Ana-QA: Multi Hop Question Answering through DecompRC and DrQA

Aaron Beomjun Bae
UC Irvine
aaron.bae@uci.edu

Apoorva Muthineni
UC Irvine
amuthine@uci.edu

Nishitha Suvarna
UC Irvine
nssuvarn@uci.edu

Abstract

This is a replication study where we combine two papers, DecompRC (Sewon Min et al., 2019) and DrQA (Danqi Chen et al., 2018), and present a new Question-Answering (QA) system for multi-hop questions via question decomposition. The difference between our-model and the DecompRC model is that we replace the BERT-based single-hop QA system with the one based on DrQA Document Reader. DrQA reader module is different from BERT-based system as it uses a multi-layer recurrent neural network. As we anticipated, Ana-QA underperformed compared to DecompRC. We describe our system pipeline and share our analysis on the results.

1 Introduction

With the breakthroughs in vector representations of sentences and recurrent neural networks, question answering (QA) systems have made big strides in answering simple reading comprehension questions. For example, questions like "who is the president of the United States?" can easily be answered by providing the state of the art QA system with a Wikipedia article on President Trump. The tougher challenge at the horizon, now, is the ability to answer multi-hop questions. As shown in Table 1, Multi-hop questions require the ability to reason through multiple documents, often needing to manage intermediary answers throughout. This type of question is common in our natural conversations, and, therefore, is a crucial next step in developing an intelligent QA system.

Existing approaches on multi-hop QA systems, like GNNs, take entities as graph nodes and achieve reasoning by passing messages over the nodes set of target entities. (Welbl, Stenetorp, and Riedel, 2018). However, for complicated questions like multi-hop, performing reasoning on entity graphs often result in accumulated inaccuracies. Furthermore, the need to construct a comprehensive graph structure for a large domain is often challenging, if not unrealistic. On the other hand, our approach does not have these dependencies. Instead, we focus on answering questions by question decomposition.

Q Which team does the player named 2015 Diamond Head Classic's MVP play for?
P1 The 2015 Diamond Head Classic was . . . Buddy Hield was named the tournament's MVP.
P2 Chavano Rainier Buddy Heild is a Bahamian professional basketball player for the Sacramento Kings . . .
Q1 Which player named 2015 Diamond Head Classic's MVP?
Q2 Which team does `[ANS]` play for?

Table 1: An example of a multi-hop question along with supporting paragraphs and decomposed single-hop questions. Figure from (Sewon Min et al., 2019).

In this project, we aim to develop a multi-hop QA system by combining DecompRC's(Sewon Min et al., 2019) question decomposition method along with DrQA's(Danqi Chen et al., 2018) efficient single-hop question answering engine. Diverging from the traditional graphical approach, DecompRC involves breaking down complex multi-hop questions into simpler single-hop questions. Then, we will answer the decomposed questions using DrQA, which is a recurrent neural network based QA system.

We use F1 score, exact match (EM) score, precision and recall to evaluate our model on HotpotQA dataset. Just as we anticipated, however, the results from the original DecompRC using BERT (Devlin et al., 2019) performed better than our model, which uses DrQA instead. However, the possible factors that might have caused the performance discrepancy is outlined, and we suggest a few different ways to improve this DrQA-based model.

2 Related Work

2.1 QA Datasets

QA system development research has recently caught a trend, and many datasets have been collected and made available online. SQuAD (Rajpurkar et al., 2016) is one of the most commonly cited because of its clear file structure. It consists of a question, an answer, and a context paragraph that contains the exact answer. This structure is to allow QA systems to predict an answer span from the given relevant document, rather than formulate a novel answer. Another often cited dataset is SearchQA (Dunn et al., 2017), which differs from SQuAD in that it was generated by collecting relevant context paragraph to existing question-answer pairs, not the other way around. This allows for more natural questions that a person might ask.

However, most of the prevalent datasets focus on single hop Question-Answering. The focus of our work is on multi-hop QA datasets which require reasoning over multiple paragraphs to answer the question. Thankfully, there are two main datasets for multi-hop questions: WikiHop (Welbl et al., 2018) and HotpotQA (Yang et al., 2018). WikiHop is constructed using existing knowledge bases or structured databases, and as a result, all the answers to the questions are single entities that existed in the knowledge base. On the other hand, HotpotQA consists of hand-written questions along with hand-picked Wikipedia articles as its context. We found the diversity in HotpotQA valuable, and consequently, in our project, we only focus on HotpotQA.

