Asynchronous Multi-grained Graph Network For Interpretable Multi-hop Reading Comprehension

Ronghan Li, Lifang Wang*, Shengli Wang and Zejun Jiang
School of Computer Science and Engineering, Northwestern Polytechnical University, Xi’an, China
{lrh000, wsljsj}@mail.nwpu.edu.cn, {wanglf,claud}@nwpu.edu.cn

Abstract

Multi-hop machine reading comprehension (MRC) task aims to enable models to answer the compound question according to the bridging information. Existing methods that use graph neural networks to represent multiple granularities such as entities and sentences in documents update all nodes synchronously, ignoring the fact that multi-hop reasoning has a certain logical order across granular levels. In this paper, we introduce an Asynchronous Multi-grained Graph Network (AMGN) for multi-hop MRC. First, we construct a multi-grained graph containing entity and sentence nodes. Particularly, we use independent parameters to represent relationship groups defined according to the level of granularity. Second, an asynchronous update mechanism based on multi-grained relationships is proposed to mimic human multi-hop reading logic. Besides, we present a question reformulation mechanism to update the latent representation of the compound question with updated graph nodes. We evaluate the proposed model on the HotpotQA dataset and achieve top competitive performance in distractor setting compared with other published models. Further analysis shows that the asynchronous update mechanism can effectively form interpretable reasoning chains at different granularity levels.

1 Introduction

Compared with single-hop Machine Reading Comprehension (MRC), where the question can be answered by simply matching a span [Rajpurkar et al., 2016], multi-hop MRC requires models to answer compound questions based on bridging information. Recent datasets such as HotpotQA [Yang et al., 2018] and QAngaroo [Welbl et al., 2018] have been proposed for studying multi-hop MRC over multiple pieces of evidence. Particularly, HotpotQA also requires models to predict supporting sentences for explainable question answering (QA).

Existing work has proved that Graph Neural Networks (GNN) is useful for multi-hop reasoning because of their natural relationship representation ability and inductive bias [Song et al., 2018; Cao et al., 2019; Ding et al., 2019]. Fine-grained nodes in the entity graph [Qiu et al., 2019] or heterogeneous graph [Tu et al., 2019] pass messages with each other based on defined relationships. Fang et al. [2020] combine the question and different levels of granularity representations (entity, sentence, paragraph) in the document to construct a hierarchical graph for multi-hop reasoning. However, there are still several limitations of the current GNN-based approaches. First, the above methods perform message passing synchronously at each step of the graph update, ignoring the fact that different-level relationships have different priorities and the reasoning needs to follow an ordered logic. For example, as shown in Figure 1, the same entity-level mentions “Robert W. McElroy” and “McElroy” can locate the third sentence in Paragraph 1 as bridge information, which is further used to infer that “America” is the name of the magazine instead of the American magazine “Texas Monthly”. This is sentence-level descriptive information and the model is easily misled by the distractor if the reasoning is not effectively performed in a certain fine-grained logical order, which is ignored by previous work.

Second, existing GNN-based methods either only use the updated entity nodes to reformulate the question embeddings or take the question as a node to update synchronously, which is insufficient since the supporting facts may not be evident in the question. Additionally, the guiding effect of a compound question on the reasoning chain should be reflected in the granularity and sequentiality of supporting fact prediction.

To this end, we propose an Asynchronous Multi-grained Graph Network (AMGN) for multi-hop MRC, which asynchronously updates multi-grained nodes based on different levels of relationships to mimic the logical order of multi-hop reasoning. Specifically, we first use a large-scale pre-trained language model such as Roberta [Liu et al., 2019] to encode the context and the question and then construct a graph with entity and sentence nodes. Particularly, we define relationship groups according to different granularity levels, and each relationship group is represented using independent parameters. We propose an algorithm for asynchronous message propagation according to the relationship levels (e.g., entity-entity

*Corresponding author
Robert Walter McElroy (born February 5, 1954) is a Roman Catholic. McElroy was educated by the Jesuits and writes for their official publication in the United States, “America.”

