Interpretable AMR-Based Question Decomposition for Multi-hop Question Answering

Zhenyun Deng, **Yonghua Zhu**, **Yang Chen**, **Michael Witbrock**, **Patricia Riddle**
School of Computer Science, University of Auckland, New Zealand
{zden658, yzhu970}@aucklanduni.ac.nz, {yang.chen, m.witbrock, p.riddle}@auckland.ac.nz

Abstract

Effective multi-hop question answering (QA) requires reasoning over multiple scattered paragraphs and providing explanations for answers. Most existing approaches cannot provide an interpretable reasoning process to illustrate how these models arrive at an answer. In this paper, we propose a Question Decomposition method based on Abstract Meaning Representation (QDAMR) for multi-hop QA, which achieves interpretable reasoning by decomposing a multi-hop question into simpler sub-questions and answering them in order. Since annotating the decomposition is expensive, we first delegate the complexity of understanding the multi-hop question to an AMR parser. We then achieve decomposition of a multi-hop question via segmentation of the corresponding AMR graph based on the required reasoning type. Finally, we generate sub-questions using an AMR-to-Text generation model and answer them with an off-the-shelf QA model. Experimental results on HotpotQA demonstrate that our approach is competitive for interpretable reasoning and that the sub-questions generated by QDAMR are well-formed, outperforming existing question-decomposition-based multi-hop QA approaches.

1 Introduction

Multi-hop question answering (QA) has long been a grand challenge in Artificial Intelligence [Chen *et al.*, 2020; Yang *et al.*, 2018b]. General solutions could benefit from interpretability, i.e., providing evidence for the answer by mimicking human reasoning in answering a multi-hop question. Such evidence is usually embodied in key information scattered in multiple paragraphs of the text. To endow a multi-hop QA model with better interpretability, it is desired to capture clues from the given questions and subsequently extract correct pieces of evidence based on these clues.

Currently, most multi-hop QA approaches achieve post-hoc-interpretable reasoning by constructing an auxiliary model to provide explanations for an existing QA model, *e.g.,* [Qiu *et al.*, 2019; Tu *et al.*, 2020; Fang *et al.*, 2019] perform interpretable multi-hop reasoning using a supporting-evidence prediction model. As a result, methods of this type usually cannot employ a context-specific reasoning process for answering multi-hop questions; this disconnect between reasoning and explanation lessens the extent to which humans can understand the behaviour of the machine reasoning models and therefore how these models might be improved. In contrast, methods based on intrinsically interpretable reasoning may better help humans to understand and trust the mechanism of a QA system using multi-hop reasoning, because they construct a self-explanation system and incorporate interpretability directly into that system.

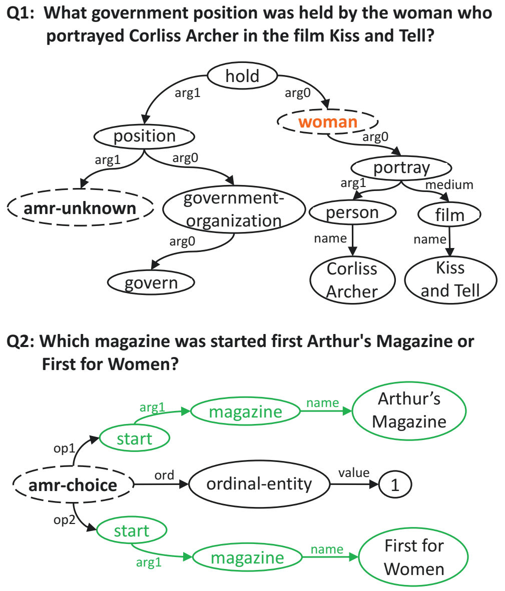

Figure 1: Examples of delegating the complexity of understanding multi-hop questions to AMR parsers. Q1 is a bridging multi-hop question; these can be interpreted by identifying the secondary "unknown" variable *woman*. Q2 is a comparison question; these can be interpreted by capturing the longest identical path.

Divide-and-conquer is a promising strategy for intrinsic interpretability, in which complex multi-hop questions are decomposed into simpler sub-questions that can be answered by an off-the-shelf QA model. The simplified sub-questions may help humans to understand the logic of complex multi-hop questions better, and also to diagnose where QA models fail. Since annotations for decomposition are expensive and time-consuming, existing studies have trained a question decomposition (QD) model using a combination of hand-crafted labelled examples and a rule-based algorithm, or have trained a pseudo decomposition model to extract spans of text as sub-questions [Min *et al.*, 2019; Perez *et al.*, 2020]. However, the resulting sub-questions are often ungrammatical, may not even be explicit questions, or require additional language processing, leading to errors in downstream prediction tasks.