2.2 Multi-hop Reading Comprehension

Not only are there previous research on multi-hop QA datasets, there are also previous research on the QA systems themselves. Most of these developed multi-hop QA systems can be largely categorized into 3 types: ones that use knowledge graphs, attention based neural networks, and decomposition algorithm. Knowledge graphs are helpful in developing QA systems because it can graphically show the entities that it utilized to reach the answer. A successful example of this would be Xia et al's graph neural networks (Xiao et al., 2019). Secondly, attention based mechanisms (Ming et al., 2020) are also effective in answering multi-hop questions because by its architecture, it is able to extract the most relevant sentence amidst other distracting sentences (Jiang et al., 2019). Lastly, Decomposition is also successful because it strives to convert the multi-hop question into single-hop sub-questions (Nishida et al., 2019; Min et al., 2018). Since research on single-hop questions are abundant, we anticipated that the high performance in answering single-hop question will translate to multi-hop decomposition answering. Therefore, we decided to further look into this third approach in solving multi-hop questions.

To note some of the more recent findings, here are 3 new approaches: Query Focused Extractor, DFGN, and the Select, Answer, and Explain models. The Query Focused Extractor(QFE) model (Nishida et al., 2019) regards the evidence extraction as a query-focused summarization task, and reformulates the query in each hop. The DFGN model (Xiao et al., 2019) constructs a dynamic entity graph based on entity mentioned in the query and documents and this is iterated for multiple rounds to achieve multi hop reasoning. The Select, Answer and Explain(SAE)(Ming et al., 2020) model first filters the answer-unrelated documents and only the relevant documents are fed to a model which is optimized with a multi-task learning objective on both token and sentence levels for answer and supporting sentences prediction, together with an attention-based interaction between the two tasks.

Our approach in using DecompRC and DrQA is different from all previous models described because the specific pairing of the systems have never been done before. DecompRC uses a BERT-based decomposition model to predict the sub-questions, and DrQA uses a LSTM-based RNN architecture to predict the answer to those sub-questions. These two systems appear to have success in their own respective domains of tasks, so we decide to combine them to evaluate their performance.

3 Approach

In this section, we describe the details of our data set and the pipeline we developed to make the comparison between BERT-based DecompRC and DrQA-based DecompRC.

3.1 Data filtering

While exploring HotpotQA dataset, we noticed that it consists of two types of question types: bridge and comparison. Yang et al states that bridge type questions have a bridge entity that links two questions together whereas comparison type questions

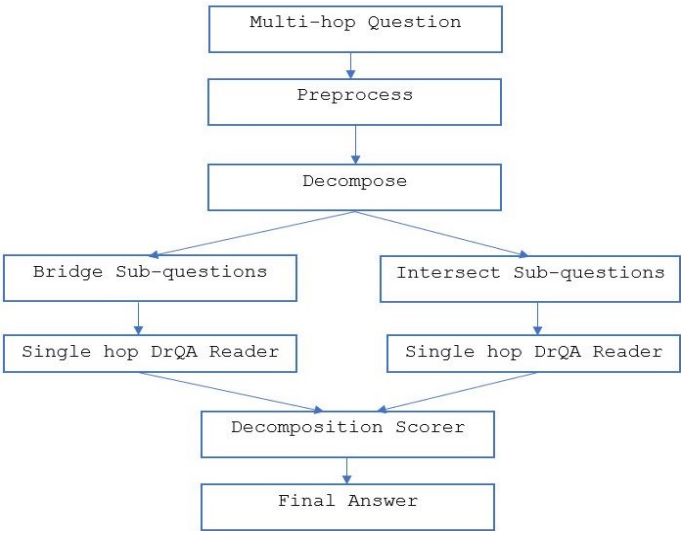

Figure 1: The overall diagram of how a multi-hop question is answered through DecompRC and DrQA in Ana-QA

have two entities that share a particular property (Yang et al., 2018). Since DrQA does not inherently have an understanding of logical equivalence, we assumed that the comparison type questions will be too difficult for DrQA to solve. So, we only use the bridge type questions in this project. It should be noted that in DecompRC paper, this bridge type questions are further divided into “bridge" and “intersection reasoning" types. There is a subtle difference between the two types, but within the context of our project, we only need to acknowledge the fact that the decomposition result is different for a given question. An example of this difference is shown in the Figure 2 below.

