A Sequential Flow Control Framework for Multi-hop Knowledge Base Question Answering

Minghui Xie, Chuzhan Hao, and Peng Zhang*
College of Intelligence and Computing, Tianjin University
{minghuixie, chuzhanhao, pzhang}@tju.edu.cn

Abstract

One of the key challenges of knowledge base question answering (KBQA) is the multi-hop reasoning. Since in different hops, one attends to different parts of question, it is important to dynamically represent the question semantics for each hop. Existing methods, however, (i) infer the dynamic question representation only through *coarse-grained* attention mechanisms, which may bring information loss, (ii) and have not effectively modeled the *sequential logic*, which is crucial for the multi-hop reasoning process in KBQA. To address these issues, we propose a sequential reasoning self-attention mechanism to capture the crucial reasoning information of *each single hop* in a more fine-grained way. Based on Gated Recurrent Unit (GRU) which is good at modeling sequential process, we propose a simple but effective GRU-inspired Flow Control (GFC) framework to model sequential logic in *the whole multi-hop process*. Extensive experiments on three popular benchmark datasets have demonstrated the superior effectiveness of our model. In particular, GFC achieves new state-of-the-art Hits@1 of 76.8% on WebQSP and is also effective when KB is incomplete. Our code and data are available at https://github.com/Xie-Minghui/GFC.

1 Introduction

Knowledge base question answering (KBQA) aims to answer questions from structured knowledge bases. In real application scenarios of KBQA, reasoning with multiple hops over knowledge graph (KG) is necessary for answering complex questions. Therefore, how to perform multi-hop reasoning effectively becomes a key challenge for multi-hop KBQA task (Sun et al., 2018; Zhang et al., 2018; Ho et al., 2020; Shi et al., 2020; Han et al., 2020).

Existing methods for multi-hop KBQA have three main strands. The first is semantic parsing based methods, which generate query graphs or statements by parsing questions (Yih et al., 2015; Luo et al., 2018; Lan and Jiang, 2020). The second is embedding-based methods which score the embeddings of question objectives and candidate answers (Dong et al., 2015; Miller et al., 2016; Hao et al., 2017; Saxena et al., 2020). The third is path-based methods, which start from topic entities of question and walk on KG to find answers. The third direction has its own advantages in terms of interpretability and extensibility (Sen et al., 2021). In recent years, more and more works have focused on path-based multi-hop reasoning methods (He et al., 2021; Sen et al., 2021; Shi et al., 2021).

However, existing methods still face some critical problems. First, path-based methods and some embedding-based methods usually leverage coarse-grained attention mechanisms to capture reasoning

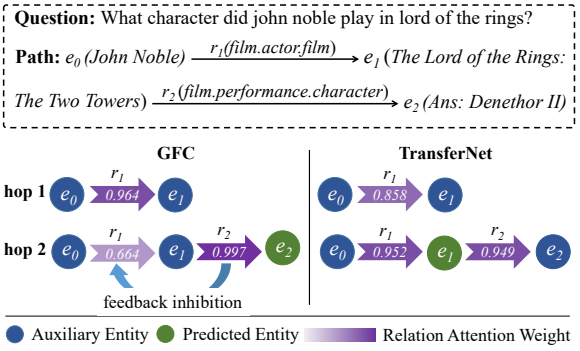

Figure 1: The above picture shows relations attention weights on the reasoning paths of GFC and the strong path-based method TransferNet. The final entity scores are the weighted sum of two hops which are positive correlation with relation attention weights. TransferNet tends to give r_1 high score in the 2nd hop, thus obtaining wrong answer (right). GFC can effectively weaken the attention of r_1 in the 2nd hop by introducing GRU-like sequential logic into the multi-hop process (left). People tend to pay more attention to current relations while pay less attention to past relations. Thus GFC is more consistent with human reasoning habit.

*Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 8450 - 8460
December 7-11, 2022

information of each hop. For example, KVMemNN (Xu et al., 2019) adopts cross-attention between key-value memory and sentence-level question representation. IRN (Zhou et al., 2018) uses the sentence-level question representation to eliminate relation embeddings of previous hop. Some methods (He et al., 2021; Shi et al., 2021) adopt cross-attention between the sentence-level question representation and question tokens. However, compressing all the necessary information into the sentence-level representation may lose some crucial information. Although these methods have achieved good performance, there is still room for improvement.

Second, they lack modeling sequential logic effectively in the whole multi-hop process. Humans often reason sequentially and consider past and present information comprehensively, which is a kind of sequential logic. However, the dynamic question representation of each hop is relatively independent (Cohen et al., 2020; Shi et al., 2021). And they do not control information flow effectively in different hops. For example, in Figure 1, models need to inhibit past relations for getting the right answer. However, existing methods cannot do this well.