America is a national weekly magazine published by the Jesuits of the United States and headquartered in midtown Manhattan.

Texas Monthly is a monthly American magazine headquartered in Downtown Austin, Texas.

Question: Where is the magazine headquartered that Robert W. McElroy writes for?
Answer: Manhattan

Figure 1: An illustration of multi-hop MRC with an example from the HotpotQA dataset. The initial entity is marked as red. Bridging information (supporting facts) are marked as blue. The final answer is marked in green. The solid lines denote the correct reasoning chain, and its color represents different granularity levels. Grey dotted lines indicate situations that may be misleading.

$ightarrow$ entity-sentence → sentence-sentence) to update the graph to mimic human multi-hop reading logic. Besides, for the second challenge, a RNN based reformulation mechanism is introduced to iteratively update the latent question representation with sentence nodes. These sentence nodes are directly used for supporting fact prediction. In this way, we have performed asynchronous updates between the question, entities, and sentences to represent the sequential process of multi-hop reasoning.

We evaluate the proposed model on the HotpotQA dataset and achieve top competitive performance in distractor setting compared to other published models. Extensive experiments show that the asynchronous update mechanism can effectively form interpretable reasoning chains at different granularity levels.

## 2 Related Work

Multi-hop MRC task aims to enable models to answer the compound question according to the bridge information scattered in multiple documents. Several multi-hop MRC datasets such as HotpotQA [Yang et al., 2018], QAncargo [Welbl et al., 2018], and MultiRC [Khashabi et al., 2018] are recently released. In this work, we focus on extractive MRC and choose HotpotQA for experimental analysis. Existing work for multi-hop MRC can be mainly divided into two categories: recurrent reasoning based on memory retrieval and multi-hop reasoning based on graph neural networks. The first group focuses on decomposing the question [Min et al., 2019] and updating the latent representations with the interaction of the question and the context in a recurrent network [Das et al., 2019; Jiang et al., 2019; Feldman and El-Yaniv, 2019]. GOLDEN Retriever [Qi et al., 2019] generates intermediate natural language search queries given the question and available context and leverages off-the-shelf information retrieval systems to query for missing entities, which is further enhanced by [Qi et al., 2020] using iterative reranking. Nie et al. [2019] propose a pipeline system specializing in hierarchical semantic retrieval at both paragraph and sentence levels. Jiang et al. [2019] leverage a dynamical RNN to construct a self-assembling neural modular network. Asai et al. [2020] construct an offline Wikipedia graph with hyperlinks and form the reasoning chain by using a beam search over the graph, where a RNN module is responsible for multi-hop retrieval reasoning. More recently, work by [Yadav et al., 2020; Perez et al., 2020] manages to decompose compound questions in an unsupervised fashion to retrieve useful evidences effectively.

The second group manages to build a document graph for multi-hop MRC and reasoning over the constructed graph using graph neural networks. Considerable studies focus on a single level of granularity representation such as CorefGRN [Dhingra et al., 2018], Entity-GCN [Cao et al., 2019], DFGN [Qiu et al., 2019] and CogQA [Ding et al., 2019]. Tu et al. [2019] construct a heterogeneous graph to enrich the interaction among document nodes, entity nodes, and candidate nodes. SAE [Tu et al., 2020] builds a GNN model over sentence-level embeddings to explicitly facilitate multi-hop reasoning over all sentences from the predicted gold documents. C2F Reader [Shao et al., 2020] enhances DFGN and argues that graph-attention can be considered as a special case of self-attention, which may not be necessary for multi-hop MRC. HGN [Fang et al., 2020] combines the question and different levels of granularity representations (entity, sentence, paragraph) in the document to construct a hierarchical graph for multi-hop reasoning. DDR [Zhang et al., 2020] employs an entity-linked document graph for multi-document interaction and iteratively retrieves, reranks, and filters documents. Different from the above methods, our work focuses on the asynchronous message propagation and latent question update.

