FROM EASY TO HARD: TWO-STAGE SELECTOR AND READER FOR MULTI-HOP QUESTION ANSWERING

Xin-Yi Li  Wei-Jun Lei  Yu-Bin Yang

State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
yangyubin@nju.edu.cn

ABSTRACT
Multi-hop question answering (QA) is a challenging task that requires complex reasoning over multiple documents. Existing works commonly introduce techniques such as graph modeling and question decomposition to explore precise intermediate results of multi-hop reasoning, leading to complexity growth and error accumulation. In this paper, we propose FE2H, a simple yet effective framework without extra tasks to address these problems. FE2H is based on our key observation that a standard fine-tuned pre-trained language model (PLM) for QA could achieve strong performance once the input context could be encoded by PLM without truncation. Specifically, a novel two-stage document selector is proposed to generate sufficient context while avoiding input truncation. Additionally, an enhanced reader trained with a two-stage strategy is devised to further boost the performance. Extensive experiments on the popular multi-hop QA benchmark HotpotQA show that despite its simplicity, FE2H achieves competitive results compared to state-of-the-art methods.

Index Terms— Question answering, multi-hop reasoning, natural language understanding

1. INTRODUCTION
Multi-hop question answering is a challenging task where questions are answered by composing information from multiple documents. Recently, many datasets [1, 2, 3] that require multi-hop reasoning over multiple documents have been constructed to drive progress in this field.

Different from single-hop questions whose answers can be derived from a single text span, multi-hop QA cannot achieve the desired performance using techniques that are widely employed in single-hop QA [4]. To achieve better performance, existing studies have proposed many sophisticated modules specially designed for multi-hop reasoning to obtain intermediate results in different forms [5, 6]. Recent studies can be roughly divided into three categories according to the types of intermediate results calculated during reasoning. The first category is the methods that utilize graph neural networks (GNNs) to reason over the constructed graph, and the graph nodes serve as candidate intermediate results [7, 8, 9]. The second category extracts text spans from context as intermediate results, such as the methods that focus on question decomposing [10, 11, 12]. The third category maintains a hidden state, which is updated during each iteration [13, 14].

Despite the success achieved by the above methods, there are still several drawbacks. First, finding fine-grained intermediate results, such as the bridge entities, is typically hard and probably lead to error accumulation. Second, extra errors are inevitably introduced to the QA system by the adopted tasks including named entity recognition, graph modeling, question decomposition, etc. Third, the results are hard to reproduce because of the growth of the complexity caused by these additional tasks.

In previous QA work, it is generally acknowledged that the first step is to obtain contextual token embeddings by feeding the document text to a PLM. If the text exceeds the limit length of the PLM, it will be truncated into segments and encoded separately. Unfortunately, this truncation is a common practice in multi-hop QA since their inputs are often composed of multiple documents. Consequently, the interaction within the context is severely constrained and the capability of PLMs is limited. Experimentally, we observe that a standard PLM-based span prediction model can achieve surprisingly strong performance once the input text contains sufficient information while staying within the length limitation. Therefore, we believe that precise intermediate results are not necessary for better performance. Instead, as much relevant context as possible should be retained before extracting the final answer while keeping the input within the length restriction.

In this paper, we present FE2H, a simple but strong multi-hop QA model that follows a from-easy-to-hard manner based on the above insight. FE2H comprises a selector and a reader that are both directly fine-tuned on PLMs and do not introduce extra techniques or tasks, which reflects their simplicity. Specifically, we explore intermediate results at the document level, which is at a much coarser granularity and is easier to implement compared to previous methods. Furthermore, both the training processes of our selector and reader are divided
into two stages, respectively inspired by the iterative document selection process and the progressive learning custom of humans.

The widely used and representative multi-hop QA benchmark HotpotQA [3] is adopted as our testbed, which requires the QA system to provide the supporting sentences together with the answers. Despite its simplicity, FE2H still achieves strong performance and places third on the leaderboard. Furthermore, comprehensive ablations demonstrate that our two-stage design benefits substantially.

2. RELATED WORK

2.1. Selector for Multi-hop QA

Selectors have been widely applied in multi-document QA to minimize noisy information. While selectors that treat candidates separately are effective for single-hop QA, they are suboptimal for multi-hop QA [7, 15]. Therefore, SAE [16] and S2G [17] apply an MHSA layer to encourage interaction across documents. Since a vanilla classifier is unable to locate all of the supporting documents, SAE and S2G reformulate the classification task into ranking and scoring problems, respectively. However, existing selectors ignore the different roles of the gold documents and select them simultaneously. Therefore, we propose to extract documents sequentially and present a simple but powerful two-stage selector.