In a multi-hop question, predicate-argument structure plays a key role in understanding its meaning. Utilizing predicate-argument structures to help understand natural language questions has become a common approach [Rakshit and Flanigan, 2021]. In the explicit graph structure of AMR, the predicate-argument structure appears in the form of nodes. Recently, AMR parsers have made great progress in reducing the complexity of understanding natural language questions by improving their semantic graph representations [Ribeiro *et al.*, 2020], benefiting downstream tasks such as summarization [Liao *et al.*, 2018] and question answering [Xu *et al.*, 2021]. Furthermore, AMR-to-Text generation models have been developed in recent years that produce well-formed text by encoding the AMR graph as an input to pretrained language models [Bevilacqua *et al.*, 2021; Ribeiro *et al.*, 2021].

Motivated by this, we propose to perform question decomposition based on AMR for multi-hop QA. Our method is designed for four goals: i) delegating the complexity of understanding multi-hop questions to AMR parsing; as shown in Figure 1 we understand multi-hop questions better by identifying the intermediate unknown variable or longest identical path; ii) segmenting the AMR graph according to the required type of reasoning; iii) generating sub-questions by AMR-to-Text generation and sorting them into a logical order, *e.g.,* based on primary and secondary unknowns; iv) answering sub-questions and providing an interpretable reasoning process that can form a basis for explanation.

The contributions of this paper are summarised as follows:

- We transform the task of decomposition of the multi-hop question into the task of segmentation of the corresponding AMR graph.
- We propose two different AMR graph segmentation methods for multi-hop QD according to the reasoning type required: 1) unknowns-based and 2) path-based graph segmentation.
- We propose a pipeline-based modular method that integrates multiple modules, *i.e.,* AMR parsing, AMR graph segmentation and AMR-To-Text generation, for multi-hop QD and multi-hop QA, which may help humans to better understand the reasoning behaviors of an interpretable multi-hop QA model.

2 Related Work

Intrinsic Interpretability. Intrinsically interpretable QA models incorporate interpretability directly into the structure of QA systems, making them self-explanatory. Recent studies have used divide-and-conquer, simplifying multi-hop questions to achieve self-explainable reasoning. DecompRC [Min *et al.*, 2019] decomposes complex questions into several sub-questions and uses a scoring model to jointly select the final answer and the most appropriate decomposition. OUNS [Pan *et al.*, 2020] creates a pseudo-decomposition model with unsupervised seq2seq learning to map a hard question to many simple questions, and builds a recomposition model that combines answers of those simple questions to obtain the final answer. Unlike the above methods, QDAMR decomposes a multi-hop question based on the high-level semantic relations between the concepts in its AMR parse graph, and answers sub-questions in a logical, and potentially explanatory, order.

Post-hoc Interpretability. Post-hoc-interpretable QA models provide explanations for multi-hop reasoning by creating second (supporting evidence) models, which fall into two categories. In one, the supporting evidence prediction model is constructed independently: AISO [Zhu *et al.*, 2021] defines three types of retrieval operations to find the missing evidence at each reasoning step. In the other category a multi-task model achieves answer prediction and supporting evidence prediction simultaneously. HGN [Fang *et al.*, 2019], Entity-GCN [De Cao *et al.*, 2018] and DFGN [Qiu *et al.*, 2019] fuse multi-granularity information in a graph, and then apply a multi-task prediction model over the graph to achieve both answer and evidence prediction.

AMR for Multi-hop QA. AMR parsers have made great progress in recent years as a source of explicit graph structure for conducting symbolic reasoning [Mitra and Baral, 2016]. AMR-SG [Xu *et al.*, 2021] constructs an AMR semantic graph from valid evidence items, and reasons over it to obtain interpretability. NSQA [Kapanipathi *et al.*, 2021] proposes a Neuro-Symbolic QA system based on AMR that forms logical queries and applies a neuro-symbolic reasoner to predict the final answer. Unlike the above forms of symbolic reasoning, we use AMR to convert the multi-hop question to symbolic form in support of question decomposition and identification of intermediate unknowns.

3 Method

3.1 Overview

In this section, we describe interpretable question decomposition based on AMR, for multi-hop QA. As illustrated in Figure 2, the pipeline of our proposed QA system consists of four modules: i) AMR Parsing: utilizes AMR parsers to transform the multi-hop questions into an AMR graph; ii) AMR graph segmentation: segments the AMR graph into simpler AMR sub-graphs according to the reasoning type of the question; iii) Sub-question generation: applies AMR-to-Text generation model to sub-graphs to generate sub-questions; iv) Single-hop QA model: answers sub-questions with off-the-shelf QA models.

