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

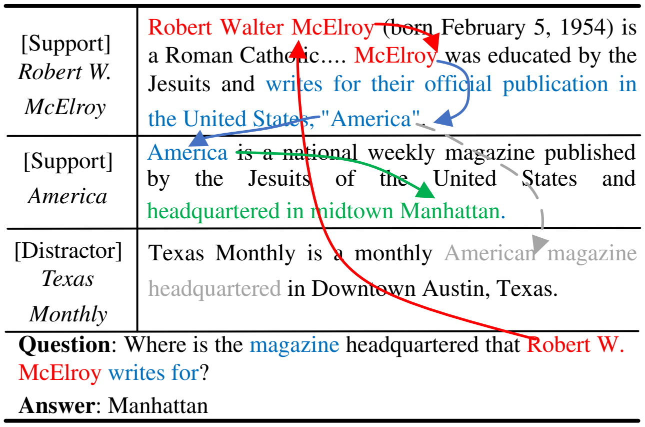
Compared with single-hop Machine Reading Comprehension (MRC), where the question can be answered by simply matching a span [Rajpurkar *et al.*, 2016], multi-hop MRC requires models to answer compound questions based on bridging information. Recent datasets such as HotpotQA [Yang *et al.*, 2018] and QAngaroo [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



[Support] *Robert W. McElroy*	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".
[Support] *America*	America is a national weekly magazine published by the Jesuits of the United States and headquartered in midtown Manhattan.
[Distractor] *Texas Monthly*	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 line denotes the correct reasoning chain, and its color represents different granularity levels. Grey dotted lines indicate situations that may be misleading.

→ 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], QAngaroo [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-step 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 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 Coref-GRN [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.

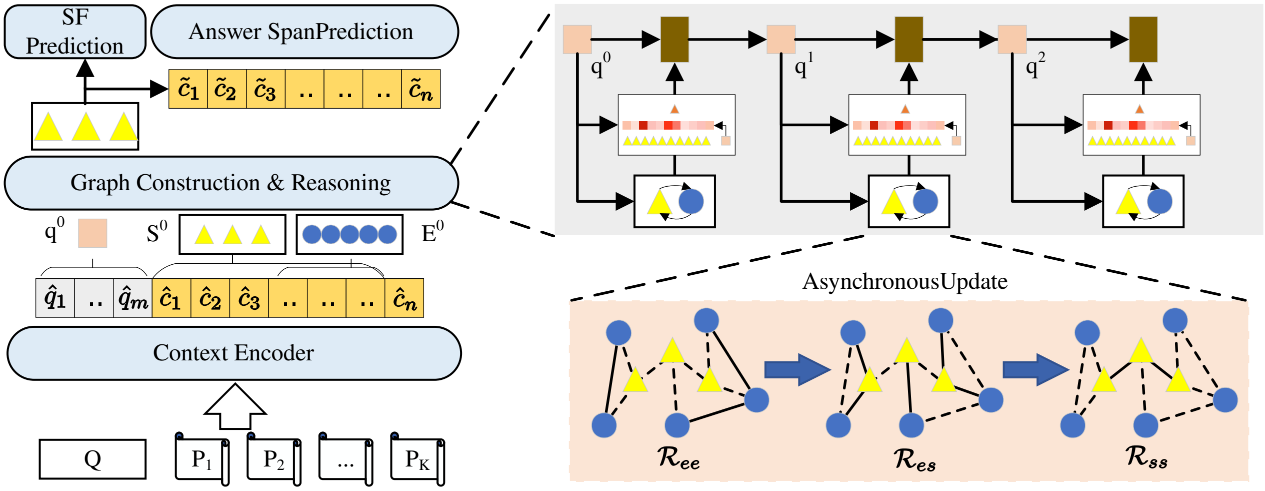

SF Prediction
Answer SpanPrediction
$\tilde{c}_1$ $\tilde{c}_2$ $\tilde{c}_3$ .. .. .. $\tilde{c}_n$
Graph Construction & Reasoning
q^0
S^0
E^0
$\hat{q}_1$ .. $\hat{q}_m$ $\hat{c}_1$ $\hat{c}_2$ $\hat{c}_3$ .. .. .. $\hat{c}_n$
Context Encoder
Q
P_1
P_2
...
P_K
q^0
q^1
q^2
AsynchronousUpdate
$\mathcal{R}_{ee}$
$\mathcal{R}_{es}$
$\mathcal{R}_{ss}$

Figure 2: The architecture of the proposed AMGN. In asynchronous update, the solid line represents the relationship currently participating in the update. In this paper, we conduct the asynchronous reasoning in the order of *entity-entity* → *entity-sentence* → *sentence-sentence* since the sentence nodes are used to reformulate the question and predict the support facts.

