems, showing that WikiHop is actually possible to solve reasonably well without using the document at all. One possible reason for this is that this model can filter possible answers based on expected answer type (Sugawara et al., 2018), as shown in the example of Figure 2, or perhaps capture other correlations between training and test. This model substantially outperforms the unlearned baseline reported in the WikiHop paper (Welbl et al., 2018) (38.8%) as well as the BiDAF (Seo et al., 2017) results reported there (42.9%).

4 Span-based vs. Multiple-choice

The no context model indicates that having multiple-choice questions may provide an avenue for a dataset to be gamed. In order to investigate the difference in multiple-choice vs. span supervision while controlling for other aspects of dataset difficulty, we first recast each dataset in the other's framework, then investigate the performance of two models each of these settings.

To modify Hotpot to be multiple-choice, we

Dataset	HotpotQA-MC	WikiHop-MC
Metric	Accuracy	Accuracy
NoContext	68.01	59.70
MC-BiDAF++	70.01	61.32
MC-MemNet	68.75	61.80
Span2MC-BiDAF++	76.01	59.85

Table 3: The performance of different models on the dev sets of WikiHop and HotpotQA. MC denotes using both the multiple-choice dataset and model. Span2MC means we train the model with span-based supervision and evaluate the model on a multiple choice setting. Our models only mildly outperform the no-context baseline in all settings.

randomly select 9 entities in all of the documents as distractors, and add the answer to make a 10-choice candidates set. To modify WikiHop to be span-based, we concatenate all documents and treat the first appearance of the answer mention as the gold span for training. Any answer occurrence is treated as correct for evaluation.

4.1 Systems to Compare

MemNet Memory networks (Weston et al., 2015) define a generic model class which can gather information from different parts of the passage. Kumar et al. (2016) and Miller et al. (2016) have demonstrated its effectiveness in certain multi-hop settings. These models process a document over several timesteps. On the ith step, the model takes a question representation q_i, attends to the context representation $\mathbf{p}$, gets an attention distribution α_i, computes a new memory cell value $m_i = \sum \alpha_i p_i$, then forms an updated $q_{i+1} = f(m_i, q_i)$. The final memory cell m_T is used to compute a score $s_i = g(m_T, c_j)$ with the jth candidate representation c_j. We modify this architecture slightly using a standard hierarchical attention module (Li et al., 2015).

We can also modify this architecture to predict an answer span – we use the memory cell m_T of the last step, and do a bi-linear product with the context representation $\mathbf{p}$ to compute a distribution over start points $P_{start} = \text{softmax}(\mathbf{p}W_{start}m_T)$ and end points distribution $P_{end} = \text{softmax}(\mathbf{p}W_{end}m_T)$ of the answer span, where W_{start} and W_{end} are two parameter matrix to be learned. We call this Span-MemNet.

BiDAF++ Recently proposed by Clark and Gardner (2018), this is a high-performing model on SQuAD. It combines the bi-directional attention flow (Seo et al., 2017) and self-attention mechanisms. We use the implementation described in Yang et al. (2018).

Dataset	HotpotQA-Span		WikiHop-Span	
Metric	EM	F1	EM	F1
BiDAF++ (Yang+ 18)	42.79	56.19	–	–
Span-BiDAF++	42.45	56.46	24.23	46.13
Span-MemNet	18.75	26.11	13.54	19.23

Table 4: The performance of different models on the dev sets of WikiHop and HotpotQA. Span denotes using both span-based dataset and model. BiDAF++ denotes the performance reported in HotpotQA (Yang et al., 2018).

We can modify this model for the multiple-choice setting as well. Specifically, we use the start P_{start} and end P_{end} distribution to do a weighted sum over the context $\mathbf{p}$ to get a summarized representation $D_{start} = \sum P_{start_i} p_i$, $D_{end} = \sum P_{end_i} p_i$ of the context. Then we concatenate them to do a bilinear dot product with each candidate representation to get the answer score as we described for MemNet. We call this model MC-BiDAF++.

4.2 Results

Table 3 and Table 4 show our results in the two settings. As a baseline on multiple-choice HotpotQA, we also test the no-context baseline, which achieves an accuracy of 68.01%, around 10% absolute higher than on WikiHop. Our candidates were randomly chosen, so this setting may not be quite as challenging as a well-constructed multiple-choice dataset. From Table 3 and Table 4 we draw the following conclusions.

When trained and tested on multiple-choice datasets, our models do not learn multi-hop reasoning. Comparing MC-BiDAF++ and MC-MemNet on the multiple-choice setting of both datasets as shown in Table 3, the models appear to have similar capabilities to learn multi-hop reasoning. However, looking at the no-context baseline for comparison, we find that it is only around 2% lower than the two relatively more complex models. This indicates that much of the performance is achieved by "cheating" through the correlation between the candidates and question/context. Surprisingly, this is true even for HotpotQA, which seems stronger based on the analysis in Table 1.