3.2 AMR Parsing

Multi-hop QA requires multiple steps of reasoning over multiple scattered paragraphs to arrive at an answer. In order to help humans understand how these models make decisions, it is useful to capture intermediate unknown variables in the multi-hop question. To do this, we delegate the complexity

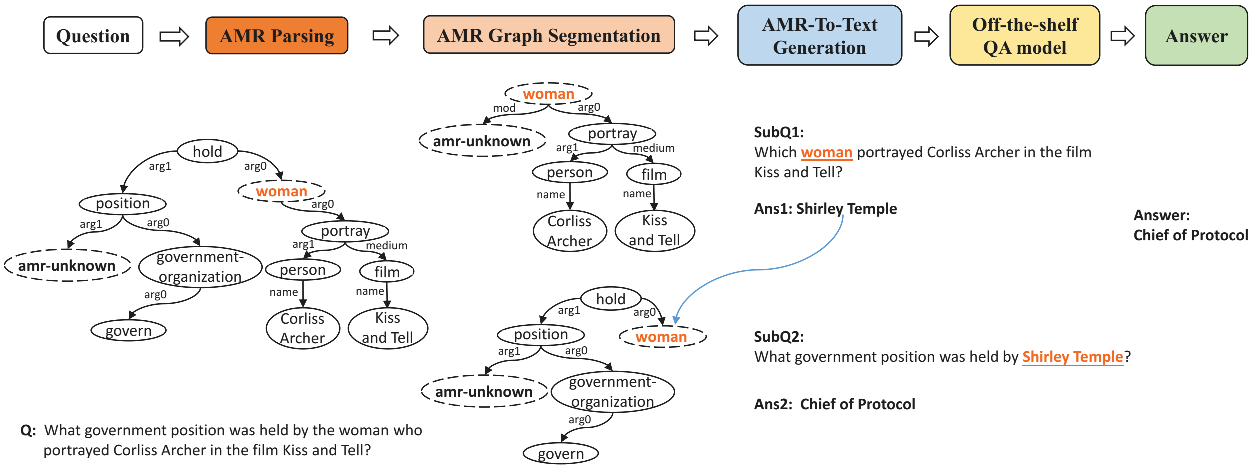

Figure 2: An overview of our proposed QDAMR framework for multi-hop QA. QDAMR first parses question Q, *What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell?*, into an AMR graph; it then segments the graph based on the secondary unknown variable *woman*; next it applies AMR-to-Text generation to generate well-formed sub-questions, subQ1 and subQ2; after that it uses off-the-shelf QA to predict the answer of the secondary sub-question subQ1; and finally QDAMR substitutes that answer into the unknown *woman* in the primary sub-question to predict the final answer, again via off-the-shelf QA.

of understanding the multi-hop question to an AMR graph. As shown in Figure 2, we parse a multi-hop question Q into an AMR graph, where nodes represent concepts and edges represent relations in between. An *amr-unknown* node denotes a primary unknown concept that represents the answer to the given question Q. In the example of Figure 2, *amr-unknown* is an AMR *position* which represents the type of the answer. Furthermore, the AMR graph annotations on Q can also help identify intermediate unknown variables needed for multi-hop reasoning, *e.g., woman* is a secondary unknown.

We exploit transfer learning, using a pretrained model based on BART [Lewis *et al.*, 2019] for AMR parsing to a linearized AMR graph [Bevilacqua *et al.*, 2021]. Its advantage is that it overcomes the data sparsity issue in seq2seq-based methods. Since the linearized graph output is fully graph-isomorphic, we encode the question into a graph without losing adjacency information.

3.3 AMR Graph Segmentation

The key challenge of question decomposition is that it is difficult to obtain annotations for the decomposition. We therefore propose to transform decomposition of the multi-hop question into a corresponding AMR graph segmentation operation. The advantage of AMR graph segmentation is that it can be done without linguistic expertise and without training. Multi-hop questions can be divided into the following categories according to the reasoning type: bridging, intersection and comparison. Thus, we propose two graph segmentation methods across the three reasoning types, *i.e.,* unknown-based and path-based AMR graph segmentation, which we describe in the following subsection.

