Dynamically Fused Graph Network for Multi-hop Reasoning

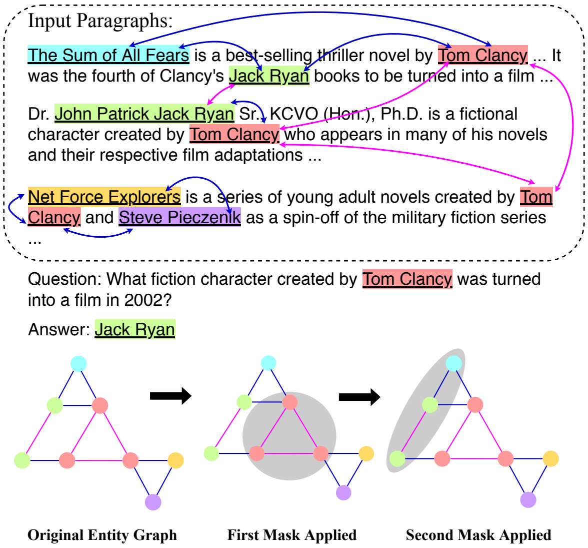
Lin Qiu[1†] Yunxuan Xiao[1†] Yanru Qu[1†]
Hao Zhou[2] Lei Li[2] Weinan Zhang[1] Yong Yu[1]
[1] Shanghai Jiao Tong University [2] ByteDance AI Lab, China
{lqiu, kevinqu, yyu}@apex.sjtu.edu.cn
{xiaoyunxuan, wnzhang}@sjtu.edu.cn
{zhouhao.nlp, lilei.02}@bytedance.com

Abstract

Text-based question answering (TBQA) has been studied extensively in recent years. Most existing approaches focus on finding the answer to a question within a single paragraph. However, many difficult questions require multiple supporting evidence from scattered text across two or more documents. In this paper, we propose the Dynamically Fused Graph Network (DFGN), a novel method to answer those questions requiring multiple scattered evidence and reasoning over them. Inspired by human's step-by-step reasoning behavior, DFGN includes a dynamic fusion layer that starts from the entities mentioned in the given query, explores along the entity graph dynamically built from the text, and gradually finds relevant supporting entities from the given documents. We evaluate DFGN on HotpotQA, a public TBQA dataset requiring multi-hop reasoning. DFGN achieves competitive results on the public board. Furthermore, our analysis shows DFGN could produce interpretable reasoning chains.

1 Introduction

Question answering (QA) has been a popular topic in natural language processing. QA provides a quantifiable way to evaluate an NLP system's capability on language understanding and reasoning (Hermann et al., 2015; Rajpurkar et al., 2016, 2018). Most previous work focus on finding evidence and answers from a single paragraph (Seo et al., 2016; Liu et al., 2017; Wang et al., 2017). It rarely tests deep reasoning capabilities of the underlying model. In fact, Min et al. (2018) observe that most questions in existing QA benchmarks can be answered by retrieving a small set of sentences without reasoning. To address this issue, there are several recently proposed QA datasets particularly designed to evaluate a system's multi-hop reasoning capabilities, including WikiHop (Welbl et al., 2018), ComplexWebQuestions (Talmor and Berant, 2018), and HotpotQA (Yang et al., 2018).


Input Paragraphs:

The Sum of All Fears is a best-selling thriller novel by Tom Clancy ... It was the fourth of Clancy's Jack Ryan books to be turned into a film ...

Dr. John Patrick Jack Ryan Sr., KCVO (Hon.), Ph.D. is a fictional character created by Tom Clancy who appears in many of his novels and their respective film adaptations ...

Net Force Explorers is a series of young adult novels created by Tom Clancy and Steve Pieczenik as a spin-off of the military fiction series ...

Question: What fiction character created by Tom Clancy was turned into a film in 2002?

Answer: Jack Ryan

Figure 1: Example of multi-hop text-based QA. One question and three document paragraphs are given. Our proposed DFGN conducts multi-step reasoning over the facts by constructing an entity graph from multiple paragraphs, predicting a dynamic mask to select a subgraph, propagating information along the graph, and finally transfer the information from the graph back to the text in order to localize the answer. Nodes are entity occurrences, with the color denoting the underlying entity. Edges are constructed from co-occurrences. The gray circles are selected by DFGN in each step.

In this paper, we study the problem of multi-hop text-based QA, which requires multi-hop reasoning among evidence scattered around multiple raw documents. In particular, a query utterance and a set of accompanying documents are given, but not

[†]These authors contributed equally. The order of authorship is decided through dice rolling. Work done while Lin Qiu was a research intern in ByteDance AI Lab.

all of them are relevant. The answer can only be obtained by selecting two or more evidence from the documents and inferring among them (see Figure 1 for an example). This setup is versatile and does not rely on any additional predefined knowledge base. Therefore the models are expected to generalize well and to answer questions in open domains.

There are two main challenges to answer questions of this kind. Firstly, since not every document contain relevant information, multi-hop text-based QA requires filtering out noises from multiple paragraphs and extracting useful information. To address this, recent studies propose to build entity graphs from input paragraphs and apply graph neural networks (GNNs) to aggregate the information through entity graphs (Dhingra et al., 2018; De Cao et al., 2018; Song et al., 2018a). However, all of the existing work apply GNNs based on a static global entity graph of each QA pair, which can be considered as performing implicit reasoning. Instead of them, we argue that the query-guided multi-hop reasoning should be explicitly performed on a dynamic local entity graph tailored according to the query.

Secondly, previous work on multi-hop QA (e.g. WikiHop) usually aggregates document information to an entity graph, and answers are then directly selected on entities of the entity graph. However, in a more realistic setting, the answers may even not reside in entities of the extracted entity graph. Thus, existing approaches can hardly be directly applied to open-domain multi-hop QA tasks like HotpotQA.