Context and question encoder. We concatenate the question Q with the context C and feed them to into a pre-trained BERT model to obtain representations $\boldsymbol{Q} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_m\} \in \mathbb{R}^{m \times d_1}$ and $\boldsymbol{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_n\} \in \mathbb{R}^{n \times d_1}$, 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 graph update and enhance the interactions of the question and the context, obtaining $\hat{\boldsymbol{Q}} \in \mathbb{R}^{m \times d_2}$ and $\hat{\boldsymbol{C}} \in \mathbb{R}^{n \times d_2}$, where $d_2 < d_1$.

Multi-step asynchronous reasoning module. First we construct a multi-grained graph $\mathcal{G} = (\boldsymbol{V}, \mathcal{R})$ where the node set $\boldsymbol{V} = \boldsymbol{E} \cup \boldsymbol{S}$. Entity node representations $\boldsymbol{E} \in \mathbb{R}^{n_e \times d_2}$ and sentence node representations $\boldsymbol{S} \in \mathbb{R}^{n_s \times d_2}$ are initialized with $\hat{\boldsymbol{C}}$ and connected by semantic relationships $\mathcal{R}$ (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:

$$\mathcal{G}^T, \boldsymbol{q}^T = \text{AMGN}(\mathcal{G}^0, \boldsymbol{q}^0) \tag{1}$$

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

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

$$\boldsymbol{O}_{sent} = \text{FFN}(\boldsymbol{S}^T) \tag{2}$$

where $\boldsymbol{O}_{sent} \in \mathbb{R}^{n_s}$. Following DFGN, we map the logits of the supporting facts $\boldsymbol{O}_{sent} \in \mathbb{R}^{n_s}$ to $\tilde{\boldsymbol{C}} \in \mathbb{R}^n$. Then $\tilde{\boldsymbol{C}}$ is concatenated with $\hat{\boldsymbol{C}}$ 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 $\hat{\boldsymbol{C}}$ to initialize entity and sentence nodes. Specifically, these representations are fed into a self-attentive pooling layer to calculate the node representations. Given the start and end positions $< p_{start}(v_i), p_{end}(v_i) >$ in the text span corresponding to the i-th node v_i, the initial node representation $\boldsymbol{v}_i^0$ is calculated as:

$$\boldsymbol{v}_i^0 = \sum_{k=p_{start}(v_i)}^{p_{end}(v_i)} \alpha_k * \hat{\boldsymbol{c}}_k \tag{3}$$

$$\alpha_k = \frac{\exp(\boldsymbol{w} \cdot \hat{\boldsymbol{c}}_k)}{\sum_{k=p_{start}(v_i)}^{p_{end}(v_i)} \exp(\boldsymbol{w} \cdot \hat{\boldsymbol{c}}_k)} \tag{4}$$

where $\boldsymbol{v}_i^0 \in \mathbb{R}^{d_2}$ and $\boldsymbol{w} \in \mathbb{R}^{d_2}$. 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 entity. Obviously, the graph depicts the three levels of relationships of entity-entity $\mathcal{R}_{ee}$ (i, ii), entity-sentence $\mathcal{R}_{es}$ (iii, iv), and sentence-sentence $\mathcal{R}_{ss}$ (v, vi). 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:

$$\boldsymbol{g}_i^t = \text{ELU}(\boldsymbol{W}^t[\boldsymbol{v}_i^{t-1}; \boldsymbol{q}^{t-1}]) \tag{5}$$

$$\hat{\boldsymbol{v}}_i^{t-1} = \boldsymbol{g}^t \odot \boldsymbol{v}_i^{t-1} \tag{6}$$

where $\boldsymbol{W}^t \in \mathbb{R}^{d_2 \times 2d_2}$, $\odot$ means the element-wise multiplication, and $[;]$ denotes the concatenation operation. The question-aware nodes $\hat{\boldsymbol{V}}^{t-1}$ 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:

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

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

where $\mathcal{N}_i^r$ represents the neighbors of the i-th node through the relation r, $\boldsymbol{W}_{\mathcal{R}_{ee}}^t \in \mathbb{R}^{d_2}$ is a parameter representing the entity-entity level relationship group, $\boldsymbol{W}_r^t \in \mathbb{R}^{2d_2}$ 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 attention weight differentiating each relationship while conduct information propagation at the relationship group level. The $\boldsymbol{u}_i^t$ is then added to $\hat{\boldsymbol{v}}_i^t$ to update the nodes at the current relationship level:

$$\hat{\boldsymbol{m}}_i^{t,ee} = \begin{cases} \hat{\boldsymbol{v}}_i^{t-1} + \boldsymbol{u}_i^t, & i \in \{x | \exists r \in \mathcal{R}_{ee}, \mathcal{N}_x^r \neq \emptyset\} \\ \hat{\boldsymbol{v}}_i^{t-1}, & otherwise \end{cases} \tag{9}$$

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

$$\hat{\boldsymbol{M}}^{t,ee} = \text{Update\_ee}(\hat{\boldsymbol{V}}^{t-1}) \tag{10}$$

For $\mathcal{R}_{es}$ and $\mathcal{R}_{ss}$, we perform the update asynchronously in the same way:

$$\hat{\boldsymbol{M}}^{t,es} = \text{Update\_es}(\hat{\boldsymbol{M}}^{t,ee}) \tag{11}$$

$$\boldsymbol{V}^t = \text{Update\_ss}(\hat{\boldsymbol{M}}^{t,es}) \tag{12}$$

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 $\mathcal{R}_{ee}$ → locating the sentence nodes to which these entities belong through $\mathcal{R}_{es}$ → comparing the descriptive information in these sentences through $\mathcal{R}_{ss}$ 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 $\boldsymbol{q}^{t-1}$ is reformulated with updated sentence nodes. The reformulation mechanism is implemented as follow:

$$\boldsymbol{q}^t = \text{GRU}(\boldsymbol{q}^{t-1}, \tilde{\boldsymbol{s}}^t) \tag{13}$$

$$\tilde{\boldsymbol{s}}^t = \sum_{i=1}^{n_s} \beta_i * \boldsymbol{s}_i^t \tag{14}$$

$$\beta_i = \frac{\exp(\boldsymbol{q}^{t-1} \cdot \boldsymbol{s}_i^t)}{\sum_{j=1}^{n_s} \exp(\boldsymbol{q}^{t-1} \cdot \boldsymbol{s}_j^t)} \tag{15}$$

The $\boldsymbol{q}^t$ and $\boldsymbol{V}^t$ 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 $\boldsymbol{V}^T$ 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/woshiyyya/DFGN-pytorch

Dataset	Model	Ans	Sup	Joint
Dev	DFGN[‡]	81.03	87.67	73.18
	SAE-large [Tu *et al.*, 2020]	80.75	87.38	72.75
	C2F Reader [Shao *et al.*, 2020]	-	-	73.93
	HGN-large [Fang *et al.*, 2020]	82.22	88.58	74.37
	AMGN (ours)	83.11	88.69	74.76
	AMGN+ (ours)	**83.46**	**89.13**	**75.48**
Test	Baseline [Yang *et al.*, 2018]	59.02	64.49	40.16
	QFE [Nishida *et al.*, 2019]	68.06	84.49	59.61
	DecompRC [Min *et al.*, 2019]	69.63	-	-
	DFGN [Qiu *et al.*, 2019]	69.69	81.62	59.82
	TAP 2 [Glass *et al.*, 2020]	78.59	85.57	69.12
	SAE-large [Tu *et al.*, 2020]	79.62	86.86	71.45
	C2F Reader [Shao *et al.*, 2020]	81.24	87.63	72.73
	Longformer [Beltagy *et al.*, 2020]	81.25	88.34	73.16
	ETC-large [Zaheer *et al.*, 2020]	81.18	**89.09**	73.62
	HGN-large [Fang *et al.*, 2020]	82.19	88.47	74.21
	SpiderNet-large[†]	83.02	88.85	74.88
	AMGN (ours)	82.79	88.12	74.20
	AMGN+ (ours)	**83.37**	88.83	**75.24**

Table 1: F1 performance comparison on the HotpotQA in the *Distractor* setting. (†) denotes unpublished work. (‡) is our re-implementation using the Roberta-large retriever and encoder in the fine-tuning setting.