Unknowns-based Graph segmentation. Bridging questions require asking for the answer to an entity that is not explicitly mentioned. Therefore, the key to answering the question is how to identify the secondary unknown entity. According to [Rakshit and Flanigan, 2021], each predicate-argument relationship at the sentence level can be expressed as a question-answer pair. Motivated by this, we propose using unknowns-based graph segmentation for bridging questions. We first extract all predicate nodes, then take subject nodes connected to the predicate nodes as candidate unknowns, and finally segment the AMR graph based on subject nodes. This is done by line 1-5 in Algorithm 1. In the example of Figure 2, *portray* is the predicate node and *woman* is the corresponding subject node which is the secondary unknown in the question Q.

Path-based Graph segmentation. Intersection questions and comparison questions share the commonality that they both have two parallel conditions or entities. Intersection questions require asking for an entity that satisfies multiple conditions (*"Are both Coldplay and Pierre Bouvier from the same country?"*), and comparison questions require comparing a property of two entities (*"Who is older, Annie Morton or Terry Richardson?"*). Thus, we propose a path-based method to capture these parallel conditions or entities by retrieving the longest identical paths in the AMR. Specifically, we take the linearized AMR graph as a sequence of symbols, made up of concepts and relations, and then view the longest identical sequences as identifying parallel conditions or entities (line 7 in Algorithm 1). We demonstrate this longest path retrieval on question Q2 in Figure 1. The following is the longest identical path; fields prefixed with a colon represent the relationship between concepts, *e.g.,* :arg and :name.

$$\text{start} \rightarrow :\text{arg} \rightarrow \text{magazine} \rightarrow :\text{name} \rightarrow < \text{Entity} >$$

Finally, we generate sub-graphs by disconnecting the path instances in turn, as shown in Figure 3.

3.4 Sub-question Generation

After graph segmentation, we apply an AMR-to-Text generator with the same architecture as the AMR parser to generate sub-questions from AMR sub-graphs [Bevilacqua *et al.*, 2021]. This model first transforms the graph into a sequence of symbols using a linearization technique, and then feeds it

Algorithm 1 : Question Decomposition Based on AMR

```
Input: Multi-hop Question Q
Output: Bridging/Intersection question: $q_1$, $q_2$;
Comparison question: $q_1$, $q_2$, $q_3$
 1: Linearized AMR Graph G :=AMR Parsing(Q)
 2: $Q_{type}$ := getReasoningType(G)
 3: if $Q_{type}$ is bridging then
 4:     secUnkNode := getSecondaryUnknows(Q, G)
 5:     $G_1$, $G_2$ := unkBasedSegmentation(G, secUnkNode)
 6: else
 7:     longestPath := getLongestIdenticalPath(G)
 8:     $G_1$, $G_2$ := pathBasedSegmentation(G, longestPath)
 9: end if
10: $q_1$ := AMR-to-Text Generation($G_1$)
11: $q_2$ := AMR-to-Text Generation($G_2$)
12: if $Q_{type}$ is bridging then
13:     $q_1^{Ans}$ := off-the-shelf QA($q_1$)
14:     $G_2^{'}$ := replace(G2, $q_1^{Ans}$, secUnkNode)
15:     $q_2$ := AMR-to-Text Generation($G_2^{'}$)
16: else if $Q_{type}$ is intesection then
17:     Continue
18: else
19:     op := findOperation(Q)
20:     $q_1^{Ans}$, $q_2^{Ans}$ := off-the-shelf QA($q_1$, $q_2$)
21:     $q_3$ := constructOpQuestion(op, $q_1^{Ans}$, $q_2^{Ans}$)
22: end if
23: if $Q_{type}$ is bridging or intersection then
24:     return $q_1$, $q_2$
25: else
26:     return $q_1$, $q_2$, $q_3$
27: end if
```

into the pretrained language model BART to obtain the subquestion. Similar to AMR parsing, the full graph-isomorphic linearization technique encodes the graph into a sequence of symbols without losing adjacency information. This helps make the generated text fluent. Especially, for comparison questions, we apply a single discrete operation [Min *et al.*, 2019] over the answers of sub-questions to generate the third sub-question (line 21 in Algorithm).

3.5 Single-hop Question Answering

To achieve interpretable reasoning, we first determine the order of sub-questions before answering them, because the answer to one sub-question may be part of another, *e.g.,* the answer to question Q1 *Shirley Temple* is equivalent to *woman* in AMR graph of Figure 2. Thus, we use different strategies to answer sub-questions for different reasoning types. For bridging questions, we first predict answers to sub-questions with secondary unknowns, then substitute the answer for the secondary unknowns in the primary sub-question, and finally take the answer of this primary sub-question as the final answer. For intersection questions, we do not need to identify the order of sub-questions because sub-questions are parallel, and the final answer is the intersection of answers of sub-questions. Similarly, the final answer of the comparison question is the result of applying some discrete operation over the answers of sub-questions [Min *et al.*, 2019].