In response, we propose a novel model for multi-hop KBQA, dubbed **GFC**. First, we design a sequential reasoning self-attention mechanism to obtain more fine-grained reasoning information of each hop. Our update mechanism combines the self-attention mechanism with sequential logic in the reasoning scenario. It can capture more nuanced reasoning information to distinguish similar relations on KG. Second, we design a simple but effective GRU-inspired flow control framework to model the sequential logic in the whole multi-hop process more effectively. This framework controls reasoning information flow among different hops, which enables GFC to consider reasoning information of past and present comprehensively. Besides, it tactfully integrates the proposed update mechanism into itself through our heuristic thinking about GRU. Inspired by the gating mechanism of GRU, we also introduce a self-gate unit to filter out redundant past reasoning information. As integral parts of framework, these mechanisms further enhance the capability of the overall flow control framework. Our key contributions are as follows:

- We design a sequential reasoning self-attention mechanism to extract the crucial reasoning information of single hop in a more fine-grained way.
- We propose a GRU-inspired flow control framework to model the sequential logic in the whole multi-hop process more effectively.
- Through controlling reasoning flow among hops and our novel update mechanism, GFC is superior to most existing methods. Specially, GFC achieves new state-of-the-art Hits@1 result of 76.8% on WebQSP and is also highly effective when KB is incomplete.

2 Related Work

In this paper, we mainly focus on neural network based methods.

2.1 Path-based Methods

These methods usually infer hop by hop over knowledge graph. Thus they can produce the reasoning chains to provide better interpretability.

Differentiable Knowledge Graph These methods use a sparse-matrix reified KB proposed by ReifKB (Cohen et al., 2020) to represent a symbolic knowledge base. The reasoning process is formulated as the multiplication of entity vector and relation matrix. E2EQA (Sen et al., 2021) handles multiple-entity questions by intersecting answers of different topic entities. TransferNet (Shi et al., 2021) proposes an effective and trasparent framework. These methods need no retraining for new entities and are easy to apply in large knowledge graph because they encode entities as one-hot embeddings. However, they lack modeling sequential logic in the whole multi-hop process effectively.

Reinforcement Learning These methods view the multi-hop reasoning process as a multi-step decision making process using reinforcement learning. MINERVA (Das et al., 2018) and SRN (Qiu et al., 2020) define states as tuple of question and entities, actions as traverse operation from the current entity on knowledge graph. NSM (He et al., 2021) uses teacher network to provide weak intermediate supervision signals of reasoning paths to student network. Although they have strong interpretability, they usually suffer from the convergency issue due to the huge search space and are harder to train compared to other approaches.

2.2 Embedding-based Methods

KVMemNN (Miller et al., 2016) reads key-value memory iteratively to conduct multi-hop reasoning.

EmbedKGQA (Saxena et al., 2020) utilizes KG embeddings to score question and condidate answers. GraftNet (Sun et al., 2018) and PullNet (Sun et al., 2019) retrieve a qustion-specific subgraph and then use graph convolutional network (Kipf and Welling, 2017) to implicitly infer answers. They enjoy high recall but suffer from much noisy entities. They have relatively weak interpretability because they usually cannot produce the reasoning chains.

3 Methodology

The diagram of our proposed model GFC is shown in Figure 2. The task of KBQA is to find the answer entities for natural language question q with the help of a relation graph $\mathcal{G}$. The entities mentioned in a question are called topic entities. Starting from topic entities, we derive the gold answer entities through the multi-hop reasoning on knowledge graph. Our proposed model GFC adopts a sparse-matrix reified KB proposed by RefiedKB (Cohen et al., 2020) to represent symbolic knowledge base. This representation method enables our model to perform rapid calculations on large scale knowledge graphs and need no retraining for new entities.

3.1 Sequential Reasoning Self-Attention Mechanism

To capture the crucial reasoning information in a more fine-grained way and alleviate the loss of crucial reasoning information of each hop, we combine the self-attention mechanism with the sequential logic in the multi-hop process. Specially, we view the initial question representation as query, and the question representation of previous hop as key and value. Given the initial question representation H^0 and the question representation H^{t-1} at hop $t-1$ ($t \in [1, T]$), we firstly transform H^{t-1} and then compute attention matrix S with H^0. After that, we do row-wise softmax on S to figure out which parts of H^{t-1} are more important in current hop. The processed matrix is noted as S_q. Then we apply the computed attention matrix S_q to H^{t-1} to obtain the crucial reasoning information $\tilde{U}^t$. The detailed computing process is as follows:

$$\boldsymbol{S} = \mathcal{F}^k(\boldsymbol{H}^{t-1}) \times \boldsymbol{H}^0 \quad (1)$$

$$\boldsymbol{S}_q = \text{row-wise softmax}(\boldsymbol{S}) \quad (2)$$

$$\tilde{\boldsymbol{U}}^t = \boldsymbol{H}^{t-1} \times \boldsymbol{S}_q \quad (3)$$

where $\mathcal{F}^k$ denotes a linear fully connected layer, $\{H^0, H^{t-1}, \tilde{U}^t\} \in \mathbb{R}^{L \times d}$ and $\{S, S_q\} \in \mathbb{R}^{L \times L}$. L is the length of question and d is the hidden size.