2.2. Simple Baselines for Multi-hop QA

Existing sophisticated models specially designed for multi-hop QA have been discussed in Section 1. In recent years, there have been opposing views of whether graph modeling is indispensable. For example, S2G [17] proposes a strong graph-free method and [18] claims that graph attention is a special case of self-attention. There are also several works exploring simple methods for multi-hop QA [19, 20], which are close to our motivation. However, they believe that independent document selection is sufficient for multi-hop QA, while we stress the interaction between documents.

3. PROPOSED FRAMEWORK

In this paper, we adopt the distractor setting of HotpotQA [3] as our testbed, where each question is equipped with two relevant documents and eight distractors. In addition to the exact answer spans, supporting sentences are also provided as explanations to train and evaluate the QA systems. The overview of FE2H is shown in Figure 1. We first iteratively select the two relevant documents and then pass them to the reader to extract supporting facts and answer simultaneously.

3.1. Document Selection Module

The goal of document selection module is to filter the distracting information and generate high-quality context for the following question answering module. Concretely, we aim to select the document that is most relevant to the question, i.e. the hop-1 document. Given the \( q_i \) and its candidate document set \( D_i = \{d_{i,1}, \ldots, d_{i,M}\} \), we construct the sequence as “[CLS] + \( q_i \) + [SEP] + \( d_{i,j} \) + [SEP]” and feed it to PLM, where \( j \in [1, M] \), \( M \) is the size of \( D_i \), and “[CLS]” and “[SEP]” are the special tokens defined by PLM for start and separation. Next, we project the embedding of “[CLS]” and calcu-

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
late the probability that \(d_{i,j}\) is the hop-1 document, which is denoted as \(P(d_{i,j}|q_i)\). For simplicity, we label all of the gold documents as hop-1 documents and optimize the binary cross-entropy loss. After training, the document with the maximum \(P(d_{i,j}|q_i)\) for each \(q_i\) is selected as the predicted hop-1 document and is denoted as \(d_{i,p_{1,j}}\).

For the second stage, we aim to find the hop-2 documents based on the previously selected hop-1 documents. Given question \(q_i\), document set \(D_i\) and the predicted hop-1 document \(d_{i,p_{1,j}}\), the input sequence is generated as “[CLS] + \(q_i\) + [SEP] + \(d_{i,p_{1,j}}\) + \(d_{i,j}\) + [SEP]”, where \(j \in [1, M]\) and \(j \neq p_{1,j}\). Similarly, we denote \(P(d_{i,j}|q_i,d_{i,p_{1,j}})\) as the probability of \(d_{i,j}\) being the hop-2 document and calculate the binary cross-entropy loss using the same labels as the first stage. Document with the highest \(P(d_{i,j}|q_i,d_{i,p_{1,j}})\) is selected as the hop-2 document and denoted as \(d_{i,p_{2,j}}\).

Owing to this iterative selection, our selector is able to effectively identify the gold documents via two normal classification tasks and a simple model structure.

### 3.2. Question Answering Module

In this section, we first introduce a strong baseline reader, and then we present an enhanced two-stage version that further improves performance.

Our baseline reader is a multi-task model that extracts the answer and supporting sentences simultaneously. Its training procedure consists of the following three steps. First, the input for question \(q_i\) is formulated as “[CLS] + yes + no + \(d_{i,p_{1,j}}\) + \(d_{i,p_{2,j}}\) + [SEP] + \(q_i\) + [SEP]”. Second, linear prediction layers are utilized to identify the start and end positions of the answer and the supporting sentences. Third, the training objective is to optimize the above three loss items jointly.

Furthermore, we enhance the model structure and the training strategy of the reader to obtain better performance. Specifically, we add a cross-attention layer to improve the interaction between the context and query. The difference from previous models is that we adopt two cross-attention blocks, including one with layer normalization and one without. As for the training strategy, drawing inspiration from continual learning, our model is first trained on a single-hop QA dataset SQuAD [21] and then transferred to the multi-hop task. The intuition behind this two-stage training procedure is the gap between the pre-trained tasks and multi-hop QA. Hence, directly fine-tuned a PLM on a multi-hop QA task may be too hard and confusing for the models. As a result, the model may answer the multi-hop questions with a single-hop shortcut, namely word-matching the question with a single sentence [22]. Experiments show that our proposed from-easy-to-hard training strategy is a possible remedy for this problem.

### 4. EXPERIMENTS

Two versions of FE2H are trained based on ELECTRA [23] and ALBERT [24], respectively, which are implemented via Transformers [25]. Due to resource limitations, the following ablation studies are conducted on ELECTRA-base.