In this paper, we propose Dynamically Fused Graph Network (DFGN), a novel method to address the aforementioned concerns for multi-hop text-based QA. For the first challenge, DFGN constructs a *dynamic entity graph* based on entity mentions in the query and documents. This process iterates in multiple rounds to achieve multi-hop reasoning. In each round, DFGN generates and reasons on a dynamic graph, where irrelevant entities are masked out while only reasoning sources are preserved, via a mask prediction module. Figure 1 shows how DFGN works on a multi-hop text-based QA example in HotpotQA. The mask prediction module is learned in an end-to-end fashion, alleviating the error propagation problem.

To solve the second challenge, we propose a *fusion process* in DFGN to solve the unrestricted QA challenge. We not only aggregate information from documents to the entity graph (doc2graph), but also propagate the information of the entity graph back to document representations (graph2doc). The fusion process is iteratively performed at each hop through the document tokens and entities, and the final resulting answer is then obtained from document tokens. The *fusion process* of doc2graph and graph2doc along with the *dynamic entity graph* jointly improve the interaction between the information of documents and the entity graph, leading to a less noisy entity graph and thus more accurate answers.

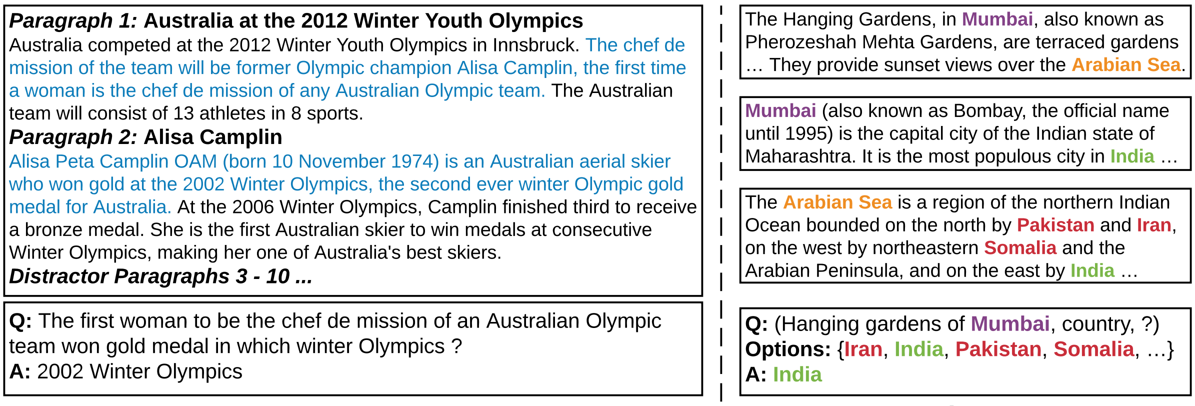
As one merit, DFGN's predicted masks implicitly induce reasoning chains, which can explain the reasoning results. Since the ground truth reasoning chain is very hard to define and label for open-domain corpus, we propose a feasible way to weakly supervise the mask learning. We propose a new metric to evaluate the quality of predicted reasoning chains and constructed entity graphs.

Our contributions are summarized as follows:

- We propose DFGN, a novel method for the multi-hop text-based QA problem.
- We provide a way to explain and evaluate the reasoning chains via interpreting the entity graph masks predicted by DFGN. The mask prediction module is additionally weakly trained.
- We provide an experimental study on a public dataset (HotpotQA) to demonstrate that our proposed DFGN is competitive against state-of-the-art unpublished work.

2 Related work

Text-based Question Answering Depending on whether the supporting information is structured or not, QA tasks can be categorized into knowledge-based (KBQA), text-based (TBQA), mixed, and others. In KBQA, the supporting information is from structured knowledge bases (KBs), while the queries can be either structure or natural language utterances. For example, SimpleQuestions is one large scale dataset of this kind (Bordes et al., 2015). In contrast, TBQA's supporting information is raw text, and hence the query is also text. SQuAD (Rajpurkar et al., 2016) and HotpotQA (Yang et al., 2018) are two such datasets. There are also mixed QA tasks which combine both text and KBs, e.g. WikiHop (Welbl



Paragraph 1: **Australia at the 2012 Winter Youth Olympics**
Australia competed at the 2012 Winter Youth Olympics in Innsbruck. The chef de mission of the team will be former Olympic champion Alisa Camplin, the first time a woman is the chef de mission of any Australian Olympic team. The Australian team will consist of 13 athletes in 8 sports.

Paragraph 2: **Alisa Camplin**
Alisa Peta Camplin OAM (born 10 November 1974) is an Australian aerial skier who won gold at the 2002 Winter Olympics, the second ever winter Olympic gold medal for Australia. At the 2006 Winter Olympics, Camplin finished third to receive a bronze medal. She is the first Australian skier to win medals at consecutive Winter Olympics, making her one of Australia's best skiers.

Distractor Paragraphs 3 - 10 ...

Q: The first woman to be the chef de mission of an Australian Olympic team won gold medal in which winter Olympics ?
A: 2002 Winter Olympics

HotpotQA

The Hanging Gardens, in **Mumbai**, also known as Pherozeshah Mehta Gardens, are terraced gardens … They provide sunset views over the **Arabian Sea**.