3.2 Three Step Pipeline

The implementation of our DrQA-based model mirrors DecompRC’s three step process very closely (Sewon Min et al., 2019). As shown in the pipeline depicted in Figure 1, the first step consists of preprocessing and decomposition of the question. During the preprocessing step, we filter out the comparison type questions from HotpotQA, as per reason exalained in the previous section “Data Filtering". During the decomposition, we reformulate the decomposition problem into another span prediction problem, since all it needs to do is create two different spans from the question that represents two different single-hop questions. This process is carried out by Pointer model from DecompRC, which maps the question to the set of indices in the an input sequence and returns the indices having the highest probability.

In the second step, we use a single-hop RC model (DrQA) to answer the sub-question for each type. The input to the single-hop RC model would be a sub-question span created by the previous Pointer model and the 10 contextual paragraphs from HotpotQA. For the given question in Figure 2, both bridge and intersection question decomposition results in “Chief of Protocol" as the answer being returned by the single hop RC model, however when the given multi-hop question is considered as a whole then it results in ’united states ambassador’ as the answer. Note that the pretrained DecompRC uses a BERT based model (Devlin et al., 2019) for this step. This is where our approach differs from the original DecompRC model as we seek to use a different single- hop RC model (DrQA).

The final step is basically a selection step. It scores the decomposition generated for each reasoning type and compares which is better. Whichever decomposition the scorer deems better, it gives its answer as the final answer. The scorer in this process uses BERT to encode an input sequence representing the given question, its reasoning type, the predicted answer, and the evidence for that reasoning type. Note that the evidence is the entire sentence where the answer span was retrieved. As shown in Figure 2, the final answer to our example problem is ’Chief of Protocol’, because the score for intersection type decomposition resulted in a higher score than bridge type decomposition.

3.3 Implementation Workarounds

To make a correct performance comparison between the original DecompRC and our DrQA ver-

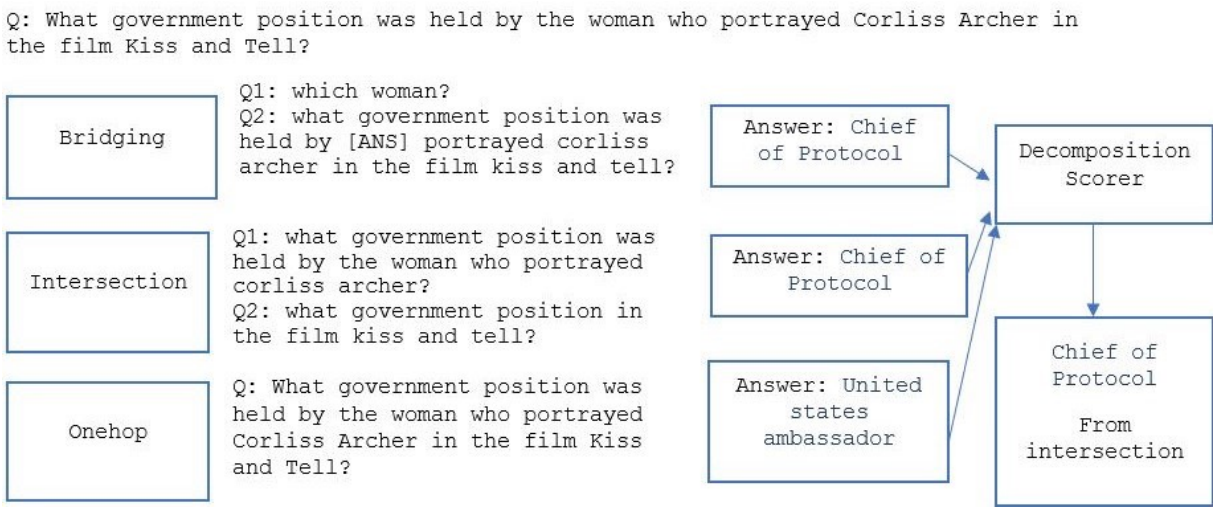

Figure 2: Given the question, our model decomposes the question through both the bridge and intersection reasoning types. Then, each sub-question interacts with Dr.QA and produces the answer. Lastly, the decomposition scorer decides which answer will be the final answer.

sion, we needed to implement and intertwine both system's pipeline. The original DecompRC model was easily implemented since the code base was readily available. However, to implement our model, we had to take DecompRC's output file that contained the decomposed questions and feed them to DrQA's Reader module.