Given ordered sub-questions, we train a single-hop QA on SQUAD [Rajpurkar *et al.*, 2016] and on the new single-hop QA dataset. The dataset consists of single-hop QA pairs constructed by [Pan *et al.*, 2020] on HotpotQA. Once trained, we use the resulting single-hop QA system as the off-the-shelf QA model and no longer fine-tune the model on multi-hop questions. In addition, we compute a forward pass on each sentence with sub-questions, if the sentence contains the answer of sub-questions, we take it as supporting evidence. In the end, sub-question-evidence pairs are used to support the interpretability of the single-hop QA model.

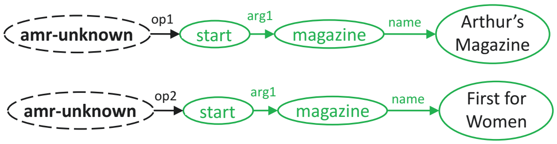

Figure 3: An example of path-based AMR graph segmentation by disconnecting instances of the longest identical path in turn.

4 Experiments

We empirically evaluate our QDAMR against state-of-the-art QD-based QA models on the HotpotQA [Yang *et al.*, 2018a].

4.1 Data and Setup

Dataset. We evaluate our model under the distractor setting of HotpotQA, which consists of questions and a collection of 10 paragraphs. Each question is originally annotated with either of two reasoning types: bridging and comparison. For the sake of fine-grained type annotations, all our methods further separate out a third type of question, intersection, from the bridging type, based on specific AMR concepts.

In addition, following [Pan *et al.*, 2020], we construct new single-hop QA pairs from HotpotQA using the following procedures: i) for bridging questions, we first find an entity *a* that links the gold-standard evidence, then use the pretrained transformer model T5 [Raffel *et al.*, 2019] to generate a question *Q* with *a* as the answer from the gold-standard evidence, and finally take *(Q, a)* as a new QA pair; ii) for intersection and comparison questions, we first find potential comparison entities e_1/e_2 from the gold-standard evidence, then generate a question Q_1/Q_2 that contains a specific entity e_1/e_2, and finally take *(Q_1, e_1)* and *(Q_2, e_2)* as a new QA pair.

Baselines. We compare our method with other QD-based QA on HotpotQA. For a fair comparison, we use the BERT-based QA from [Min *et al.*, 2019] as the off-the-shelf QA.

- **DecompRC** [Min *et al.*, 2019] trains a pointer model to identify split points in a question; these are subsequently used to compose sub-questions for each reasoning type.
- **OUNS** [Perez *et al.*, 2020] creates a noisy pseudo-decomposition for each multi-hop question, and then trains a decomposition model with unsupervised seq2seq learning to improve the pseudo-decomposition.
- **QDAMR**: QDAMR$^{\text{T5}}$ denotes the T5-based transformer model used for AMR parsing and AMR-to-Text generation; QDAMR$^{\text{SP}}$ denotes that we use Spring-based transformer model for AMR parsing and AMR-to-Text generation; QDAMR$^{\text{SP}}_{\text{RoBERTa}}$ denotes a finetuned RoBERTa-based QA model that is used as the off-the-shelf QA model.

Decomp Method	Annota$_{\mathrm{QDAMR}}$		Annota$_{\mathrm{DRC}}$	
	EM	F1	EM	F1
DecomRC	54.82	68.23	55.12	69.99
OUNS	58.74	73.40	58.74	73.40
QDAMR$^{\mathrm{T5}}$	62.24	77.19	60.07	75.34
QDAMR$^{\mathrm{SP}}$	64.57	78.44	63.13	77.61
QDAMR$^{\mathrm{SP}}_{\mathrm{RoBERTa}}$	**67.13**	**81.17**	**66.83**	**80.12**

Table 1: Results for QD-based multi-hop QA models on the dev set of HotpotQA. Annota$_{\mathrm{QDAMR}}$ denotes that the reasoning types of multi-hop questions are annotated by QDAMR. Annota$_{\mathrm{DRC}}$ denotes that the reasoning types of questions are annotated by the baseline DecompRC. RoBERTa denotes that we use a finetuned RoBERTa as the QA model. QDAMR achieves state-of-the-art results under both annotations.

Decomp Method	GPT2 ↓ NLL	%Well- ↑ formed	Edit ↓ Dist	Length ↓ Ratio
DecompRC	6.04	32.6	7.08	1.22
OUNS	5.56	60.9	5.96	1.08
QDAMR	**5.37**	**68.2**	**5.19**	**1.06**

Table 2: Quality Analysis of the sub-questions. GPT2 NLL reflects the fluency of the question.%Well-formed, Edit distance and Length Ratio reflect the wellformedness of the sub-question. ↓ means lower is better, and ↑ means higher is better.