## 3 Methodology

### 3.1 Overview

An overview of our model is shown in Figure 2. The model consists of four main components: a paragraph selector for reducing search space, an encoder for encoding the selected context and the given question, a reasoning module for multi-grained graph construction and multi-step asynchronous node update, and a multi-task prediction module. Our major contribution is the reasoning module AMGN, which constructs a graph containing multi-grained relationship groups and leverages an asynchronous update mechanism to explicitly consider the logic order at different granularity levels.

**Paragraph selector.** Similar to DFGN [Qiu et al., 2019], we first pre-train a BERT-based network to retrieve paragraphs relevant to the question. Instead of using a threshold to limit search space in DFGN, we take top-$K$ paragraphs $\{P_1, P_2, ..., P_K\}$ and concatenate them as context $C$ based on the output score. Specifically, we independently encode each candidate paragraph along with the given question $Q$. The output [CLS] token representation is input to a binary classifier to predict whether the input paragraph contains the ground-truth supporting facts or not.
Context and question encoder. We concatenate the question $Q$ with the context $C$ and feed them to a pre-trained BERT model to obtain representations $Q = \{q_1, q_2, \ldots, q_m\} \in \mathbb{R}^{m \times d_1}$ and $C = \{c_1, c_2, \ldots, c_n\} \in \mathbb{R}^{n \times d_1}$, where $d_1$ is the size of hidden states. Following the design of [Qiu et al., 2019], we further feed the representations into a bi-attention [Seo et al., 2017] layer to reduce the dimension of the question and the context, obtaining $\hat{Q} \in \mathbb{R}^{m \times d_2}$ and $\hat{C} \in \mathbb{R}^{n \times d_2}$, where $d_2 < d_1$.

Multi-step asynchronous reasoning module. First we construct a multi-grained graph $\hat{G} = (\hat{V}, \hat{R})$ where the node set $\hat{V} = E \cup S$. Entity node representations $E \in \mathbb{R}^{n_e \times d_2}$ and sentence node representations $S \in \mathbb{R}^{n_s \times d_2}$ are initialized with $\hat{C}$ and connected by semantic relationships $R$ (see Section 3.2). $n_e$ and $n_s$ denote the number of the entity and sentence nodes, respectively. Then we perform multi-step asynchronous reasoning on the graph based on AMGN, which can be formally described as:

$$G^T, q^T = \text{AMGN}(\hat{G}^0, q^0)$$

where $T$ is the number of iterations and $q^0 = \text{MaxPooling}(\hat{Q})$. The details of AMGN will be explained in Section 3.3.

Multi-task prediction. For supporting facts prediction, the final sentence node representations $S^T$ after asynchronous propagation are exploited directly:

$$O_{sent} = \text{FFN}(S^T)$$

where $O_{sent} \in \mathbb{R}^{n_s}$. Following DFGN, we map the logits of the supporting facts $O_{sent} \in \mathbb{R}^{n_s}$ to $\hat{C} \in \mathbb{R}^n$. Then $\hat{C}$ is concatenated with $\hat{C}$ and fed into a cascade BiLSTM for answer and type prediction. We jointly train the model in a multi-task way. Please refer to [Qiu et al., 2019] for more details due to space limitation.

3.2 Multi-grained Graph Construction

Given the related paragraphs retrieved by the selector, we focus on entity-level and sentence-level information used to extract a span as the answer. Note that paragraph retrieval also needs multi-hop reasoning in open-domain QA, which beyond the scope of this work. We leave it as future work and provide a retrieval quality analysis in the appendix.

Node representation. We use token representations of the corresponding text spans in $C$ to initialize entity and sentence nodes. Specifically, these representations are fed into a self-attention pooling layer to calculate the node representations. Given the start and end positions $< p_{\text{start}}(v_i), p_{\text{end}}(v_i) >$ in the text span corresponding to the $i$-th node $v_i$, the initial node representation $v_i^0$ is calculated as:

$$v_i^0 = \sum_{k=p_{\text{start}}(v_i)}^{p_{\text{end}}(v_i)} \alpha_k \cdot \hat{c}_k$$

where $v_i^0 \in \mathbb{R}^{d_2}$ and $w \in \mathbb{R}^{d_2}$. In practice, we find using the self-attention pooling achieves better performance. We also experiment with other common pooling methods. Please refer to the supplementary materials for more details.