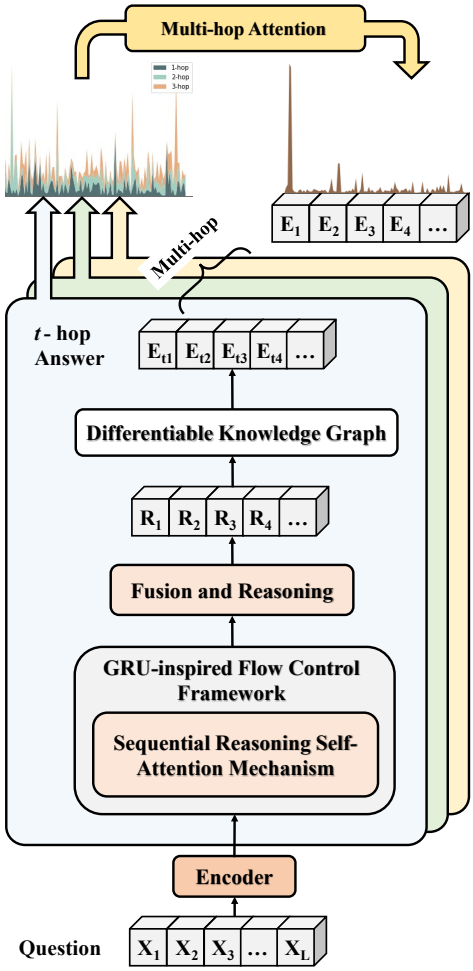

Figure 2: Overall architecture of our proposed GFC model.

At the first hop, H^{t-1} equals H^0. In this case, this process is a vanilla self-attention mechanism. As the reasoning process goes on, H^{t-1} is no longer equal to H^0, which means we use the question representation of previous hop and the initial question representation to capture the crucial reasoning information through the self-attention mechanism. Therefore, we call this process the sequential reasoning self-attention mechanism.

3.2 GRU-inspired Flow Control Framework

After capturing the crucial reasoning information of current hop, how to control the reasoning information flow is crucial in modeling the sequential logic in the whole multi-hop process effectively. Motivated by the Gated Recurrent Unit (GRU) (Cho et al., 2014), we propose a simple but effective reasoning information flow control framework. Here is how we get inspired. The main part of GRU is as follows:

$$\tilde{h}^t = tanh(\boldsymbol{W}_h x^t + \boldsymbol{U}_h(r^t \odot h^{t-1}) + b_h) \quad (4)$$

$$h^t = z^t \odot h^{t-1} + (1 - z^t) \odot \tilde{h}^t \quad (5)$$

where x^t, h^{t-1} and $\tilde{h}^t$ are the input, the previous hidden state and the new hidden state, respectively. r^t and z^t are the *reset* gate and *update* gate, respectively. $\boldsymbol{W}_h$, $\boldsymbol{U}_h$ and b_h are trainable parameters.

Analogy to the above formulas, we view the crucial reasoning information $\tilde{U}^t$ heuristically as the new hidden state $\tilde{h}^t$ because $\tilde{U}^t$ is also updated information like $\tilde{h}^t$. This means our sequential reasoning self-attention mechanism plays the same role as Eq. 4. Similar to Eq. 5, we synthesize the past and present reasoning information by introducing the gate mechanism. As pointed out in Cho et al. (2014) , the *update* gate z^t selects whether the hidden state is to be updated with a new hidden state $\tilde{h}^t$ while the *reset* gate r^t decides whether the previous hidden state h^{t-1} is ignored. In our sequential reasoning self-attention mechanism, $\tilde{U}^t$ is the crucial reasoning information of current hop. Therefore, we do not need an *update* gate z^t but a *reset* gate r^t to decide how much the past reasoning information is retained. Therefore, we deduce the following equation:

$$\boldsymbol{U}^t = r^t \odot \boldsymbol{U}^{t-1} + \tilde{\boldsymbol{U}}^t \quad (6)$$

To achieve the effect of the *reset* gate r^t, we introduce the self-gate unit (SGU). Figure 3 illustrates the architecture of SGU.

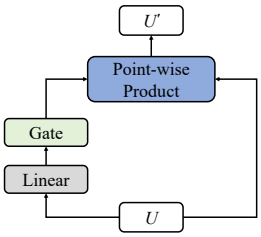

Figure 3: Self-Gate Unit (SGU).