Mumbai (also known as Bombay, the official name until 1995) is the capital city of the Indian state of Maharashtra. It is the most populous city in **India** …

The **Arabian Sea** is a region of the northern Indian Ocean bounded on the north by **Pakistan** and **Iran**, on the west by northeastern **Somalia** and the Arabian Peninsula, and on the east by **India** …

Q: (Hanging gardens of **Mumbai**, country, ?)
Options: {**Iran**, **India**, **Pakistan**, **Somalia**, …}
A: **India**

WikiHop

Figure 2: Comparison between HotpotQA (left) and WikiHop (right). In HotpotQA, the questions are proposed by crowd workers and the blue words in paragraphs are labeled supporting facts corresponding to the question. In WikiHop, the questions and answers are formed with relations and entities in the underlying KB respectively, thus the questions are inherently restricted by the KB schema. The colored words and phrases are entities in the KB.

et al., 2018) and ComplexWebQuestions (Talmor and Berant, 2018). In this paper, we focus on TBQA, since TBQA tests a system's end-to-end capability of extracting relevant facts from raw language and reasoning about them.

Depending on the complexity in underlying reasoning, QA problems can be categorized into single-hop and multi-hop ones. Single-hop QA only requires one fact extracted from the underlying information, no matter structured or unstructured, e.g. "which city is the capital of California". The SQuAD dataset belongs to this type (Rajpurkar et al., 2016). On the contrary, multi-hop QA requires identifying multiple related facts and reasoning about them, e.g. "what is the capital city of the largest state in the U.S.". Example tasks and benchmarks of this kind include WikiHop, ComplexWebQuestions, and HotpotQA. Many IR techniques can be applied to answer single-hop questions (Rajpurkar et al., 2016). However, these IR techniques are hardly introduced in multi-hop QA, since a single fact can only partially match a question.

Note that existing multi-hop QA datasets WikiHop and ComplexWebQuestions, are constructed using existing KBs and constrained by the schema of the KBs they use. For example, the answers are limited in entities in WikiHop rather than formed by free texts in HotpotQA (see Figure 2 for an example). In this work, we focus on multi-hop text-based QA, so we only evaluate on HotpotQA.

Multi-hop Reasoning for QA Popular GNN frameworks, e.g. graph convolution network (Kipf and Welling, 2017), graph attention network (Veličković et al., 2018), and graph recurrent network (Song et al., 2018b), have been previously studied and show promising results in QA tasks requiring reasoning (Dhingra et al., 2018; De Cao et al., 2018; Song et al., 2018a).

Coref-GRN extracts and aggregates entity information in different references from scattered paragraphs (Dhingra et al., 2018). Coref-GRN utilizes co-reference resolution to detect different mentions of the same entity. These mentions are combined with a graph recurrent neural network (GRN) (Song et al., 2018b) to produce aggregated entity representations. MHQA-GRN (Song et al., 2018a) follows Coref-GRN and refines the graph construction procedure with more connections: sliding-window, same entity, and co-reference, which shows further improvements. Entity-GCN (De Cao et al., 2018) proposes to distinguish different relations in the graphs through a relational graph convolutional neural network (GCN) (Kipf and Welling, 2017). Coref-GRN, MHQA-GRN and Entity-GCN explore the graph construction problem in answering real-world questions. However, it is yet to investigate how to effectively reason about the constructed graphs, which is the main problem studied in this work.

Another group of sequential models deals with multi-hop reasoning following Memory Networks (Sukhbaatar et al., 2015). Such models construct representations for queries and memory cells for contexts, then make interactions between them in a multi-hop manner. Munkhdalai and Yu (2017) and Onishi et al. (2016) incorporate a hypothesis testing loop to update the query representation at each reasoning step and select the best answer among the candidate entities at the last step. IR-Net (Zhou et al., 2018) generates a subject state and a relation state at each step, computing the similarity score between all the entities and relations given by the dataset KB. The ones with the highest score at each time step are linked together to form an interpretable reasoning chain. However, these models perform reasoning on simple synthetic datasets with a limited number of entities and relations, which are quite different with large-scale QA dataset with complex questions. Also, the supervision of entity-level reasoning chains in synthetic datasets can be easily given following some patterns while they are not available in HotpotQA.

3 Dynamically Fused Graph Network

We describe dynamically fused graph network (DFGN) in this section. Our intuition is drawn from the human reasoning process for QA. One starts from an entity of interest in the query, focuses on the words surrounding the start entities, connects to some related entity either found in the neighborhood or linked by the same surface mention, repeats the step to form a reasoning chain, and lands on some entity or snippets likely to be the answer. To mimic human reasoning behavior, we develop five components in our proposed QA system (Fig. 3): a paragraph selection subnetwork, a module for entity graph construction, an encoding layer, a fusion block for multi-hop reasoning, and a final prediction layer.

3.1 Paragraph Selection

For each question, we assume that N_p paragraphs are given (e.g. $N_p = 10$ in HotpotQA). Since not every piece of text is relevant to the question, we train a sub-network to select relevant paragraphs. The sub-network is based on a pre-trained BERT model (Devlin et al., 2018) followed by a sentence classification layer with sigmoid prediction. The selector network takes a query Q and a paragraph as input and outputs a relevance score between 0 and 1. Training labels are constructed by assigning 1's to the paragraphs with at least one supporting sentence for each Q&A pair. During inference, paragraphs with predicted scores greater than η ($= 0.1$ in experiments) are selected and concatenated together as the context C. η is properly chosen to ensure the selector reaches a significantly high recall of relevant paragraphs. Q and C are further processed by upper layers.

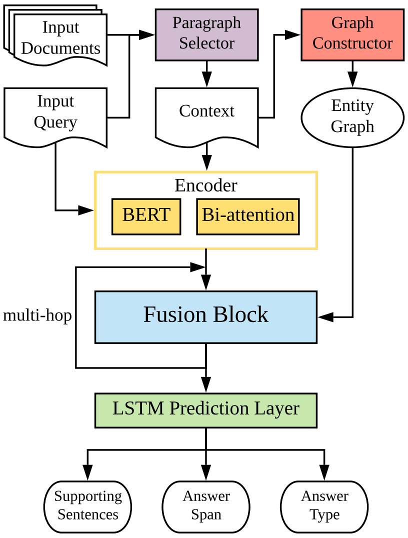


Figure 3: Overview of DFGN.