Group-based relationship definition. We define the following six types of edges in the graph to describe the semantic relationship between entities and sentences: (i) edges between entities that appear in the same sentence; (ii) edges between entities that have the same mentions text in $C$; (iii) edges between entity nodes and the sentence nodes to which they belong; (iv) edges between entity nodes and sentence nodes containing the same mentions; (v) edges between adjacent sentence nodes in the same paragraph; (vi) edges between sentence nodes that contain mentions of the same en-
entity. Obviously, the graph depicts the three levels of relationships of entity-entity $R_{ee}$, entity-sentence $R_{es}$, and sentence-sentence $R_{ss}$. Different from previous studies, we represent each level of relationships using independent parameters (see Eq. 7). We believe that this has the following advantages: 1) Representing relationship groups based on the granularity level is more scalable for new structured knowledge; 2) It is more interpretable to construct the reasoning chain based on different granularity levels.

3.3 Multi-step Asynchronous Graph Reasoning

AMGN is designed to perform multi-step asynchronous reasoning in a certain logical order and is a core module of our model. At each iteration $t \in \{1, ..., T\}$, after further interacting with the question, the nodes pass messages to each other asynchronously according to the relationship of different granularities. Then, the question will also be updated through sentence nodes in order to find the next-hop clue.

**Question-aware graph.** The multi-grained graph needs to highlight the entities and sentences relevant to the question. To this end, we adopt element-wise gate to perform filtering on entity and sentence nodes. Specifically:

$$g_i^t = \text{ELU}(W^r [v_i^t; q_i^{t-1}])$$  \hspace{1cm} (5)

$$\hat{v}_i^{t-1} = g_i^t \odot v_i^{t-1}$$  \hspace{1cm} (6)

where $W^r \in \mathbb{R}^{d_2 \times 2d_2}$, $\odot$ means the element-wise multiplication, and $[;]$ denotes the concatenation operation. The question-aware nodes $\hat{v}_i^{t-1}$ will be used to pass message in the next layer.

**Asynchronous information propagation.** Intuitively, the reasoning chain is formed by the semantic relationship between different levels of granularity representations in a certain logical order. In this paper, we use Graph Attention Network (GAT) to update nodes asynchronously. Formally:

$$u_i^t = \text{ReLU}(\sum_{r \in \mathcal{R}_{ee}} \sum_{j \in \mathcal{N}_i^r} \alpha_{i,j} W^r_{ee} \hat{v}_j^{t-1})$$  \hspace{1cm} (7)

$$\alpha_{i,j} = \frac{\exp(f(W^r_{ee} [\hat{v}_i^{t-1}; \hat{v}_j^{t-1}])))}{\sum_{r \in \mathcal{R}_{ee}} \sum_{k \in \mathcal{N}_j^r} \exp(f(W^r_{ee} [\hat{v}_i^{t-1}; \hat{v}_k^{t-1}])))$$  \hspace{1cm} (8)

where $\mathcal{N}_i^r$ represents the neighbors of the $i$-th node through the relation $r$, $W^r_{ee} \in \mathbb{R}^{2d_2}$ is a parameter representing the entity-entity level relationship group, $W^r_{ee} \in \mathbb{R}^{2d_2}$ is the weight vector corresponding to the relation $r$ between $i$-th and $j$-th nodes, and $f$ denotes the LeakyRelu activation function. In this way, we calculate the weight differentiating each relationship while conduct information propagation at the relationship group level. The $u_i^t$ is then added to $v_i^t$ to update the nodes at the current relationship level:

$$m_i^{t,ee} = \begin{cases} v_i^{t-1} + u_i^t, & i \in \{x | \exists r \in \mathcal{R}_{ee}, \mathcal{N}_x^r \neq \emptyset\} \\ v_i^{t-1}, & \text{otherwise} \end{cases}$$  \hspace{1cm} (9)