The SGU will get the internal attention distribution for eliminating the irrelevant information of previous reasoning information U^{t-1}. The detailed process is as follows:

$$\boldsymbol{SGU}(\boldsymbol{U}^{t-1}) = \boldsymbol{T}(\boldsymbol{U^{t-1}}) \odot \boldsymbol{U}^{t-1} \quad (7)$$

$$\boldsymbol{T}(\boldsymbol{U^{t-1}}) = \sigma(\boldsymbol{U}^{t-1}\boldsymbol{W}_1 + \boldsymbol{b}_1) \quad (8)$$

where $T(\cdot)$ indicates the transform gate, $\sigma(\cdot)$ is the element-wise sigmoid function that confines the point-wise weights into a fixed range. $\odot$ denotes the Hadamard product. $W_1 \in \mathbb{R}^{d\times d}$ and $b_1 \in \mathbb{R}^d$ are trainable parameters. Thus the final reasoning information U^t of current hop is calculated as follows:

$$\boldsymbol{U}^t = \boldsymbol{SGU}(\boldsymbol{U}^{t-1}) + \tilde{\boldsymbol{U}}^t \quad (9)$$

To alleviate large semantic deviation in the whole multi-hop process, we add the reasoning information U^t to the initial question semantics. Finally, the dynamic question representation of each hop in our model is computed as follows:

$$\boldsymbol{H}^t = \boldsymbol{H}^0 + \boldsymbol{SGU}(\boldsymbol{U}^{t-1}) + \tilde{\boldsymbol{U}}^t \quad (10)$$

As shown in Figure 4, our proposed framework is similar to the architecture of GRU (Cho et al., 2014) and Bert (Devlin et al., 2019), which can be viewed as a fusion of two powerful NLP models approximatively. This inspires us to model the multi-hop reasoning process in the same way that we model language sequences.

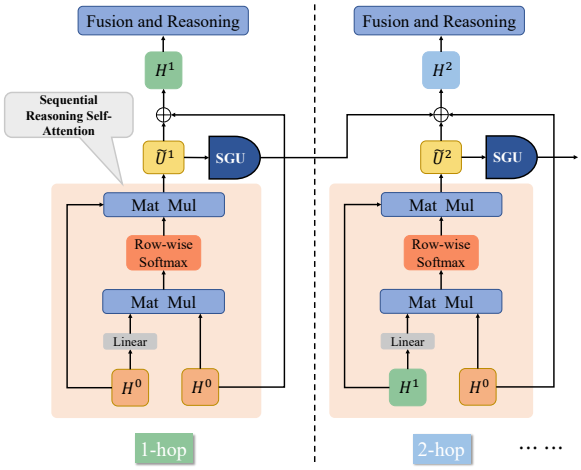

Figure 4: The schematic diagram of the GRU-inspired Flow Control Framework.

3.3 Fusion and Reasoning Module

After getting the fine-grained dynamic question representation, we use it to determine which relations we should walk on knowledge graph in current hop. In detail, we sum the cross attention matrix S_q in column direction and then do softmax to obtain the weight of each token. Then we fuse the dynamic question representation H^t using these weights to get the question vector $q^t \in \mathbb{R}^d$ for relation prediction. The computing process is as follows:

$$\boldsymbol{q}^t = \boldsymbol{H}^t \times \mathrm{softmax}(\sum \mathrm{S_q}) \quad (11)$$

Then we use q^t to make a multi-label classification on the relations of knowledge graph, which makes our model can lookup multiple paths in parallel on knowledge graph:

$$\boldsymbol{r}^t = \text{sigmoid}(\mathcal{F}^r(\boldsymbol{q}^t)) \tag{12}$$

where $\mathcal{F}^r$ is a linear fully connected layer. The $follow$ operation will calculate the scores of all entities in the t hop. The resulting entity vector e_t of each hop is computed as:

$$\mathbf{e}^t = \text{follow}(\mathbf{e}^{t-1}, \mathbf{r}^t) \tag{13}$$

where $e^t \in [0,1]^n$ is the scores of all entities in the t hop. e^0 is the initial score where only the topic entities get 1.

3.4 Output Layer

At the end of all T hops, we will calculate the multi-hop attention distribution $a \in \mathbb{R}^T$ to determine which hop answers are located in. We argue that the question semantics of t hop is wrong if the right answers can be obtained within $t-1$ hop. Thus we collect the dynamic question representations of all hops to calculate the multi-hop attention score:

$$\boldsymbol{a} = \text{softmax}(\mathcal{F}^h([q^1; \cdots; q^T])) \tag{14}$$

where $\mathcal{F}^h$ denotes a linear fully connected layer.

The final predicted answers $\hat{y}$ are computed as:

$$\hat{\mathbf{y}} = \sum_{t=1}^{T} a^t e^t \tag{15}$$

Given the golden answer set y, we take the $L2$ Euclidean distance between $\hat{y}$ and y as our training objective:

$$\mathcal{L} = ||\hat{y} - y|| \tag{16}$$

Since the framework is totally differentiable, we can learn all the intermediate probability values via this simple goal.