This process was surprisingly challenging because of few implementation issues. First, the available DrQA code base only supports single paragraph as the context for each question. This was a significant roadblock since HotpotQA contains 2 gold paragraphs and 8 distractor paragraphs for each question. To get around this problem, we combine all the 10 paragraphs into one single paragraph and feed it to the DrQA reader. Second, the decomposition scorer requires the evidence span to be part of the encoded input. However, this is not an easy task to find the entire span post-prediction, because the output file from Dr.QA seems to output only the predicted answer span. Therefore, we had to input an arbitrary token "UNKNOWN" in the place of evidence attribute.

4 Experiments

4.1 Evaluation Metrics

Following the DecompRC paper, we use F1 score, Exact Match(EM), Precision and Recall scores to analyse the models. All the metrics ignore punctuation and article tokens (a, an, the).

Exact Match(EM) This metric measures the percentage of predictions that match the ground truth answer exactly.

Precision This metric measures the overlap between the predicted and ground truth answer against the length of predicted answer. We take the prediction and ground truth answers as a bag of tokens and compute their precision.

Recall This metric measures the overlap between the predicted and ground truth answer against the length of ground truth answer.

F1 score This metric measures the average overlap between the predicted and ground truth answers.

4.2 Replicated DecompRC

This model simply is the recreated results by using the available code base. The only difference form the results from the DecompRC paper is that it is evaluated only on the filtered HotpotQA dataset, which only have bridge type questions. This verifies the results claimed in the original paper, since we observe that the generated F1 score (72.024) matches closely with that of the one claimed in the paper(72.53)

4.3 Baseline Model

This baseline model uses DecompRC with DrQA but without the default scorer from DecompRC. Instead, during the selection process, we select the answer with a higher confidence score provided by DrQA. Note that without the default scorer from DecompRC, we perform much poorer than the original DecompRC, because selection process does not take into account the reasoning type and the evidence of the predicted answer. As expected,

Model	F1	Precision	Recall	EM
DecompRC(original paper scores with Bert based single-hop)*	72.53	-	-	-
DecompRC (our evaluation scores with Bert based single-hop)	72.074	75.077	73.418	56.185
DecompRC+DrQA without decomposition scorer (Baseline)	9.27	-	-	5.84
DecompRC+DrQA with one answer candidate	28.731	31.375	29.718	17.414
Ana-QA(DecompRC+DrQA with 3 best answer candidates)	30.948	33.101	32.500	18.515

Table 2: Results showing F1, Precision, Recall and EM scores of all the models. * indicates original paper results. The original paper don't have Precision, Recall and EM scores, these scores are marked with -

we observe that this model resulted in an F1 score of only 9.27 compared to that of 72.074 from the original implementation. The following models, however, show an improvements in closing the gap in performance.

4.4 Baseline + Scorer

This model is the full implementation of DecompRC combined with DrQA with the scoring and selection process carried out exactly like DecompRC. We observe that this approach has shown a significant performance improvement from the baseline model without the scorer. It achieved 28.731 in F1 and 17.414 in EM score whereas our baseline scored 9.27 in F1 and 5.84 in EM. This experiment shows us that the decomposition selection algorithm from DecompRC has a significant impact on the overall end-to-end QA system.

4.5 Baseline + Scorer + 3 best answer candidates(Ana-QA)

From the results of the model with the DecompRC's scorer, we realized that the best answer candidate from DrQA might not always be correct. In fact, because the 8 of the 10 context documents are technically a "distractor" paragraphs, often times DrQA selects an incorrect answer. Also, we observe that the actual answer is often is not the highest scoring but somewhere between the first and the third answers. To take advantage of this observation, we created a new model that uses the top 3 answer candidates from DrQA and derives all 6 possible answers: 3 from intersection and 3 from bridge type decomposition. At the end, the scorer system selected the best solution from these 6 candidate answer choice, which resulted in an improved result. This model observed 2% performance improvement over both F1 and EM, resulting in 30.948 F1 and 18.515 EM scores.

4.6 Performance Analysis

From these results, we learn 3 important aspects of QA systems. The first is that we identified that the performance is greatly influenced by the selection process. The staggering difference in results between the base line model with or without the proper scoring function was over 19 points in F1. This tells us that neither decomposition is perfect but with the correct selection process, the end-to-end QA system is able to answer the multi-hop question with higher accuracy than with single type of decomposition.