3.2 Constructing Entity Graph

We do not assume a global knowledge base. Instead, we use the Stanford corenlp toolkit (Manning et al., 2014) to recognize named entities from the context C. The number of extracted entities is denoted as N. The entity graph is constructed with the entities as nodes and edges built as follows. The edges are added 1. for every pair of entities appear in the same sentence in C (sentence-level links); 2. for every pair of entities with the same mention text in C (context-level links); and 3. between a central entity node and other entities within the same paragraph (paragraph-level links). The central entities are extracted from the title sentence for each paragraph. Notice the context-level links ensures that entities across multiple documents are connected in a certain way. We do not apply co-reference resolution for pronouns because it introduces both additional useful and erroneous links.

3.3 Encoding Query and Context

We concatenate the query Q with the context C and pass the resulting sequence to a pre-trained BERT model to obtain representations $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_L] \in \mathbb{R}^{L \times d_1}$ and $\mathbf{C}^\top = [\mathbf{c}_1, \ldots, \mathbf{c}_M] \in \mathbb{R}^{M \times d_1}$, where L,M are lengths of query and context, and d_1 is the size of BERT hidden states. In experiments, we find concatenating queries and contexts performs better than passing them separately to BERT.

The representations are further passed through a bi-attention layer (Seo et al., 2016) to enhance cross interactions between the query and the context. In practice, we find adding the bi-attention layer achieves better performance than the BERT encoding only. The output representation are $\mathbf{Q}_0 \in \mathbb{R}^{L \times d_2}$ and $\mathbf{C}_0 \in \mathbb{R}^{M \times d_2}$, where d_2 is the output embedding size.

3.4 Reasoning with the Fusion Block

With the embeddings calculated for the query Q and context C, the remaining challenge is how to identify supporting entities and the text span of potential answers. We propose a fusion block to mimic human's one-step reasoning behavior – starting from $\mathbf{Q}_0$ and $\mathbf{C}_0$ and finding one-step supporting entities. A fusion block achieves the following: 1. passing information from tokens to entities by computing entity embeddings from tokens (Doc2Graph flow); 2. propagating information on entity graph; and 3. passing information from entity graph to document tokens since the final prediction is on tokens (Graph2Doc flow). Fig. 4 depicts the inside structure of the fusion block in DFGN.

Document to Graph Flow. Since each entity is recognized via the NER tool, the text spans associated with the entities are utilized to compute entity embeddings (Doc2Graph). To this end, we construct a binary matrix $\mathbf{M}$, where $\mathbf{M}_{i,j}$ is 1 if i-th token in the context is within the span of the j-th entity. $\mathbf{M}$ is used to select the text span associated with an entity. The token embeddings calculated from the above section (which is a matrix containing only selected columns of $\mathbf{C}_{t-1}$) is passed into a mean-max pooling to calculate entity embeddings $\mathbf{E}_{t-1} = [\mathbf{e}_{t-1,1}, \ldots, \mathbf{e}_{t-1,N}]$. $\mathbf{E}_{t-1}$ will be of size $2d_2 \times N$, where N is the number of entities, and each of the $2d_2$ dimensions will produce both mean-pooling and max-pooling results. This module is denoted as Tok2Ent.

Dynamic Graph Attention. After obtaining entity embeddings from the input context $\mathbf{C}_{t-1}$, we apply a graph neural network to propagate node information to their neighbors. We propose a dynamic graph attention mechanism to mimic human's step-by-step exploring and reasoning behavior. In each reasoning step, we assume every node has some information to disseminate to neighbors. The more relevant to the query, the neighbor nodes receive more information from nearby.

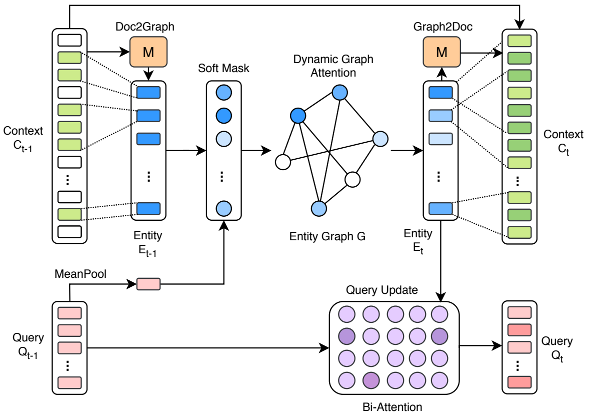


Figure 4: Reasoning with the fusion block in DFGN

We first identify nodes relevant to the query by creating a soft mask on entities. It serves as an information gatekeeper, i.e. only those entity nodes pertaining to the query are allowed to disseminate information. We use an attention network between the query embeddings and the entity embeddings to predict a soft mask $\mathbf{m}_t$, which aims to signify the start entities in the t-th reasoning step:

$$\tilde{\mathbf{q}}^{(t-1)} = \text{MeanPooling}(\mathbf{Q}^{(t-1)}) \tag{1}$$

$$\gamma_i^{(t)} = \tilde{\mathbf{q}}^{(t-1)} \mathbf{V}^{(t)} \mathbf{e}_i^{(t-1)} / \sqrt{d_2} \tag{2}$$

$$\mathbf{m}^{(t)} = \sigma([\gamma_1^{(t)}, \cdots, \gamma_N^{(t)}]) \tag{3}$$

$$\tilde{\mathbf{E}}^{(t-1)} = [m_1^{(t)} \mathbf{e}_1^{(t-1)}, \ldots, m_N^{(t)} \mathbf{e}_N^{(t-1)}] \tag{4}$$

where $\mathbf{V}_t$ is a linear projection matrix, and σ is the sigmoid function. By multiplying the soft mask and the initial entity embeddings, the desired start entities will be encouraged and others will be penalized. As a result, this step of information propagation is restricted to a dynamic sub-part of the entity graph.