4 Experiments

4.1 Datasets

MetaQA (Zhang et al., 2018) is a large-scale dataset of multi-hop KBQA with more than 400k questions, which are generated using dozens of templates and have up to 3 hops. Its knowledge graph is from the movie domain, including 43k entities, 9 predicates and 135k triples. Each sample has a corresponding hop label.

WebQSP (Yih et al., 2016) is a subset of WebQuestions and completes the corresponding query statement. It contains 4,737 questions (2,998 train, 1,639 test) based on Freebase (Bollacker et al., 2008) which has millions of entities and triples. These questions can be solved under the reasoning chain of 1 hop or 2 hops. Following (Saxena et al., 2020), we pruned the KB to contain only mentioned relations and within 2-hop triples of mentioned entities. In order to improve the reasoning ability, we add reversed predicates. Finally, the KB includes 1.8 million entities, 1144 predicates and 11.4 million triples.

CompWebQ (Talmor and Berant, 2018) is a further enhanced version of WebQSP with 34,689 questions (27,649 train, 3,509 dev, 3,531 test). It contains more complex multi-hop questions, mainly including type constraints, display or implicit time constraints and aggregation operations.

4.2 Baselines

- **KVMemNN** (Miller et al., 2016) uses the key-value memory to store triplet knowledge and conducts multi-hop reasoning by reading the memory iteratively.
- **GraftNet** (Sun et al., 2018) uses Personalized PageRank method to extract a question-specific subgraph and then infers answers using graph nerual network.
- **PullNet** (Sun et al., 2019) uses an iterative process to construct a question-specific subgraph and infers with heterogeneous information to find the best answers.
- **ReifKB** (Cohen et al., 2020) proposes a sparse-matrix reified KB to represent a symbolic knowledge base, which can be trained in an end-to-end way.
- **EmbedKGQA** (Saxena et al., 2020) utilizes the link predict ability of KG embeddings (Bordes et al., 2013; Trouillon et al., 2016) to handle multi-hop reasoning questions, especially on incomplete knowledge graph.
- **EMQL** (Sun et al., 2020) proposes set operators to construct a more faithful query method for deductive reasoning.
- **NSM** (He et al., 2021) proposes teacher network to provide weak supervision signals of

Model		MetaQA			WebQSP		CompWebQ
		1-hop	2-hop	3-hop	Hits@1	F1	Hits@1
Embed-based	KVMemNN (Miller et al., 2016)	96.2	82.7	48.9	46.7	38.6	21.1
	GraftNet (Sun et al., 2018)	97.0	94.8	77.7	67.8	62.4	32.8
	PullNet (Sun et al., 2019)	97.0	99.9	91.4	68.1	–	47.2
	EmbedKGQA (Saxena et al., 2020)	97.5	98.8	94.8	66.6	–	–
	EMQL (Sun et al., 2020)	–	98.6	99.1	75.5	–	–
Path-based	ReifiedKB (Cohen et al., 2020)	96.2	81.1	72.3	52.7	–	–
	NSM (He et al., 2021)	–	–	–	74.3	67.4	–
	TransferNet (Shi et al., 2021)	97.5	**100.0**	**100.0**	71.4	–	48.6
	GFC (ours)	**97.7**	**100.0**	**100.0**	**76.8**	**69.2**	**50.4**

Table 1: Experimental results of Hits@1 on MetaQA, WebQSP and CompWebQ and F1 on WebQSP.

reasoning paths for the student network.

- **LSRL** (Yan et al., 2021) proposes three relation learning tasks for BERT-based KBQA, including relation extraction, relation matching, and relation reasoning.
- **TransferNet** (Shi et al., 2021) proposes an effective and transparent framework, which supports both label and text relations.

4.3 Experimental Settings

In order to intuitively reflect the ability of our model in multi-hop questions, we label each question on WebQSP with the number of hops according to the reasoning chains in the original data. Only about 20 questions have no reasoning chains. So we manually label the missing reasoning chains. The label information is only used when evaluating.

For the experiments of WebQSP and CompWebQ, we use the uncased base version of pretrained BERT (Devlin et al., 2019) as the question encoder. We download the bert-base-uncased model from HuggingFace [1]. We set the hop sizes $T = 2$ for WebQSP and CompWebQ dataset. For the experiments of MetaQA, we use bi-directional GRU (Chung et al., 2014) as the question encoder and set the hop size $T = 3$.