Setup. Each of the tested methods separates the questions by type in a different manner. For both fairness and comprehensiveness in comparisons, we conduct two sets of experiments, each using either the DecompRC and QDAMR strategies with each tested system. Specifically, DecompRC uses the different question decomposition methods per reasoning type to answer the multi-hop question, and then uses a decomposition scorer to select the best answer as the final result. The reasoning type that yields the best result is returned as the reasoning type of the multi-hop question. While, our QDAMR first identifies the reasoning type of the question (see line 2 in Algorithm 1), and then uses the corresponding AMR-based decomposition method to answer the question.

Performance Metrics. Exact Match (EM) and partial match (F1) between the prediction and the ground truth answer are used as performance metrics [Yang *et al.*, 2018a].

AMR Parsing	AMR-to-Text Generation	EM	F1
CAIL	T5	58.89	74.34
T5	T5	62.24	77.19
Graphere	T5	63.36	77.83
Spring	Spring	**67.13**	**81.17**

Table 3: Analysis of the combination of AMR parsing and AMR-to-Text generation. The combination of Spring-based AMR parsing and Spring-based AMR-to-Text achieves the best EM/F1 score.

4.2 Main Results

Table 1 shows how the question decomposition method affects performance using the two question-type identification methods. Under the same setting, using a BERT-based QA model, our QDAMR models yield substantial improvements, where QDAMR$^{\mathrm{SP}}$ respectively improves EM/F1 by 9.75/10.21 and 5.83/5.04 in Annota$_{\mathrm{QDAMR}}$, and by 8.01/7.62 and 4.39/4.21 in Annota$_{\mathrm{DRC}}$, compared to DecompRC and OUNS. This demonstrates the effectiveness of our model's use of AMR to decompose multi-hop questions. Moreover, the performance under the Annota$_{\mathrm{QDAMR}}$ setting consistently outperforms that under the Annota$_{\mathrm{DRC}}$ setting, validating improved effectiveness identifying reasoning type over the DecomRC baseline. By fine-tuning RoBERTa on generated single-hop QA pairs and SQUAD, our model shows further improvements on EM/F1 with 2.56/2.73 in Annota$_{\mathrm{QDAMR}}$ and 3.7/2.59 in Annota$_{\mathrm{DRC}}$, compared to the QDAMR$^{\mathrm{SP}}$, and a substantial improvement over both the DecompRC and OUNS baselines. In the following ablation studies section, we analyse the quality of the sub-questions and sources of performance gain.

4.3 Quality of Sub-questions

We evaluate the quality of sub-questions from two perspectives: fluency and wellformedness. We measure fluency using the GPT-2 [Radford *et al.*, 2019] negative log-likelihood (NLL). For wellformedness, we use three metrics from OUNS: i) the proportion of well-formed sub-questions [Faruqui and Das, 2018]; ii) the token Levenstein (edit) distance between the multi-hop question and its generated sub-questions; and iii) the ratio between the length of the multi-hop question and the length of the sub-question.

As shown in Table 2, QDAMR outperforms DecompRC and OUNS on these four metrics; QDAMR achieves the best result on GPT-2 NLL score with 5.37 (lower is better). For the proportion of well-formed questions, QDAMR achieves the best score with 68.2, indicating that sub-questions generated by our model are more grammatical. The lowest Levenstein edit distance score indicates that the sentence structure of our generated sub-questions is more similar to that of the original multi-hop question.

4.4 Ablation Studies

Next, we verify the effectiveness of two modules, AMR paring and AMR-to-Text generation, in our proposed framework and analyse the performance of different decomposition methods across three reasoning types.

Effectiveness of AMR Parsing and AMR-to-Text Generation. Since we delegate the complexity of understanding multi-hop questions to the process of turning it into an AMR graph, and the generation of a well-formed sub-question to the process of AMR-based text synthesis, it is necessary to select a good combination of AMR parsing and AMR-to-Text generation. To this end, we use four AMR parsers (CAIL [Cai and Lam, 2020], Graphere [Hoang *et al.*, 2021], T5 and SPRING [Bevilacqua *et al.*, 2021]) to generate the AMR graph from a multi-hop question, and we generate sub-questions from AMR sub-graphs using two AMR-to-Text