The next step is to disseminate information across the dynamic sub-graph. Inspired by GAT (Veličković et al., 2018), we compute attention score α between two entities by:

$$\mathbf{h}_i^{(t)} = \mathbf{U}_t \tilde{\mathbf{e}}_i^{(t-1)} + \mathbf{b}_t \tag{5}$$

$$\beta_{i,j}^{(t)} = \text{LeakyReLU}(\mathbf{W}_t^\top [\mathbf{h}_i^{(t)}, \mathbf{h}_j^{(t)}]) \tag{6}$$

$$\alpha_{i,j}^{(t)} = \frac{\exp(\beta_{i,j}^{(t)})}{\sum_k \exp(\beta_{i,k}^{(t)})} \tag{7}$$

where $\mathbf{U}_t \in \mathbb{R}^{d_2 \times 2d_2}$, $\mathbf{W}_t \in \mathbb{R}^{2d_2}$ are linear projection parameters. Here the i-th row of α represents the proportion of information that will be assigned to the neighbors of entity i.

Note that the information flow in our model is different from most previous GATs. In dynamic graph attention, each node sums over its column, which forms a new entity state containing the total information it received from the neighbors:

$$\mathbf{e}_i^{(t)} = \text{ReLU}(\sum_{j \in B_i} \alpha_{j,i}^{(t)} \mathbf{h}_j^{(t)}) \tag{8}$$

where B_i is the set of neighbors of entity i. Then we obtain the updated entity embeddings $\mathbf{E}^{(t)} = [\mathbf{e}_1^{(t)}, \ldots, \mathbf{e}_N^{(t)}]$.

Updating Query. A reasoning chain contains multiple steps, and the newly visited entities by one step will be the start entities of the next step. In order to predict the expected start entities for the next step, we introduce a query update mechanism, where the query embeddings are updated by the entity embeddings of the current step. In our implementation, we utilize a bi-attention network (Seo et al., 2016) to update the query embeddings:

$$\mathbf{Q}^{(t)} = \text{Bi-Attention}(\mathbf{Q}^{(t-1)}, \mathbf{E}^{(t)}) \tag{9}$$

Graph to Document Flow. Using Tok2Ent and dynamic graph attention, we realize a reasoning step at the entity level. However, the unrestricted answer still cannot be backtraced. To address this, we develop a Graph2Doc module to keep information flowing from entity back to tokens in the context. Therefore the text span pertaining to the answers can be localized in the context.

Using the same binary matrix $\mathbf{M}$ as described above, the previous token embeddings in $\mathbf{C}_{t-1}$ are concatenated with the associated entity embedding corresponding to the token. Each row in $\mathbf{M}$ corresponds to one token, therefore we use it to select one entity's embedding from $\mathbf{E}_t$ if the token participates in the entity's mention. This information is further processed with a LSTM layer (Hochreiter and Schmidhuber, 1997) to produce the next-level context representation:

$$\mathbf{C}^{(t)} = \text{LSTM}([\mathbf{C}^{(t-1)}, \mathbf{M}\mathbf{E}^{(t)\top}]) \tag{10}$$

where ; refers to concatenation and $\mathbf{C}^{(t)} \in \mathbb{R}^{M \times d_2}$ serves as the input of the next fusion block. At this time, the reasoning information of current subgraph has been propagated onto the whole context.

3.5 Prediction

We follow the same structure of prediction layers as (Yang et al., 2018). The framework has four output dimensions, including 1. supporting sentences, 2. the start position of the answer, 3. the end position of the answer, and 4. the answer type. We use a cascade structure to solve the output dependency, where four isomorphic LSTMs $\mathcal{F}_i$ are stacked layer by layer. The context representation of the last fusion block is sent to the first LSTM $\mathcal{F}_0$. Each $\mathcal{F}_i$ outputs a logit $\mathbf{O} \in \mathbb{R}^{M \times d_2}$ and computes a cross entropy loss over these logits.

$$\mathbf{O}_{sup} = \mathcal{F}_0(\mathbf{C}^{(t)}) \tag{11}$$

$$\mathbf{O}_{start} = \mathcal{F}_1([\mathbf{C}^{(t)}, \mathbf{O}_{sup}]) \tag{12}$$

$$\mathbf{O}_{end} = \mathcal{F}_2([\mathbf{C}^{(t)}, \mathbf{O}_{sup}, \mathbf{O}_{start}]) \tag{13}$$

$$\mathbf{O}_{type} = \mathcal{F}_3([\mathbf{C}^{(t)}, \mathbf{O}_{sup}, \mathbf{O}_{end}]) \tag{14}$$

We jointly optimize these four cross entropy losses. Each loss term is weighted by a coefficient.

$$\mathcal{L} = \mathcal{L}_{start} + \mathcal{L}_{end} + \lambda_s \mathcal{L}_{sup} + \lambda_t \mathcal{L}_{type} \tag{15}$$

Weak Supervision. In addition, we introduce a weakly supervised signal to induce the soft masks at each fusion block to match the heuristic masks. For each training case, the heuristic masks contain a start mask detected from the query, and additional BFS masks obtained by applying breadth-first search (BFS) on the adjacent matrices give the start mask. A binary cross entropy loss between the predicted soft masks and the heuristics is then added to the objective. We skip those cases whose start masks cannot be detected from the queries.