Our model is trained using RAdam (Liu et al., 2020) optimizer with a learning rate of $1e^{-3}$. We use a scheduler that decreases linearly after increasing from 0 to $1e^{-3}$ during a linear warmup period. For BERT, we use a smaller learning rate $3e^{-5}$. The mini-batch size on WebQSP is set to 16, on CompWebQ is 64 and on MetaQA is 128. Besides Hits@1, we also use the average question-wise F_1 score as our evaluation metrics. We trained our model on a single GPU of Tesla P40, which took about 16 hours for WebQSP, 40 hours for CompWebQ and 6 hours for MetaQA.

[1] https://huggingface.co/bert-base-uncased

4.4 Main Results

Table 1 compares different models on three benchmarks. As we can see, GFC performs pretty much the same way as state-of-the-art model TransferNet of MetaQA. GFC performs perfectly in the 2-hop and 3-hop questions on MetaQA. As for the 1-hop questions of MetaQA, GFC achives 97.7% which surpasses TransferNet and EmbedKGQA. The reason why the performance on 1-hop is worse than 2-hop and 3-hop is that more relation constraints can alleviate the noise of the dataset itself.

WebQSP is more complex than MetaQA, because it has much more relations and triplets but much less training samples. Specially, GFC gets 76.8% on WebQSP dataset for Hits@1, which achieves new state-of-the-art performance. Our path-based method beats the most effective embedding-based method EMQL (75.5%). In other words, our path-based method not only has better interpretability and extensibility, but also has better performance. GFC also achieves very competitive result 69.2% for F1. On CompWebQ dataset, we compare the results with Shi et al. (2021) and Sun et al. (2019) on the dev set. GFC achieves 50.4% for Hits@1, which performs better than TransferNet (48.6%) and PullNet (47.2%).

4.5 Ability to model sequential logic

To verify GFC can model the sequential logic in the whole multi-hop process effectively, we compare Hits@1 of 1-hop and 2-hop questions respectively between GFC and the strong path-based model TransferNet (Shi et al., 2021) based on the hop labels of WebQSP.

Model	WebQSP	
	1-hop	2-hop
TransferNet	79.4	58.7
GFC (ours)	**81.6**	**68.4**

Table 2: Hits@1 comparison of 1-hop and 2-hop questions on WebQSP between GFC and TransferNet.

Table 2 shows that Hits@1 of 1-hop and 2-hop questions increase by 2.2% and 9.7% respectively. The fact that GFC performs much better in 2-hop questions proves the effectiveness of our proposed GRU-inspired flow control framework. The whole framework can consider what has already been focused on and alleviate some illogical reasoning (Please refer Figure 1).

4.6 Reasoning ability over incomplete KG

In real application scenarios, knowledge graph (KG) is usually incomplete, which requires models to have stronger reasoning ability and robustness. In general, there are several similar paths from the topic entities to the answers entities. But some paths are incorrect even if they can lead to the right answers. As shown in Figure 6, there are two paths from the topic entity *George VI* to the answer *Queen Elizabeth*. But the path above is wrong, because the relations *people.person.children* and *people.person.parents* are not correct for the specific question *What is the name of king george vi wife*. Some of these paths will disappear when KB is incomplete. In this case, we must follow the right paths to get the right answers, which requires stronger reasoning ability and robustness of models. For embedding-based methods, they will get worse embeddings of entities and relations because the number of triplets for training KG embeddings becomes much less. we compare GFC with other competitive methods on the incomplete WebQSP with half KG preprocessed by EmbedKGQA (Saxena et al., 2020). The results in Table 3 shows that GFC achieves 59.5% for Hits@1 and performs much better than EmbedKGQA (53.2%), which aims to handle the multi-hop KBQA on incomplete KG specially. GFC also surpasses the strong path-based method TransferNet by a large margin, which proves our method has stronger reasoning ability. In particularly, GFC surpasses previous state-of-the-art LSRL (58.8%), while keeping simple without additional pre-trained tasks.

Model	WebQSP	WebQSP KG-50
KVMemNN	46.7	32.7
GRAFT-Net	67.8	48.2
PullNet	68.1	50.3
EmbedKGQA	66.6	53.2
TransferNet	71.4	52.4
LSRL	72.9	58.8
GFC (ours)	**76.8**	**59.5**

Table 3: The performance comparison of Hits@1 with the full KG and the 50% KG on WebQSP.

4.7 Impact of hop size

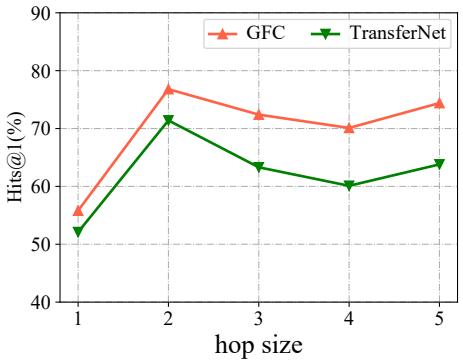

Figure 5: Results when setting different hop sizes

Hop size is a crucial hyperparameter. To investigate its impact, we compare the performance of GFC and TransferNet when choosing different hop sizes. As shown in Figure 5, the performance of both models decreases to varying degrees when the hop size increases. Most questions on WebQSP need no more than 2-hop reasoning. Excessive hop sizes will introduce additional noise. But compared to TransferNet, GFC has a more stable performance among different hops. As hop size increases, the gap between two models gradually widens.