Span-based data is less "hackable", but models still may not be doing multi-hop reasoning. We then compare the results of Span-BiDAF++

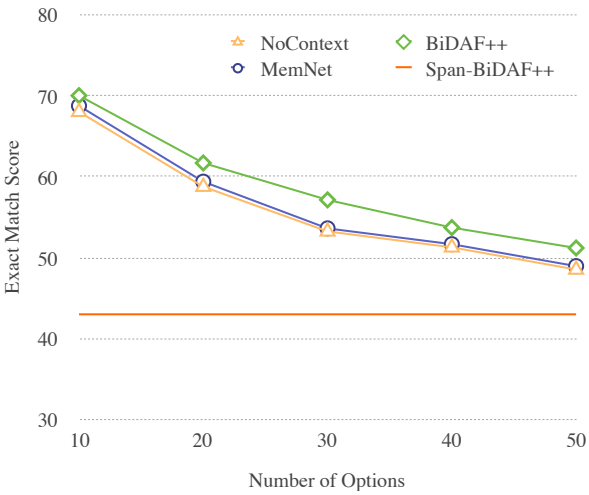

Figure 3: Performance of different options on HotpotQA-MC. Adding more options does not strengthen the model's ability of learning multi-hop reasoning.

and Span-MemNet on the span-based settings of both datasets, which are substantially different from the multiple-choice setting as shown in Table 4. BiDAF++ substantially outperforms the MemNet on both datasets, indicating that BiDAF++ is a stronger model for multi-hop reasoning, despite being less explicitly designed for this task. However, this model still underperforms the Factored BiDAF model, indicating that it could just be doing strong single-sentence reasoning.

Adding more options does not qualitatively change the multiple choice setting. The span-based model requires dealing with a much larger output space than the multiple-choice setting. To test the effects of this, we conduct another experiment by making more spurious options on HotpotQA-MC using the method described in Section 4. The results are shown in Figure 3. As we increase the number of options, we can see that the performance of all models drops. However, even with more options, the no-context baseline can still achieve comparable performance to the other two more complex models, which indicates that these models still aren't learning multi-hop reasoning in such a strengthened setting.

Span-based training data is more powerful. To further understand the two different supervision signals, we conduct another experiment where we train using span-based supervision and evaluate on the multiple-choice setting. Specifically, during evaluation, we select all document spans that map onto some answer candidate, then max over the scores of all spans to pick the predicted answer candidate. The multiple choice options therefore filter the span model's predictions.

From the results in Table 3, we can see that Span2MC-BiDAF++ achieves higher performance compared to MC-BiDAF++ on HotpotQA and nearly comparable performance on WikiHop even with random span selection during training. This shows that with the span-based supervision, the model can learn at least the same thing as the multiple-choice and avoid "cheating" through learning question-candidate correspondences.

5 Discussion and Conclusion

There exist several other multi-hop reasoning datasets including WorldTree (Jansen et al., 2018), OpenBookQA (Mihaylov et al., 2018), and MultiRC (Khashabi et al., 2018). These datasets are more complex to analyze since the answers may not appear directly in the passage and may simply be entailed by passage content. We leave a detailed investigation of these for future work.

For researchers working on the problem of multi-hop reasoning, we think the following points should be considered: (1) Prefer models using span-based supervision to avoid "cheating" by using the extra candidate information. (2) If using multiple-choice supervision, check the no-context baseline to see whether there are strong correlations between question and candidates. (3) When constructing a multi-hop oriented dataset, it would be best to do an adversarial test using a sentence-factored model to see whether multi-hop reasoning is really needed. Both HotpotQA and WikiHop contain good examples for evaluating multi-hop reasoning, but this evaluation is clouded by the presence of easily-solvable examples, which can confuse the learning process as well.

Acknowledgments

This work was partially supported by NSF Grant IIS-1814522, NSF Grant SHF-1762299, a Bloomberg Data Science Grant, and an equipment grant from NVIDIA. The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources used to conduct this research. Results presented in this paper were obtained using the Chameleon testbed supported by the National Science Foundation. Thanks as well to the anonymous reviewers for their helpful comments.

References

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. *ACL*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *Deep Learning workshop at NeurIPS 2014*.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. *ACL*.

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

Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2018. Neural Models for Reasoning over Multiple Mentions using Coreference. *NAACL*.

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.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. *arXiv preprint arXiv:1511.02301*.

Peter Jansen. 2018. Multi-hop Inference for Sentence-level TextGraphs: How Challenging is Meaningfully Combining Information for Science Question Answering? *TextGraphs*.

Peter A Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton T Morrison. 2018. Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. *LREC*.

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

Divyansh Kaushik and Zachary C Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. *EMNLP*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 252–262.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. *ACL*.

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. *EMNLP*.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *EMNLP*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. *EMNLP*.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. In *EMNLP*, volume 3, page 4.

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

Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2017. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1047–1055. ACM.

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*.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What Makes Reading Comprehension Questions Easier? *EMNLP*.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making Neural QA as Simple as Possible but not Simpler. *CoNLL*.

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.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards AI-Complete Question Answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. *EMNLP*.

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. *ICLR*.