The second point we conclude is that the small implementation details of a QA system is critical to the overall performance. More specifically, details like providing "UNKNOWN" as the evidence and combining the 10 documents into a single super-document has a significant reduction in the performance. To illustrate this point, we analyze a specific question example provided in Table 3. Here, the DecompRC is able to successfully decompose the bridge type question Q into meaningful sub-questions Q1 and Q2. Subsequently, both the models predict right answer for Q1 but Ana-QA fails to answer Q2 correctly. We attribute this to the fact that we provided "UNKNOWN" token in place of the actual evidence, since all other variables are controlled. This type of inner-question-level errors are repeated throughout the project, resulting in 40 points decrease in F1 score between Ana-QA and DecompRC. Therefore, we note that for the best possible performance of QA system, these small details will cause significant performance decrease.

Also, another detail that potentially is causing a big decrease in performance is the super-documents. When we are feeding in the context documents to DrQA, we provide a concatenation of the 10 documents instead of a single relevant document. This seems to be a distraction for DrQA, as it

Q:The arena where the lewiston maineiacs played their home games can seat how many people?	DecompRC	Ana-QA
Sub-question1 (Bridge)	**Q1**:which arena where the lewiston maineiacs played their home games	**Q1**:which arena where the lewiston maineiacs played their home games
Answer1	androscoggin bank colisée	androscoggin bank colisée
Evidence1	the lewiston maineiacs were a junior ice hockey team of the quebec major junior hockey league based in lewiston, maine. the team played its home games at the **androscoggin bank colisée**. they were the second qmjhl team in the united states, and the only one to play a full season	UNKNOWN
Sub-question2 (Bridge)	**Q2**:androscoggin bank colisée can seat how many people?	**Q2**:androscoggin bank colisée can seat how many people?
Answer2	3,677	4,000
Evidence2	the androscoggin bank colisée (formerly central maine civic center and lewiston colisee) is a 4,000 capacity (**3,677** seated) multi-purpose arena, in lewiston, maine	UNKNOWN
Final Answer	3,677	4,000

Table 3: Qualitative performance comparison of DecompRC vs Ana-QA

was originally trained with the assumption that the entirety of the document is relevant to the question. With additional time allotted for this project, we expect to achieve results closer to that of the original DecompRC model by managing these details better.

Lastly, we attribute the biggest factor in the performance difference between DecompRC and Ana-QA to the difference in architecture. The two systems are different in that the first uses a bidirectional neural network architecture, BERT, whereas the second uses a single directional architecture, LSTM. We conclude that the small difference in performance in the single-hop level are translating to a bigger performance discrepancy when inserted into DecompRC's multi-hop question answering system. As foreseen before, the bottleneck to our QA system's performance was the accuracy in the single hop model, and we seem to be able to make progress if we were to develop a better single-hop question answering system.

5 Conclusions and Future Work

From this project, we were able to create a novel multi-hop QA system by combining DecompRC and DrQA. Despite small increase in performance from the baseline, Ana-QA ultimately fell short to DecompRC, because of the fundamental differences between BERT and DrQA along with some implementation differences as discussed in section 3.3. If we had more time, we plan to fix the implementation issues by extending DrQA to handle multiple paragraphs as context or look into another replacement for the single-hop question answering engine.

For future work, we suggest a feature expansion using DrQA's retriever module. By adding the retriever module to our current QA system, we will be able to answer multi-hop questions without the need for a context documents. This will allow the system to be truly domain free and let it be applied to a broader set of domain knowledge.

6 References

1 Sewon Min et al.“Multi-hop Reading Comprehension through Question Decomposition and Rescoring”. In: arXiv preprint arXiv:1906.02916(2019).

2 Zhilin Yang et al.“HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering”. In: arXiv preprint arXiv:1809.09600(2018)

3 Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. "Reading Wikipedia to answer open domain questions". In:arXiv:1809.09600(2018).

4 Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. 2019. Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction. *In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy*, July 28- August 2, 2019, Volume 1: Long Papers, pages 2335–2345.

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

6 Matthew Dunn, Levent Sagun, Mike Higgins, Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A new Q&A dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

7 Xiao, Y.; Qu, Y.; Qiu, L.; Zhou, H.; Li, L.; Zhang, W.; and Yu, Y. 2019. Dynamically fused graph network for multihop reaso ning. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6140–6150

8 Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, Bowen Zhou. 2019. "Select, Answer and Explain: Interpretable Multi-hop Reading Comprehension over Multiple Documents". In: arXiv:1911.00484v4

9 Yichen Jiang, Mohit Bansal. 2019. "Self-Assembling Modular Networks for Interpretable Multi-Hop Reasoning".In: arXiv:1909.05803v2

10 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language under- standing. In NAACL.