We denote the above reasoning process (Eq. 7-9) as a single function:

$$\tilde{M}^{t,ee} = \text{Update}^{ee}(V^{t-1})$$  \hspace{1cm} (10)

For $R_{es}$ and $R_{ss}$, we perform the update asynchronously in the same way:

$$\tilde{M}^{t,es} = \text{Update}^{es}(\tilde{M}^{t,ee})$$  \hspace{1cm} (11)

$$V^t = \text{Update}^{ss}(\tilde{M}^{t,es})$$  \hspace{1cm} (12)

In this way, the module mimics the heuristic of human multi-hop reasoning: finding the entity related to the question and other relevant entities through $R_{ee} \rightarrow$ locating the sentence nodes to which these entities belong through $R_{es} \rightarrow$ comparing the descriptive information in these sentences through $R_{ss}$ to determine the supporting facts. In practical, asynchronous reasoning is essentially regarded as a self-attention calculation across all nodes using different relation group masks, which improves computational efficiency.

**Question reformulation.** Questions requiring multi-hop reasoning are usually compound, that is, bridge information can answer part of the question. Hence, inspired by [Das et al., 2019], we propose a RNN-based question reformulation mechanism, where the latent representation $q_i^{t-1}$ is reformulated with updated sentence nodes. The reformulation mechanism is implemented as follow:

$$q_i^t = \text{GRU}(q_i^{t-1}, s_i)$$  \hspace{1cm} (13)

$$s_i^t = \sum_{i=1}^{n_s} \beta_i \cdot s_i^t$$  \hspace{1cm} (14)

$$\beta_i = \frac{\exp(q_i^{t-1} \cdot s_i^t)}{\sum_{j=1}^{n_s} \exp(q_i^{t-1} \cdot s_j^t)}$$  \hspace{1cm} (15)

The $q_i^t$ and $V^t$ serve as the input of the next-hop reasoning module. We perform $T$ iterations of the reasoning step (Eq. 1) and the output of the last layer $V^T$ is obtained for answer and support sentences prediction.

4 Experiments and Analysis

4.1 Experimental Setup

**Dataset.** We evaluate our method on HotpotQA [Yang et al., 2018], which is a prevalent benchmark for multi-hop MRC. Specifically, HotpotQA has two settings Distractor and Fullwiki. For each question, the Distractor setting contains two gold paragraphs with ground-truth answers and supporting facts and eight negative paragraphs as distractors, while the Fullwiki setting requires to retrieve Wikipedia to obtain relevant paragraphs. We focus on the Distractor setting since retrieval quality is not the focus of this paper. In addition to evaluating answers and supporting facts separately, a joint EM and F1 score are used to measure the final performance, which encourages the model to take both tasks into consideration.

**Implementation details.** We implement our experiments based on Huggingface\(^1\) and the official open-source implementation of DFGN\(^2\). We use Roberta-large for paragraph selection and set $K = 3$. For context encoding, we use another