4 Experiments

We evaluate our Dynamically Fused Graph Network (DFGN) on HotpotQA (Yang et al., 2018) in the distractor setting. For the full wiki setting where the entire Wikipedia articles are given as input, we consider the bottleneck is about information retrieval, thus we do not include the full wiki setting in our experiments.

4.1 Implementation Details

In paragraph selection stage, we use the uncased version of BERT Tokenizer (Devlin et al., 2018) to tokenize all passages and questions. The encoding vectors of sentence pairs are generated from a pre-trained BERT model (Devlin et al., 2018). We set a relatively low threshold during selection to

Model	Answer		Sup Fact		Joint	
	EM	F1	EM	F1	EM	F1
Baseline Model	45.60	59.02	20.32	64.49	10.83	40.16
GRN*	52.92	66.71	52.37	84.11	31.77	58.47
DFGN(Ours)	55.17	68.49	49.85	81.06	31.87	58.23
QFE*	53.86	68.06	**57.75**	**84.49**	**34.63**	59.61
DFGN(Ours)†	**56.31**	**69.69**	51.50	81.62	33.62	**59.82**

Table 1: Performance comparison on the private test set of HotpotQA in the distractor setting. Our DFGN is the second best result on the leaderboard before submission (on March 1st). The baseline model is from Yang et al. (2018) and the results with * is unpublished. DFGN(Ours)† refers to the same model with a revised entity graph, whose entities are recognized by a BERT NER model. Note that the result of DFGN(Ours)† is submitted to the leaderboard during the review process of our paper.

Setting	EM	F1
DFGN (2-layer)	55.42	69.23
- BFS Supervision	54.48	68.15
- Entity Mask	54.64	68.25
- Query Update	54.44	67.98
- E2T Process	53.91	67.45
- 1 Fusion Block	54.14	67.70
- 2 Fusion Blocks	53.44	67.11
- 2 Fusion Blocks & Bi-attn	50.03	62.83
gold paragraphs only	55.67	69.15
supporting facts only	57.57	71.67

Table 2: Ablation study of question answering performances in the development set of HotpotQA in the distractor setting. We use a DFGN with 2-layer fusion blocks as the origin model. The upper part is the model ablation results and the lower part is the dataset ablation results.

keep a high recall (97%) and a reasonable precision (69%) on supporting facts.

In graph construction stage, we use a pre-trained NER model from Stanford CoreNLP Toolkits[1] (Manning et al., 2014) to extract named entities. The maximum number of entities in a graph is set to be 40. Each entity node in the entity graphs has an average degree of 3.52.

In the encoding stage, we also use a pre-trained BERT model as the encoder, thus d_1 is 768. All the hidden state dimensions d_2 are set to 300. We set the dropout rate for all hidden units of LSTM and dynamic graph attention to 0.3 and 0.5 respectively. For optimization, we use Adam Optimizer (Kingma and Ba, 2015) with an initial learning rate of 1e^{-4}.

4.2 Main Results

We first present a comparison between baseline models and our DFGN[2]. Table 1 shows the performance of different models in the private test set of HotpotQA. From the table we can see that our model achieves the second best result on the leaderboard now[3] (on March 1st). Besides, the answer performance and the joint performance of our model are competitive against state-of-the-art unpublished models. We also include the result of our model with a revised entity graph whose entities are recognized by a BERT NER model (Devlin et al., 2018). We fine-tune the pre-trained BERT model on the dataset of the CoNLL'03 NER shared task (Sang and De Meulder, 2003) and use it to extract named entities from the input paragraphs. The results show that our model achieves a 1.5% gain in the joint F1-score with the entity graph built from a better entity recognizer.

To evaluate the performance of different components in our DFGN, we perform ablation study on both model components and dataset segments. Here we follow the experiment setting in Yang et al. (2018) to perform the dataset ablation study, where we only use golden paragraphs or supporting facts as the input context. The ablation results of QA performances in the development set of HotpotQA are shown in Table 2. From the table we can see that each of our model components can provide from 1% to 2% relative gain over the QA performance. Particularly, using a 1-layer fusion block leads to an obvious performance loss, which implies the significance of performing multi-hop reasoning in HotpotQA. Besides, the dataset abla-

[1] https://nlp.stanford.edu/software/CRF-NER.shtml

[2] Our code is available in https://github.com/woshiyyya/DFGN-pytorch.

[3] The leaderboard can be found on https://hotpotqa.github.io

tion results show that our model is not very sensitive to the noisy paragraphs comparing with the baseline model which can achieve a more than 5% performance gain in the "gold paragraphs only" and "supporting facts only" settings. (Yang et al., 2018).

4.3 Evaluation on Graph Construction and Reasoning Chains

The chain of reasoning is a directed path on the entity graph, so high-quality entity graphs are the basis of good reasoning. Since the limited accuracy of NER model and the incompleteness of our graph construction, 31.3% of the cases in the development set are unable to perform a complete reasoning process, where at least one supporting sentence is not reachable through the entity graph, i.e. no entity is recognized by NER model in this sentence. We name such cases as "missing supporting entity", and the ratio of such cases can evaluate the quality of graph construction. We focus on the rest 68.7% good cases in the following analysis.

In the following, we first give several definitions before presenting ESP (Entity-level Support) scores.

Path A path is a sequence of entities visited by the fusion blocks, denoting as $P = [e_{p_1}, \ldots, e_{p_{t+1}}]$ (suppose t-layer fusion blocks).

Path Score The score of a path is acquired by multiplying corresponding soft masks and attention scores along the path, i.e. $score(P) = \prod_{i=1}^{t} m_{p_i}^{(i)} \alpha_{p_i,p_{i+1}}^{(i)}$ (Eq. (3), (7)).