Comparison Question: Who is older, Annie Morton or Terry Richardson? **Answer: Terry Richardson**			
Methods	DecompRC	OUNS	QDAMR
subQ1	Who is older?	Who is Annie Morton?	How old is Terry Richardson?
Ans1	(Annie Morton)	(American model)	(August 14, 1965)
subQ2	Who is Annie Morton or Terry Richardson?	When was Terry Richardson born?	What was Annie Moore's age?
Ans2	(Annie Morton)	(26 July 1999)	(October 8, 1970)
subQ3	Which is smaller (Ans1)(Ans2)?	–	Which is smaller (Ans1)(Ans2)?
Final Ans	Annie Morton	26 July 1999	**Terry Richardson**

Intersection Question: Are both Coldplay and Pierre Bouvier from the same country? **Answer: No**			
Methods	DecompRC	OUNS	QDAMR
subQ1	Are both coldplay?	Where are Coldplay and Coldplay from?	From what country is ColdPlay?
Ans1	(British rock band)	(British)	(British)
subQ2	Are pierre bouvier from the same country?	What country is Pierre Bouvier from?	Where is Pierre Bouvier from?
Ans2	(Canadian)	(Canadian)	(Canadian)
operation	$Intersection$(Ans1,Ans2)	–	$Intersection$(Ans1,Ans2)
Final Ans	**No**	British	**No**

Table 4: Examples of sub-questions generated by the evaluated decomposition methods. Our method is competitive in both the accuracy of the final answer and the quality of sub-questions.

Decomp Method	bridge		intersec		comparison	
	EM	F1	EM	F1	EM	F1
DecomRC	55.24	71.53	54.55	69.29	52.81	63.44
OUNS	66.41	80.84	66.93	81.07	65.62	79.43
QDAMR	**69.45**	**82.35**	**66.98**	**81.15**	**66.02**	**80.24**

Table 5: Performance analysis of decomposition across reasoning types. QDAMR-based QA model substantially outperforms the other QD-based QA on bridging questions, and may have a small performance advantage over OUNS for the other reasoning types.

generation methods (T5 and SPRING) because CAIL and Graphere do not support AMR-to-Text generation. In Table 3, we observe that sub-questions generated by the integration of the Spring-based AMR parsing and Spring-based AMR-to-Text generation subsystems achieve the best EM/F1 scores. This indicates the value of the Spring architecture in which full graph isomorphism allows for encoding and decoding without losing adjacency information, resulting in high fluency of generated sub-questions.

Decomposition Performance across Reasoning Types. We compare the performance of the three decomposition methods across reasoning types. As shown in Table 5, QDAMR outperforms other methods on all three tested reasoning types. We observe that the EM/F1 scores of our QDAMR on bridging questions (69.45/82.35) are improved by 3.04/1.51 and 14.21/10.82 compared with OUNS and DecompRC, respectively, indicating that unknowns-based decomposition can successfully identify the primary and secondary unknowns of the multi-hop question to generate well-formed sub-questions. For intersection and comparison questions, QDAMR still has a significant margin over DecompRC and slightly outperforms OUNS. This suggests that explicit path-based decomposition may hold advantages over the implicit pointer-based decomposition of DecompRC. The examples in Table 4 illustrate how decomposition in QDAMR works across reasoning types that require additional combination operations (intersection and comparison). We also observe that sub-questions generated by QDAMR are more consistently well-formed. In addition, our proposed pipeline-based modular method is a self-explanatory system, which directly incorporates interpretability into the structure of its QA pipeline; the following are the interpretable reasoning processes applied for each of the reasoning types, **B**ridging, **I**ntersection and **C**omparison:

B : $\mathrm{Q}{\rightarrow}\mathrm{SubQ1}{\rightarrow}\mathrm{Ans1}{\rightarrow}\mathrm{SubQ2}{\rightarrow}\mathrm{Ans}$

I : $\mathrm{Q}{\rightarrow}(\mathrm{SubQ1}, \mathrm{SubQ2}){\rightarrow}\mathrm{intersec}(\mathrm{Ans1}, \mathrm{Ans2}){\rightarrow}\mathrm{Ans}$

C : $\mathrm{Q}{\rightarrow}(\mathrm{SubQ1}, \mathrm{SubQ2}){\rightarrow}(\mathrm{Ans1}, \mathrm{Ans2}){\rightarrow}\mathrm{SubQ3}{\rightarrow}\mathrm{Ans}.$