4.8 Ablation Study

We remove or replace model components and report the performance on WebQSP and CompWebQ datasets in Table 4. In (a), we remove the SGU. In (b), we replace H^0 with H^{t-1} of Eq. 10 to evaluate the importance of the initial question semantics. In (c), we remove past reasoning information, which can be viewed as a part ablation experiment of GRU-inspired information flow control framework. But sequential reasoning self-attention mechanism and some tightly connected modules still remains.

As is shown in Table 4, taking WebQSP as an example, the past reasoning information is the most

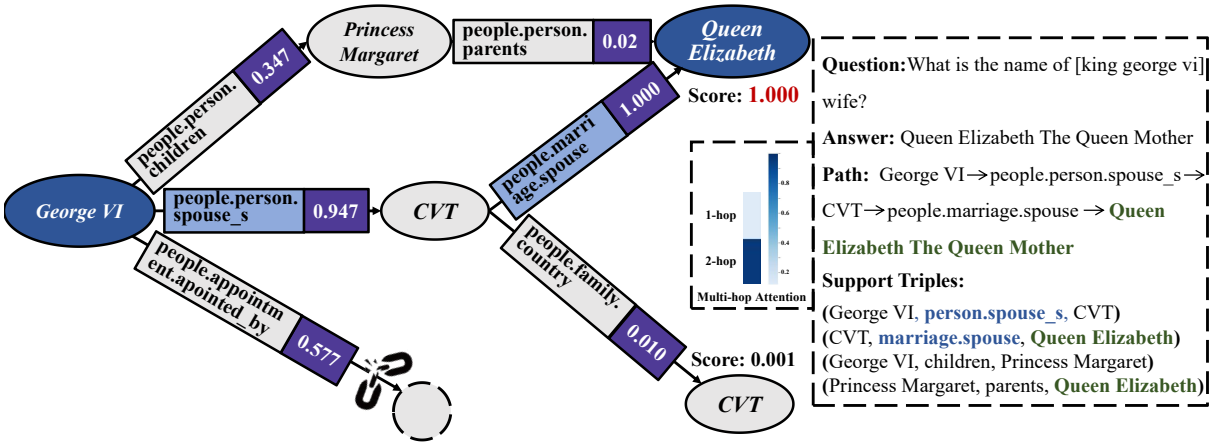

Figure 6: The crucial part of reasoning process of one example from WebQSP. We start from the topic entity *George VI*. In the 1st hop, GFC gives relation *people.person.spouse_s* the highest score 0.947. There is no path from *George VI* with relation *people.appointment.apointed_by*. Thus, this path will be broken. This is one of advantages of our method which can use rich knowledge graph topology information to filter out irrelevant relations and entities. The final score of *Queen Elizabeth* is the sum of two paths. The final answers are selected by the multi-hop attention. We restrict the score in [0,1] for training model easily.

Model	WebQSP	CWQ
GFC-full(ours)	**76.8**	**50.4**
(a) w/o SGU	75.3	49.6
(b) w/o initial semantics	76.1	49.9
(c) w/o past information	75.1	49.3

Table 4: Ablation study on WebQSP and CompWebQ (CWQ).

critical to the performance (1.7% drop), which proves past reasoning information can help current decision. In (b), the performance reduces about 0.7%, which indicates update upon the initial question representation can alleviate the large semantic deviation indeed. In (a), the SGU accounts for 1.5% performance drop respectively, which proves the effectiveness of the SGU in refining reasoning information and alleviate introducing the noise of past reasoning information.

4.9 Error Analysis

Figure 6 shows the reasoning process of one correct example of our model GFC. In addition, we explore frequently observed error cases where the proposed model fails to produce correct answers. The first type of error is that questions are tokenized incorrectly by BERT tokenizer, such as *what highs ##cho ##ol did harper lee go to* and *when's the last time the steelers won the superb ##ow ##l*. BERT tokenizes the crucial topic entity incorrectly, which causes our model unable to recognize the correct relations in current hop. A simple and easy way to think of is to add these wrong tokenized entities into the vocabulary. But in this way, the pretrained word embeddings of BERT cannot be used. We try to learn these words from scratch, but get worse results because there is no enough training samples for these entities. The second type error is because of relation confusion. Many relations have very similar meanings, such as *tv.tv_guest_role.actor* and *tv.regular_tv_appearance.actor*. GFC cannot distinguish them clearly, because the number of samples related with them is so small.