\(^1\)https://github.com/huggingface/transformers
\(^2\)https://github.com/woshiyya/DFGN-pytorch
Table 1: F1 performance comparison on the HotpotQA in the Distractor setting. (1) denotes unpublished work. (2) is our re-implementation using the Roberta-large retriever and encoder in the fine-tuning setting.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Ans</th>
<th>Sup</th>
<th>Joint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dev</td>
<td>DFGN</td>
<td>80.03</td>
<td>87.67</td>
<td>73.18</td>
</tr>
<tr>
<td></td>
<td>SAE-large [Tu et al., 2020]</td>
<td>80.75</td>
<td>87.38</td>
<td>72.75</td>
</tr>
<tr>
<td></td>
<td>C2F Reader [Shao et al., 2020]</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>HGN-large [Fang et al., 2020]</td>
<td>82.22</td>
<td>88.58</td>
<td>74.37</td>
</tr>
<tr>
<td></td>
<td>AMGN (ours)</td>
<td>83.11</td>
<td>88.69</td>
<td>74.76</td>
</tr>
<tr>
<td></td>
<td>AMGN+ (ours)</td>
<td><strong>83.46</strong></td>
<td><strong>89.13</strong></td>
<td><strong>75.48</strong></td>
</tr>
<tr>
<td>Test</td>
<td>Baseline [Yang et al., 2018]</td>
<td>59.02</td>
<td>64.49</td>
<td>40.16</td>
</tr>
<tr>
<td></td>
<td>QFE [Nishida et al., 2019]</td>
<td>68.06</td>
<td>84.49</td>
<td>59.61</td>
</tr>
<tr>
<td></td>
<td>DecomprC [Min et al., 2019]</td>
<td>69.63</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DFGN [Qiu et al., 2019]</td>
<td>69.69</td>
<td>81.62</td>
<td>59.82</td>
</tr>
<tr>
<td></td>
<td>TAP 2 [Glass et al., 2020]</td>
<td>78.59</td>
<td>85.57</td>
<td>69.12</td>
</tr>
<tr>
<td></td>
<td>SAE-large [Tu et al., 2020]</td>
<td>79.62</td>
<td>86.86</td>
<td>71.45</td>
</tr>
<tr>
<td></td>
<td>C2F Reader [Shao et al., 2020]</td>
<td>81.24</td>
<td>87.63</td>
<td>72.73</td>
</tr>
<tr>
<td></td>
<td>Longformer [Beltagy et al., 2020]</td>
<td>81.25</td>
<td>88.34</td>
<td>73.16</td>
</tr>
<tr>
<td></td>
<td>ETC-large [Zaheer et al., 2020]</td>
<td>81.18</td>
<td><strong>89.09</strong></td>
<td>73.62</td>
</tr>
<tr>
<td></td>
<td>HGN-large [Fang et al., 2020]</td>
<td>82.19</td>
<td>88.47</td>
<td>74.21</td>
</tr>
<tr>
<td></td>
<td>SpiderNet-large†</td>
<td>83.02</td>
<td>88.85</td>
<td>74.88</td>
</tr>
<tr>
<td></td>
<td>AMGN (ours)</td>
<td>82.79</td>
<td>88.12</td>
<td>74.20</td>
</tr>
<tr>
<td></td>
<td>AMGN+ (ours)</td>
<td><strong>83.37</strong></td>
<td><strong>88.83</strong></td>
<td><strong>75.24</strong></td>
</tr>
</tbody>
</table>

Table 2 summarizes the contributions of components of our model. We notice that removing the SS subgraph and removing the EE subgraph have a significant impact on supporting fact prediction and answer prediction, respectively. One possible reason is that most of the answers in HotpotQA are based on entities. Without the $R_{ee}$, the joint F1 drops by 1.31 points, showing that the cross-granular relationship contributes to the model performance. Leveraging a multi-grained graph improves the joint F1 score over the vanilla RoBERTa by 5.36%. Either using global parameters or without parameters results in performance degradation, showing the effectiveness of group-based relation definition. We further provide an analysis of group-based representation in the supplementary materials. Besides, our question reformulation mechanism also provides around 2% performance improvement and it mainly affects supporting factual predictions, which is reasonable because we use sentence representations to update the question.

4.3 Ablation Study

Table 2 presents a summary of the contributions of components of our model. We notice that removing the SS subgraph and removing the EE subgraph have a significant impact on supporting fact prediction and answer prediction, respectively. One possible reason is that most of the answers in HotpotQA are based on entities. Without the $R_{ee}$, the joint F1 drops by 1.31 points, showing that the cross-granular relationship contributes to the model performance. Leveraging a multi-grained graph improves the joint F1 score over the vanilla RoBERTa by 5.36%. Either using global parameters or without parameters results in performance degradation, showing the effectiveness of group-based relation definition. We further provide an analysis of group-based representation in the supplementary materials. Besides, our question reformulation mechanism also provides around 2% performance improvement and it mainly affects supporting factual predictions, which is reasonable because we use sentence representations to update the question.