Roberta-large model as the encoder. For graph construction, we employ DFGN pre-trained BERT-based NER model to extract entities including `Person`, `Number`, `Location` and `Organization`, etc. The numbers of entities and sentences in one graph are limited to 80 and 30, respectively. Since HotpotQA only requires two-hop reasoning, a two-step graph update is conducted thus T is 2. We finetune on the training set for 8 epochs, with batch size as 32. For optimization, We use BERTAdam with an initial learning rate of $2e^{-5}$. Since DFGN uses BERT in the feature-based setting, we leverage the Roberta-large retriever and encoder to re-implemente it in the fine-tuning setting for a fair comparison, which is similar to C2F Reader.

An enhanced variant. Since the definition of relationship group is more scalable for adding new edges, we investigate two new types of edges: 1) We add edges between entities in the question and other entities within the same paragraph to the relation group $\mathcal{R}_{ee}$; 2) We modify (v) to edges between sentence nodes within the same paragraph. This variant is called AMGN+, the same model described in Section 3 except for the above two relations.

4.2 Main Results

Table 1 shows published and unpublished models on both development and blind test set[3] of HotpotQA. Results on the dev set show that our AMGN variants outperform other Reborta-large based methods, indicating the performance gain comes from better relation and model design. Compared with HGN including paragraph nodes, the overall performance of AMGN is modest better. We believe that the asynchronous update sequence design involving more granularity can improve system performance. For AMGN+, we find that all metrics have been improve, which implies the scalability of group-based relationship definition. On the blind test set, our AMGN+ ranks No.1 at the time of submission (Jan 11, 2021).

Model	Ans F1	Sup F1	Jiont F1
AMGN	83.11	**88.69**	**74.76**
w/o SS Graph ($\boldsymbol{E}\&\mathcal{R}_{ss}$)	82.84	87.83	73.23
w/o EE Graph ($\boldsymbol{S}\&\mathcal{R}_{ee}$)	82.37	88.53	73.79
w/o $\mathcal{R}_{es}$	82.08	87.24	73.45
w/o Graph	80.27	85.78	70.96
Updating $\boldsymbol{q}$ with $\boldsymbol{E}$	**83.27**	88.04	74.22
Updating $\boldsymbol{q}$ with $\boldsymbol{E}\&\boldsymbol{S}$	82.85	88.37	74.30
Only one-step Update ($T=1$)	81.86	86.99	72.58
Global W^t in Eq. 7	82.84	87.75	73.63
w/o $W^t_{\mathcal{R}}$ in Eq. 7	82.57	87.43	72.99
w/o Question Reformulation	82.62	87.55	73.27

Table 2: Ablation Study.

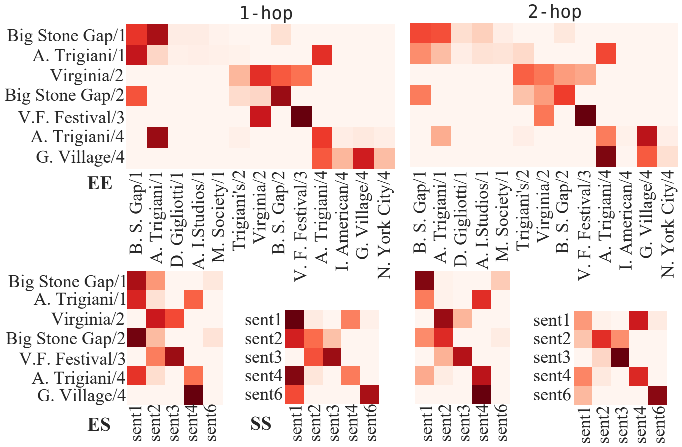
4.3 Ablation Study

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 $\mathcal{R}_{se}$, 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. $\mathcal{R}_{ss}$->$\mathcal{R}_{ee}$->$\mathcal{R}_{es}$). 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.