#### 4.1. Main Results

Table 1 shows the performance of current advanced methods on the HotpotQA test dataset in the distractor setting. FE2H achieves overall competitive performance and places third on the leaderboard. * indicates unpublished works.

| Model                                      | Ans EM | Ans F1 | Sup EM | Sup F1 | Joint EM | Joint F1 |
|--------------------------------------------|--------|--------|--------|--------|----------|----------|
| Baseline Model [3]                         | 45.60  | 59.02  | 20.32  | 64.49  | 10.83    | 40.16    |
| QFE [4]                                    | 53.86  | 68.06  | 57.75  | 84.49  | 34.63    | 59.61    |
| DFGN [7]                                   | 56.31  | 69.69  | 51.50  | 81.62  | 33.62    | 59.82    |
| SAE-large [16]                             | 66.92  | 79.62  | 61.53  | 86.86  | 45.36    | 71.45    |
| C2F Reader [18]                            | 67.98  | 81.24  | 60.81  | 87.63  | 44.67    | 72.73    |
| HGN-large [8]                              | 69.22  | 82.19  | 62.76  | 88.47  | 47.11    | 74.21    |
| AMGN+ [9]                                  | 70.53  | 83.37  | 63.57  | 88.83  | 47.77    | 75.24    |
| S2G+EGA [17]                               | 70.92  | 83.44  | 63.86  | 88.68  | 48.76    | 75.47    |
| SAE+*                                      | 70.74  | 83.61  | 63.70  | 88.95  | 48.15    | 75.72    |
| C2FM with F1 Smoothing*                    | 72.07  | 84.34  | 65.44  | 89.55  | 49.73    | 76.69    |
| PipNet*                                    | 72.26  | 84.86  | 63.71  | 89.41  | 48.76    | 76.95    |
| FE2H on ELECTRA (Ours)                     | 69.54  | 82.69  | 64.78  | 88.71  | 48.46    | 74.90    |
| FE2H on ALBERT (Ours)                      | 71.89  | 84.44  | 64.98  | 89.14  | 50.04    | 76.54    |

Table 1: Performance comparison on HotpotQA test dataset in the distractor setting. FE2H achieves overall competitive performance and places third on the leaderboard. * indicates unpublished works.
Table 2: Document selection results on the dev set of HotpotQA. “-” means the data is not available.

| Model                  | EM   | F1   |
|------------------------|------|------|
| SAE_{large}            | 91.98| 95.76|
| HGN                    | -    | 94.53|
| S2G_{large}            | 95.77| 97.82|
| FE2H on ELECTRA_{base} | 95.53| 97.59|
| FE2H on ELECTRA_{large} (Ours) | **96.32** | **98.02** |

Table 3: Ablation studies of two-stage selector on the dev set of HotpotQA.

| Setting                               | EM   | F1   |
|---------------------------------------|------|------|
| Single-stage + Top 2                  | 89.10| 94.41|
| Single-stage + THOLD                  | 85.10| 87.01|
| Single-stage (Reweight) + Top 2       | 84.17| 91.78|
| Two-stage (Relabel)                   | 92.72| 96.09|
| Two-stage (Ours)                      | **95.53** | **97.59** |

Table 4: Ablation results on components of the reader.

| PLM       | First Stage | Second Stage | Joint F1 | Joint EM |
|-----------|-------------|--------------|----------|----------|
| BERT      | SHD         | MHD          | 68.58    | 41.90    |
|           | MHD         | -            | 67.13    | 39.86    |
|           | SHD + MHD   | -            | 68.37    | 41.51    |
|           | SHD         | SHD + MHD    | 68.23    | 41.89    |
| ELECTRA   | SHD         | MHD          | **75.73**| **49.37**|
|           | MHD         | -            | 75.30    | 48.55    |
|           | SHD + MHD   | -            | 75.43    | 48.87    |
|           | SHD         | SHD + MHD    | 75.55    | 48.48    |

Table 5: Ablation results on training strategies of the reader. “SHD” and “MHD” represent the single-hop dataset SQuAD and the multi-hop dataset HotpotQA, respectively.

4.2. Selector Ablations

Results of the two-stage selector ablations are shown in Table 3. We initially remove the second stage and evaluate the results. The first two rows in Table 3 illustrate that both EM and F1 drop significantly when using the single-stage selector, regardless of whether we select the top 2 documents or the documents whose scores are above a certain threshold.

Recall that we treat the gold documents of each question equally when training our selector. However, SAE [16] and S2G [17] believe that documents with answers are more important and assign them with higher score at training time. To test this assumption, we assign greater weights to documents with answers under the single-stage setting. The results illustrate that performance drops significantly after reassigning the weights. This indicates that directly forcing the selector to identify documents containing the answers is infeasible, and relevant documents are of equal importance.

Furthermore, we attempt another labeling strategy, where documents with answers are used to train the second stage selector and the remaining ones are used to train the first stage selector. Nevertheless, the results also decrease marginally, which may be caused by the inaccurate label criteria.