4.4 Effectiveness of Asynchronous Update

To analyze the effectiveness of the asynchronous update, we experiment with different update orders. Table 3 indicates that the best results are obtained by updating the nodes with different granularities in the graph through the logical sequence closest to the human heuristic (finding related entities, reading the sentences where the entities are, judging which sentences are supporting facts, and obtaining the bridge information or final answer). We also find that not all asynchronous update variants perform better than the synchronous update (e.g., Simultaneous Update vs. $R_{es} \rightarrow R_{ec} \rightarrow R_{es}$). It is worth mentioning that the synchronous update mainly causes a decline in support F1 (88.69% to 87.51%). We argue that preferentially updating some nodes will progressively affect the representation of subsequent nodes, causing different effects on the final answer prediction and the evidence prediction.
In this paper, we present AMGN for multi-hop reading comprehension. We introduce an asynchronous update mechanism of the multi-grained graph to mimic the logic of human multi-hop reading. Additionally, a RNN-based question reformulation method is proposed to update the hidden representation of compound questions. Experimental results and analysis show that asynchronously updating the graph in accordance with the sequence of human logic can boost the performance of the model by a significant margin under the premise of the reasonable use of node representations. As for future work, both paragraph-level and document-level representations can be considered as more complex composition elements, and the calculation order of logical relationships also needs to be designed adaptively. Besides, we would like to evaluate our model on other multi-hop MRC datasets.

Table 4: Performance comparison for different reasoning types.

<table>
<thead>
<tr>
<th>Model</th>
<th>Ans F1</th>
<th>Sup F1</th>
<th>Joint F1</th>
<th>Ans F1</th>
<th>Sup F1</th>
<th>Joint F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>HGN</td>
<td>81.90</td>
<td>87.60</td>
<td>73.31</td>
<td>83.49</td>
<td>92.49</td>
<td>78.58</td>
</tr>
<tr>
<td>AMGN</td>
<td>83.10</td>
<td>87.91</td>
<td>73.85</td>
<td>83.24</td>
<td>91.83</td>
<td>78.37</td>
</tr>
<tr>
<td>AMGN+</td>
<td><strong>83.42</strong></td>
<td><strong>88.27</strong></td>
<td><strong>74.66</strong></td>
<td><strong>83.82</strong></td>
<td><strong>92.17</strong></td>
<td><strong>78.94</strong></td>
</tr>
</tbody>
</table>

4.6 Case Study

We further provide a case study to illustrate the specific reasoning chain in AMGN. We visualize the row-wise correlation \( \alpha_{i,j} \) in Eq. 8 between part of entity and sentence nodes after Update \( \text{es}_i \), Update \( \text{ex}_i \), and Update \( \text{es}_i \). The results are shown in Figure 3, where the same y-axis labels of the 2-hop EE and ES heatmaps as 1-hop are omitted. We can observe that the cross-granularity attention weights explicitly emphasize the formation process of the reasoning chain in the asynchronous update phase. For example, in the 1-hop reasoning stage, more attention focuses on between “Big Stone Gap” in sent_1 and “Adriana Trigiani” in sent_1, “Adriana Trigiani” in sent_1 and “Adriana Trigiani” in sent_1, and “Adriana Trigiani” in sent_2 and sent_1, and sent_1 and sent_2. Note that the confusing clues such as “Virginia” in sent_2 are also considered in multi-hop reasoning, which shows that AMGN compares the sentence information and makes a correct prediction.

5 Conclusion

In this paper, we present AMGN for multi-hop reading comprehension. We introduce an asynchronous update mechanism of the multi-grained graph to mimic the logic of human multi-hop reading. Additionally, a RNN-based question reformulation method is proposed to update the hidden representation of compound questions. Experimental results and analysis show that asynchronously updating the graph in accordance with the sequence of human logic can boost the performance of the model by a significant margin under the premise of the reasonable use of node representations. As for future work, both paragraph-level and document-level representations can be considered as more complex composition elements, and the calculation order of logical relationships also needs to be designed adaptively. Besides, we would like to evaluate our model on other multi-hop MRC datasets.

References

[Asai et al., 2020] Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. Learning to re-