[3] https://hotpotqa.github.io/



Context
sent_1: Big Stone Gap is a 2014 American drama romantic comedy film written and directed by Adriana Trigiani and produced by Donna Gigliotti for Altar Identity Studios, a subsidiary of Media Society.
sent_2: Based on Trigiani's 2000 best-selling novel of the same name, the story is set in the actual Virginia town of Big Stone Gap circa 1970s.
sent_3: The film had its world premiere at the Virginia Film Festival on November 6, 2014.
sent_4: Adriana Trigiani is an Italian American best-selling author of sixteen books, television writer, film director, and entrepreneur based in Greenwich Village, New York City.
sent_5: Trigiani has published a novel a year since 2000.
sent_6: Clinton is a city in Big Stone County, Minnesota, United States.
sent_7: The city was named for New York Governor DeWitt Clinton.
sent_8: The population was 449 at the 2010 census.
Question: The director of the romantic comedy "Big Stone Gap" is based in what New York city?
Answer: Greenwich Village, New York City **Prediction:** Greenwich Village
Reasoning Chain (Entity/Sentence): Big Stone Gap/1 -> Adriana Trigiani/1 -> Adriana Trigiani/4 -> Greenwich Village/4

Figure 3: Case Study. The sample is extracted from the HotpotQA development set. The supporting facts are marked in red (*sent*_1 and *sent*_4). Bridge information is marked in blue. The underlined text is clues that may cause interference.

Multi-granular Update Order	Ans F1	Sup F1	Jiont F1
Simultaneous Update	82.65	87.51	73.34
$\mathcal{R}_{ss}$->$\mathcal{R}_{ee}$->$\mathcal{R}_{es}$	82.26	87.36	72.82
$\mathcal{R}_{se}$->$\mathcal{R}_{ss}$->$\mathcal{R}_{ee}$	82.51	87.43	73.05
$\mathcal{R}_{ee}$->$\mathcal{R}_{ss}$->$\mathcal{R}_{es}$	82.43	87.84	73.29
$\mathcal{R}_{ss}$->$\mathcal{R}_{es}$->$\mathcal{R}_{ee}$	**83.20**	87.95	74.16
$\mathcal{R}_{se}$->$\mathcal{R}_{ee}$->$\mathcal{R}_{ss}$	82.76	88.37	74.48
$\mathcal{R}_{ee}$->$\mathcal{R}_{es}$->$\mathcal{R}_{ss}$	83.11	**88.69**	**74.76**

Table 3: Results with different graph update order on the HotpotQA dev dataset in the *Distractor* setting.

Model	Bridge (79.91%)			Comparison (20.09%)		
	Ans F1	Sup F1	Joint F1	Ans F1	Sup F1	Joint F1
HGN	81.90	87.60	73.31	83.49	**92.49**	78.58
AMGN	83.10	87.91	73.85	83.24	91.83	78.37
AMGN+	**83.42**	**88.27**	**74.66**	**83.82**	92.17	**78.94**

Table 4: Performance comparison for different reasoning types.

4.5 Result Analysis

We provide result analysis based on the reasoning types officially annotated by HotpotQA to illustrate which reasoning is affected by our methods. HotpotQA provides two reasoning types: "bridge" and "comparison". The former requires the model to find bridge information before reaching the final answer, while the latter requires the model to compare the attributes of two entities and then give the answer. For a fair comparison, we mainly focus on published Roberta-large based HGN. As shown in Table 4, our AMGN achieves better results on the "bridge" reasoning type and is fairly competitive on the "comparison" reasoning type. Interestingly, the performance of two variants on the "bridge" type is higher than HGN. We hypothesize that our methods mainly work on sequential reasoning as both asynchronous update and question reformulation have obvious sequential logic. For the "comparison" task, there is still room for further improvement.

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$_{ee}$, Update$_{es}$ and Update$_{ss}$. 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*_4, "*Adriana Trigiani*" in *sent*_4 and *sent*_1, and *sent*_1 and *sent*_4. 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 retrieve reasoning paths over wikipedia graph for question answering. In *ICLR*, 2020.

[Beltagy *et al.*, 2020] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *CoRR*, abs/2004.05150, 2020.