4.3. Reader Ablations

We first perform ablation studies to investigate the model structure of our proposed reader. Limited by computing resources, the following reader variants are not trained on the single-hop QA dataset in advance. Table 4 shows that performance degrades marginally when our cross-attention layer is removed or modified. Note that the model without the cross-attention layer is equal to the baseline described in Section 3.2, yielding results close to the enhanced version. This is consistent with our finding that a standard fine-tuned PLM could achieve competitive performance once the noisy information is filtered and the context can be encoded without truncation.

Next, to extensively demonstrate the benefits brought by our two-stage training strategy, ablations are performed with two PLMs, namely BERT-base [26] and ELECTRA-large [23]. Table 5 illustrates that, compared to simply training on the multi-hop dataset, our two-stage strategy brings significant performance gains under both two settings, especially for BERT. Specifically, joint F1 and EM respectively drop 1.45% and 2.04% when the reader is only trained on the multi-hop QA dataset. We presume that this is because small PLMs could learn more task-specific knowledge through the auxiliary task in advance, while large PLMs with stronger abilities may benefit less. Consequently, we recommend our two-stage reader, especially under circumstances where resources are limited and only small PLMs are permitted.

Additionally, different two-stage training strategies that also leverage both single-hop and multi-hop QA datasets are attempted. The performance degradations of these experiments demonstrate the superiority of our proposed sequential training approach.

5. CONCLUSION

In this paper, we propose a simple yet effective framework for multi-hop QA that is free of extra tasks. We find that after encoding the filtered context using PLM without truncation, simply adding a prediction layer could achieve competitive results. Therefore, we emphasize the importance of document selection and propose a powerful two-stage selector. To further improve performance, we introduce an enhanced reader trained with a two-stage strategy. We hope this work could facilitate simpler and more powerful multi-hop QA approaches with the help of current advanced PLMs.
6. REFERENCES

[1] Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel, “Constructing datasets for multi-hop reading comprehension across documents,” TACL, vol. 6, 2018.

[2] Alon Talmor and Jonathan Berant, “The web as a knowledge-base for answering complex questions,” in NAACL-HLT, 2018.

[3] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning, “HotpotQA: A dataset for diverse, explainable multi-hop question answering,” in EMNLP, 2018.

[4] Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita, “Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction,” in ACL, 2019.

[5] Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov, “Neural models for reasoning over multiple mentions using coreference,” in NAACL.

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

[7] Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu, “Dynamically fused graph network for multi-hop reasoning,” in ACL, 2019.

[8] Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu, “Hierarchical graph network for multi-hop question answering,” in EMNLP, 2020.

[9] Ronghan Li, Lifang Wang, Shengli Wang, and Zejun Jiang, “Asynchronous multi-grained graph network for interpretable multi-hop reading comprehension,” in IJCAI, 2021.

[10] Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D. Manning, “Answering complex open-domain questions through iterative query generation,” in EMNLP-IJCNLP, 2019.

[11] Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi, “Multi-hop reading comprehension through question decomposition and rescoring,” in ACL, 2019.

[12] Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela, “Unsupervised question decomposition for question answering,” in EMNLP, 2020.

[13] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum, “Multi-step retriever-reader interaction for scalable open-domain question answering,” in ICLR, 2018.

[14] Yair Feldman and Ran El-Yaniv, “Multi-hop paragraph retrieval for open-domain question answering,” in ACL, 2019.

[15] Sewon Min, Victor Zhong, Richard Socher, and Caiming Xiong, “Efficient and robust question answering from minimal context over documents,” in ACL, 2018.

[16] Ming Tu, Bowen Zhou, et al., “Select, Answer and Explain: Interpretable Multi-hop Reading Comprehension over Multiple Documents,” in AAAI, 2020.

[17] Bohong Wu, Zhousheng Zhang, and Hai Zhao, “Graph-free Multi-hop Reading Comprehension: A Select-to-Guide Strategy,” arXiv:2107.11823, 2021.

[18] Nan Shao, Yiming Cui, Ting Liu, Shijin Wang, and Guoping Hu, “Is Graph Structure Necessary for Multi-hop Question Answering?,” in EMNLP, 2020.

[19] Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer, “Compositional questions do not necessitate multi-hop reasoning,” in ACL.

[20] Dirk Groeneveld, Tushar Khot, Mausam, and Ashish Sabharwal, “A simple yet strong pipeline for HotpotQA,” in EMNLP, 2020.

[21] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang, “SQuAD: 100,000+ questions for machine comprehension of text,” in EMNLP, 2016.

[22] Yichen Jiang and Mohit Bansal, “Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA,” in ACL, 2019.

[23] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning, “ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators,” in ICLR, 2020.

[24] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut, “ALBERT: A lite bert for self-supervised learning of language representations,” in ICLR, 2020.

[25] Thomas Wolf, Alexander M. Rush, et al., “Transformers: State-of-the-art natural language processing,” in EMNLP, 2020.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in NAACL-HLT, 2019.