Hit Given a path and a supporting sentence, if at least one entity of the supporting sentence is visited by the path, we call this supporting sentence is hit[4].

Given a case with m supporting sentences, we select the top-k paths with the highest scores as the predicted reasoning chains. For each supporting sentence, we use the k paths to calculate how many supporting sentences are hit.

In the following, we introduce two metrics to evaluate the quality of multi-hop reasoning through entity-level supporting (ESP) scores.

k	1	2	5	10
ESP EM($\leq$ 40)	**7.4%**	**15.5%**	29.8%	41.0%
ESP EM($\leq$ 80)	7.1%	14.7%	**29.9%**	**44.8%**
ESP Recall($\leq$ 40)	**37.3%**	**46.1%**	58.4%	66.4%
ESP Recall($\leq$ 80)	34.9%	44.6%	**59.1%**	**70.0%**

Table 3: Evaluation of reasoning chains by ESP scores on two versions of the entity graphs in the development set. $\leq$ 40 and $\leq$ 80 indicate to the maximum number of nodes in entity graphs. Note that $\leq$ 40 refers to the entity graph whose entities are extracted by Stanford CoreNLP, while $\leq$ 80 refers to the entity graph whose entities are extracted by the aforementioned BERT NER model.

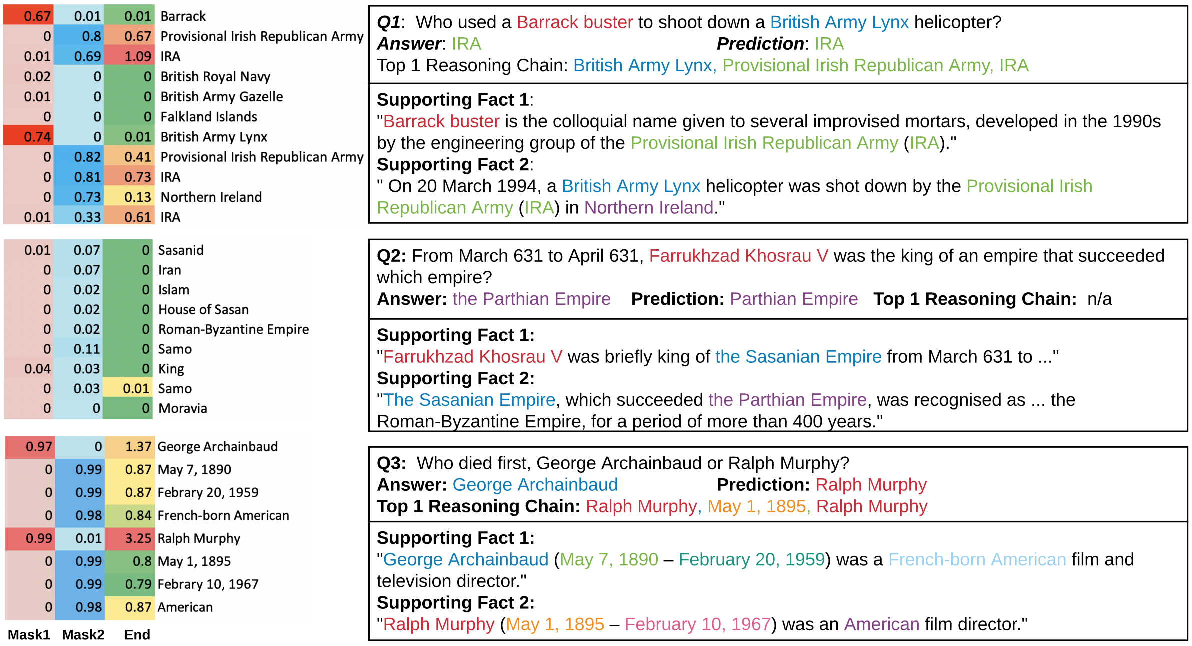
ESP EM (Exact Match) For a case with m supporting sentences, if all the m sentences are hit, we call this case exact match. The ESP EM score is the ratio of exactly matched cases.

ESP Recall For a case with m supporting sentences and h of them are hit, this case has a recall score of h/m. The averaged recall of the whole dataset is the ESP Recall.

We train a DFGN with 2 fusion blocks to select paths with top-k scores. In the development set, the average number of paths of length 2 is 174.7. We choose k as $1, 2, 5, 10$ to compute ESP EM and ESP Recall scores. As we can see in Table 3, regarding the supporting sentences as the ground truth of reasoning chains, our framework can predict reliable information flow. The most informative flow can cover the supporting facts and help produce reliable reasoning results. Here we present the results from two versions of the entity graphs. The results with a maximum number of nodes $\leq$ 40 are from the entity graph whose entities are extracted by Stanford CoreNLP. The results with a maximum number of nodes $\leq$ 80 are from the entity graph whose entities are extracted by the aforementioned BERT NER model. Since the BERT NER model performs better, we use a larger maximum number of nodes.

In addition, as the size of an entity graph gets larger, the expansion of reasoning chain space makes a Hit even more difficult. However, the BERT NER model still keeps comparative and even better performance on metrics of EM and Recall. Thus the entity graph built from the BERT NER model is better than the previous version.

[4] A supporting sentence may contain irrelevant information, thus we do not have to visit all entities in a supporting sentence. Besides, due to the fusion mechanism of DFGN, the entity information will be propagated to the whole sentence. Therefore, we define a "hit" occurs when at least one entity of the supporting sentence is visited.