[Cao *et al.*, 2019] Nicola De Cao, Wilker Aziz, and Ivan Titov. Question answering by reasoning across documents with graph convolutional networks. In *NAACL-HLT*, pages 2306–2317, 2019.

[Das *et al.*, 2019] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. Multi-step retriever-reader interaction for scalable open-domain question answering. In *ICLR*, 2019.

[Dhingra *et al.*, 2018] Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Neural models for reasoning over multiple mentions using coreference. In *NAACL-HLT*, pages 42–48, 2018.

[Ding *et al.*, 2019] Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. Cognitive graph for multi-hop reading comprehension at scale. In *ACL*, pages 2694–2703, 2019.

[Fang *et al.*, 2020] Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. Hierarchical graph network for multi-hop question answering. In *EMNLP*, pages 8823–8838, 2020.

[Feldman and El-Yaniv, 2019] Yair Feldman and Ran El-Yaniv. Multi-hop paragraph retrieval for open-domain question answering. In *ACL*, pages 2296–2309, 2019.

[Glass *et al.*, 2020] Michael R. Glass, Alfio Gliozzo, Rishav Chakravarti, Anthony Ferritto, Lin Pan, G. P. Shrivatsa Bhargav, Dinesh Garg, and Avirup Sil. Span selection pre-training for question answering. In *ACL*, pages 2773–2782, 2020.

[Jiang and Bansal, 2019] Yichen Jiang and Mohit Bansal. Self-assembling modular networks for interpretable multi-hop reasoning. In *EMNLP-IJCNLP*, pages 4473–4483, 2019.

[Jiang *et al.*, 2019] Yichen Jiang, Nitish Joshi, Yen-Chun Chen, and Mohit Bansal. Explore, propose, and assemble: An interpretable model for multi-hop reading comprehension. In *ACL*, pages 2714–2725, 2019.

[Khashabi *et al.*, 2018] Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *NAACL-HLT*, pages 252–262, 2018.

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[Min *et al.*, 2019] Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. Multi-hop reading comprehension through question decomposition and rescoring. In *ACL*, pages 6097–6109, 2019.

[Nie *et al.*, 2019] Yixin Nie, Songhe Wang, and Mohit Bansal. Revealing the importance of semantic retrieval for machine reading at scale. In *EMNLP-IJCNLP*, pages 2553–2566, 2019.

[Nishida *et al.*, 2019] 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*, pages 2335–2345, 2019.

[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.

[Qi *et al.*, 2019] Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D. Manning. Answering complex open-domain questions through iterative query generation. In *EMNLP-IJCNLP*, pages 2590–2602, 2019.

[Qi *et al.*, 2020] Peng Qi, Haejun Lee, Oghenetegiri ”TG” Sido, and Christopher D. Manning. Retrieve, rerank, read, then iterate: Answering open-domain questions of arbitrary complexity from text. *CoRR*, abs/2010.12527, 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 *ACL*, pages 6140–6150, 2019.

[Rajpurkar *et al.*, 2016] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392, 2016.

[Seo *et al.*, 2017] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *ICLR*, 2017.

[Shao *et al.*, 2020] Nan Shao, Yiming Cui, Ting Liu, Shijin Wang, and Guoping Hu. Is graph structure necessary for multi-hop question answering? In *EMNLP*, pages 7187–7192, 2020.

[Song *et al.*, 2018] 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. *CoRR*, abs/1809.02040, 2018.

[Tu *et al.*, 2019] Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. In *ACL*, pages 2704–2713, 2019.

[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 *AAAI*, pages 9073–9080, 2020.

[Welbl *et al.*, 2018] Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *TACL*, 6:287–302, 2018.

[Yadav *et al.*, 2020] Vikas Yadav, Steven Bethard, and Mihai Surdeanu. Unsupervised alignment-based iterative evidence retrieval for multi-hop question answering. In *ACL*, pages 4514–4525, 2020.

[Yang *et al.*, 2018] 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.

[Zaheer *et al.*, 2020] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *NeurIPS*, 2020.

[Zhang *et al.*, 2020] Yuyu Zhang, Ping Nie, Arun Ramamurthy, and Le Song. DDRQA: dynamic document reranking for open-domain multi-hop question answering. *CoRR*, abs/2009.07465, 2020.