5 Conclusions

In this paper, we design (i) a *sequential reasoning self-attention mechanism* to extract the crucial reasoning information of *each single hop* in a more fine-grained way and (ii) a *GRU-inspired flow control framework* to model sequential logic in *the whole multi-hop process* more effectively. Experimental results show the superior performance of GFC. Specially, GFC achieves new state-of-the-art Hits@1 performance on WebQSP. GFC also shows its high effectiveness when KB is incomplete. As a path-based method, GFC not only has better interpretability and extensibility, but also has better performance. In future work, we plan to investigate further on how to model the multi-hop reasoning process using the structures of language models.

Limitations

Although our method achieves surprising performance in the multi-hop KBQA task, there are still some limitations to be improved. The limitation of our study are summarized as follows:

1) The optimal hop size in our model depends on experimental results. On one hand, the performance of GFC are not stable enough when the hop size increases (shown in Figure 5). On the other hand, the hop size required to reason is different for complex questions in real application scenarios. Reasoning with the same hop size for all questions will greatly increase the computational cost and introduce unnecessary noise. Thus how to determine the optimal hop size for each question adaptively still remains a key challenge for multi-hop KBQA task.

2) As discussed in error analysis, some relations have very similar meanings but with few training samples. Our model does not work well with these relations.

3) Our model can only receive feedback from final answers. How to provide more supervision signals from the perspective of model design will be an interesting exploration direction.

Ethics Statement

We worked within the purview of acceptable privacy practices and strictly followed the data usage policy. In all the experiments, we use public datasets accordding to their intended usage. We have also described our experimental settings in detail to ensure the reproducibility of our work. We neither introduce any social/ethical bias to the model nor amplify any bias in the data, so we do not foresee any direct social consequences or ethical issues.

Acknowledgments

This work is supported in part by Natural Science Foundation of China (grant No.62276188 and No.61876129), the Beijing Academy of Artificial Intelligence(BAAI), TJU-Wenge joint laboratory funding, and MindSpore [2].

[2] https://www.mindspore.cn/

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

William W. Cohen, Haitian Sun, R. Alex Hofer, and Matthew Siegler. 2020. Scalable neural methods for reasoning with a symbolic knowledge base. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2018. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over Freebase with multi-column convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint*

Conference on Natural Language Processing (Volume 1: Long Papers), pages 260–269, Beijing, China. Association for Computational Linguistics.

Jiale Han, Bo Cheng, and Xu Wang. 2020. Two-phase hypergraph based reasoning with dynamic relations for multi-hop KBQA. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3615–3621. ijcai.org.

Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 221–231, Vancouver, Canada. Association for Computational Linguistics.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 553–561.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Yunshi Lan and Jing Jiang. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 969–974, Online. Association for Computational Linguistics.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the variance of the adaptive learning rate and beyond. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Kangqi Luo, Fengli Lin, Xusheng Luo, and Kenny Zhu. 2018. Knowledge base question answering via encoding of complex query graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2185–2194, Brussels, Belgium. Association for Computational Linguistics.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, Austin, Texas. Association for Computational Linguistics.

Yunqi Qiu, Yuanzhuo Wang, Xiaolong Jin, and Kun Zhang. 2020. Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 474–482. ACM.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, Online. Association for Computational Linguistics.

Priyanka Sen, Armin Oliya, and Amir Saffari. 2021. Expanding end-to-end question answering on differentiable knowledge graphs with intersection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8805–8812, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. 2021. TransferNet: An effective and transparent framework for multi-hop question answering over relation graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4149–4158, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiaxin Shi, Shulin Cao, Liangming Pan, Yutong Xiang, Lei Hou, Juanzi Li, Hanwang Zhang, and Bin He. 2020. Kqa pro: A large-scale dataset with interpretable programs and accurate sparqls for complex question answering over knowledge base. *ArXiv preprint*, abs/2007.03875.

Haitian Sun, Andrew Arnold, Tania Bedrax Weiss, Fernando Pereira, and William W Cohen. 2020. Faithful embeddings for knowledge base queries. In *Advances in Neural Information Processing Systems*, volume 33, pages 22505–22516. Curran Associates, Inc.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2380–2390, Hong Kong, China. Association for Computational Linguistics.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen.

2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242, Brussels, Belgium. Association for Computational Linguistics.

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)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org.

Kun Xu, Yuxuan Lai, Yansong Feng, and Zhiguo Wang. 2019. Enhancing key-value memory neural networks for knowledge based question answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2937–2947, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuanmeng Yan, Rumei Li, Sirui Wang, Hongzhi Zhang, Zan Daoguang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Large-scale relation learning for question answering over knowledge bases with pre-trained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3653–3660.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China. Association for Computational Linguistics.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 6069–6076. AAAI Press.

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, Santa Fe, New Mexico, USA. Association for Computational Linguistics.