Mask1	Mask2	End	
0.67	0.01	0.01	Barrack
0	0.8	0.67	Provisional Irish Republican Army
0.01	0.69	1.09	IRA
0.02	0	0	British Royal Navy
0.01	0	0	British Army Gazelle
0	0	0	Falkland Islands
0.74	0	0.01	British Army Lynx
0	0.82	0.41	Provisional Irish Republican Army
0	0.81	0.73	IRA
0	0.73	0.13	Northern Ireland
0.01	0.33	0.61	IRA

Q1: Who used a Barrack buster to shoot down a British Army Lynx helicopter?
Answer: IRA ***Prediction***: IRA
Top 1 Reasoning Chain: British Army Lynx, Provisional Irish Republican Army, IRA

Supporting Fact 1:
"Barrack buster is the colloquial name given to several improvised mortars, developed in the 1990s by the engineering group of the Provisional Irish Republican Army (IRA)."
Supporting Fact 2:
" On 20 March 1994, a British Army Lynx helicopter was shot down by the Provisional Irish Republican Army (IRA) in Northern Ireland."

Mask1	Mask2	End	
0.01	0.07	0	Sasanid
0	0.07	0	Iran
0	0.02	0	Islam
0	0.02	0	House of Sasan
0	0.02	0	Roman-Byzantine Empire
0	0.11	0	Samo
0.04	0.03	0	King
0	0.03	0.01	Samo
0	0	0	Moravia

Q2: From March 631 to April 631, Farrukhzad Khosrau V was the king of an empire that succeeded which empire?
Answer: the Parthian Empire **Prediction:** Parthian Empire **Top 1 Reasoning Chain:** n/a

Supporting Fact 1:
"Farrukhzad Khosrau V was briefly king of the Sasanian Empire from March 631 to ..."
Supporting Fact 2:
"The Sasanian Empire, which succeeded the Parthian Empire, was recognised as ... the Roman-Byzantine Empire, for a period of more than 400 years."

Mask1	Mask2	End	
0.97	0	1.37	George Archainbaud
0	0.99	0.87	May 7, 1890
0	0.99	0.87	Febrary 20, 1959
0	0.98	0.84	French-born American
0.99	0.01	3.25	Ralph Murphy
0	0.99	0.8	May 1, 1895
0	0.99	0.79	Febrary 10, 1967
0	0.98	0.87	American

Q3: Who died first, George Archainbaud or Ralph Murphy?
Answer: George Archainbaud **Prediction:** Ralph Murphy
Top 1 Reasoning Chain: Ralph Murphy, May 1, 1895, Ralph Murphy

Supporting Fact 1:
"George Archainbaud (May 7, 1890 – February 20, 1959) was a French-born American film and television director."
Supporting Fact 2:
"Ralph Murphy (May 1, 1895 – February 10, 1967) was an American film director."

Figure 5: Case study of three samples in the development set. We train a DFGN with 2-layer fusion blocks to produce the results. The numbers on the left side indicate the importance scores of the predicted masks. The text on the right side include the queries, answers, predictions, predicted top-1 reasoning chains and the supporting facts of three samples with the recognized entities highlighted by different colors.

4.4 Case Study

We present a case study in Figure 5. The first case illustrates the reasoning process in a DFGN with 2-layer fusion blocks. At the first step, by comparing the query with entities, our model generates **Mask1** as the start entity mask of reasoning, where *"Barrack"* and *"British Army Lynx"* are detected as the start entities of two reasoning chains. Information of two start entities is then passed to their neighbors on the entity graph. At the second step, mentions of the same entity *"IRA"* are detected by **Mask2**, serving as a bridge for propagating information across two paragraphs. Finally, two reasoning chains are linked together by the bridge entity *"IRA"*, which is exactly the answer.

The second case in Figure 5 is a bad case. Due to the malfunction of the NER module, the only start entity, *"Farrukhzad Khosrau V"*, was not successfully detected. Without the start entities, the reasoning chains cannot be established, and the further information flow in the entity graph is blocked at the first step.

The third case in Figure 5 is also a bad case, which includes a query of the *Comparison* query type. Due to the lack of numerical computation ability of our model, it fails to give a correct answer, although the query is just a simple comparison between two days *"February 20, 1959"* and *"February 10, 1967"*. It is an essential problem to incorporate numerical operations for further improving the performance in cases of the comparison query type.

5 Conclusion

We introduce Dynamically Fused Graph Network (DFGN) to address multi-hop reasoning. Specifically, we propose a dynamic fusion reasoning block based on graph neural networks. Different from previous approaches in QA, DFGN is capable of predicting the sub-graphs dynamically at each reasoning step, and the entity-level reasoning is fused with token-level contexts. We evaluate DFGN on HotpotQA and achieve leading results. Besides, our analysis shows DFGN can produce reliable and explainable reasoning chains. In the future, we may incorporate new advances in building entity graphs from texts, and solve more difficult reasoning problems, e.g. the cases of comparison query type in HotpotQA.

References

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2018. Neural models for reasoning over multiple mentions using coreference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 42–48.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2017. Stochastic answer networks for machine reading comprehension. *arXiv preprint arXiv:1712.03556*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Sewon Min, Victor Zhong, Richard Socher, and Caiming Xiong. 2018. Efficient and robust question answering from minimal context over documents. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1725–1735.

Tsendsuren Munkhdalai and Hong Yu. 2017. Reasoning with memory augmented neural networks for language comprehension. In *Proceedings of the International Conference on Learning Representations*.

Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2230–2235.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you dont know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 784–789.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. In *Proceedings of the International Conference on Learning Representations*.

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

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018b. A graph-to-sequence model for amr-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1616–1626.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 641–651.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the International Conference on Learning Representations*.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 189–198.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association of Computational Linguistics*, 6:287–302.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Mantong Zhou, Minlie Huang, and Xiaoyan Zhu. 2018. An interpretable reasoning network for multi-relation question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2010–2022.