5 Conclusion and Future Work

We have demonstrated use of question decomposition based on the AMR semantic representation for multi-hop QA, using an intrinsically interpretable framework to incorporate interpretability directly into the system structure. The complex task of multi-hop question interpretation is delegated to AMR parsers. These parsers produce AMR Graphs to which two segmentation methods are applied, *i.e.,* unknown-based and path-based graph segmentation, to achieve question decomposition. To generate a well-formed sub-question, we perform both AMR parsing and AMR-to-Text generation with the same architecture, which uses a fully graph-isomorphic linearization technique to complete the transformation from graph to a sequence of symbols without losing adjacency information. This provides a good foundation for generating well-formed sub-questions using the AMR-to-Text generation model. Experimental results demonstrate that our QDAMR system outperforms baseline question decomposition methods, both in performance of multi-hop QA and in the quality of generated sub-questions. Since our proposed graph segmentation methods are based on predicate-argument relations and parallel conditions/entities respectively, they could in principle be generalized to an unknown number of hops by identifying multiple predicate nodes or capturing multiple parallel conditions/entities. While, as noted, an aim of the QDAMR design is to provide inherent interpretability, the effectiveness with which its outputs serve as explanations for human users remains to be evaluated in future work.

Acknowledgments

This work was supported by a grant from the New Zealand Tertiary Education Commission and by the Strong AI Lab at the University of Auckland. The first author is supported by a PhD scholarship from the China Scholarship Council.

References

[Bevilacqua *et al.*, 2021] Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021.

[Cai and Lam, 2020] Deng Cai and Wai Lam. Amr parsing via graph-sequence iterative inference. *arXiv preprint arXiv:2004.05572*, 2020.

[Chen *et al.*, 2020] Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Wang. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. *arXiv preprint arXiv:2004.07347*, 2020.

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

[Fang *et al.*, 2019] Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. Hierarchical graph network for multi-hop question answering. *arXiv preprint arXiv:1911.03631*, 2019.

[Faruqui and Das, 2018] Manaal Faruqui and Dipanjan Das. Identifying well-formed natural language questions. *arXiv preprint arXiv:1808.09419*, 2018.

[Hoang *et al.*, 2021] Thanh Lam Hoang, Gabriele Picco, Yufang Hou, Young-Suk Lee, Lam Nguyen, Dzung Phan, Vanessa López, and Ramon Fernandez Astudillo. Ensembling graph predictions for amr parsing. *Advances in Neural Information Processing Systems*, 34, 2021.

[Kapanipathi *et al.*, 2021] Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramón Fernandez Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue-Nkoutche, et al. Leveraging abstract meaning representation for knowledge base question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3884–3894, 2021.

[Lewis *et al.*, 2019] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[Liao *et al.*, 2018] Kexin Liao, Logan Lebanoff, and Fei Liu. Abstract meaning representation for multi-document summarization. *arXiv preprint arXiv:1806.05655*, 2018.

[Min *et al.*, 2019] Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. Multi-hop reading comprehension through question decomposition and rescoring. *arXiv preprint arXiv:1906.02916*, 2019.

[Mitra and Baral, 2016] Arindam Mitra and Chitta Baral. Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2779–2785, 2016.

[Pan *et al.*, 2020] Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. Unsupervised multi-hop question answering by question generation. *arXiv preprint arXiv:2010.12623*, 2020.

[Perez *et al.*, 2020] Ethan Perez, Patrick S. H. Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. Unsupervised question decomposition for question answering. In *EMNLP*, pages 8864–8880, 2020.

[Qiu *et al.*, 2019] Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6140–6150, 2019.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[Raffel *et al.*, 2019] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

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

[Rakshit and Flanigan, 2021] Geetanjali Rakshit and Jeffrey Flanigan. Asq: Automatically generating question-answer pairs using amrs. *arXiv preprint arXiv:2105.10023*, 2021.

[Ribeiro *et al.*, 2020] Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. Investigating pretrained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426*, 2020.

[Ribeiro *et al.*, 2021] Leonardo FR Ribeiro, Yue Zhang, and Iryna Gurevych. Structural adapters in pretrained language models for amr-to-text generation. *arXiv preprint arXiv:2103.09120*, 2021.

[Tu *et al.*, 2020] Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 9073–9080, 2020.

[Xu *et al.*, 2021] Weiwen Xu, Huihui Zhang, Deng Cai, and Wai Lam. Dynamic semantic graph construction and reasoning for explainable multi-hop science question answering. *arXiv preprint arXiv:2105.11776*, 2021.

[Yang *et al.*, 2018a] 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 *ACL*, pages 2369–2380, 2018.

[Yang *et al.*, 2018b] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, pages 2369–2380, 2018.

[Zhu *et al.*, 2021] Yunchang Zhu, Liang Pang, Yanyan Lan, Huawei Shen, and Xueqi Cheng. Adaptive information seeking for open-domain question answering. *arXiv preprint arXiv:2109.06747*, 2021.