



Multi-hop Reading Comprehension Incorporating Sentence-Based Reasoning

Lijun Huo¹, Bin Ge¹, and Xiang Zhao^{1,2}(✉)

¹ Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, China

xiangzhao@nudt.edu.cn

² Collaborative Innovation Center of Geospatial Technology, Wuhan, China

Abstract. Multi-hop machine reading comprehension (MRC) requires models to mine and utilize relevant information from multiple documents to predict the answer to a semantically related question. Existing work resorts to either document-level or entity-level inference among relevant information, which can be too coarse or too subtle, resulting less accurate understanding of the texts. To mitigate the issue, this research proposes a sentence-based multi-hop reasoning approach named SMR. SMR starts with sentences of documents, and unites the question to establish several reasoning chains based on sentence-level representations. In addition, to resolve the complication of pronouns on sentence semantics, we concatenate two sentences, if necessary, to assist in constructing reasoning chains. The model then synthesizes the information existed in all the reasoning chains, and predicts a probability distribution for selecting the correct answer. In experiments, we evaluate SMR on two popular multi-hop MRC benchmark datasets - WikiHop and MedHop. The model achieves 68.3 and 62.9 in terms of accuracy, respectively, exhibiting a remarkable improvement over state-of-the-art option. Additionally, qualitative analysis also demonstrates the validity and interpretability of SMR.

Keywords: Reading comprehension · Multi-hop question answering · Sentence representation · Text understanding

1 Introduction

Machine reading comprehension (MRC) is an important and desired aspect in natural language understanding. Its purpose is to use machines to extract desired information and knowledge automatically, based on a given question and some documents. Compared to the basic tasks in natural language processing, such as named entity recognition and relation extraction, MRC is a more complicated and higher-level task, which requires deeper understanding of semantics.

In recent years, to verify the effect of MRC models, many data sets have been developed, represented by SQuAD [10]. Most of the existing datasets are aimed at single-hop MRC task such that each question corresponds to a document, and the information

for solving the question is restricted to that document. In other words, there is not a reasoning process among several documents, which nevertheless does not reflect real-life scenarios.

To better evaluate MRC models in a more realistic setting, the task of multi-hop MRC is delivered, where to answer a given question, multiple supporting documents are necessary. In other words, the multi-hop MRC task requires models to make reasoning hops among documents based on the information of the question, in order to find enough useful knowledge for predicting the answer. We focus on multi-hop MRC in this paper.

The multi-hop MRC task is notoriously challenging from at least the following three aspects. First, for each question, there are many supporting documents, but only a small portion of them contain information to resolve the question, and the rest are interference. Most existing MRC models find it difficult to handle documents of large scale, and have little anti-interference capability. Second, the information to resolve the question is distributed among multiple documents, which requires effective reasoning to form a reliable chain of information clue. However, current models are weak at performing effective reasoning over multiple documents. Third, there may be multiple possible chains of information clue formed by reasoning, which need to be screened and evaluated by quadratic sorting. The quality of this operation brings great uncertainty to MRC models in unveil the correct reasoning chain.

In view of these difficulties, in contrast to existing work resorting to either document-level or entity-level reasoning, which can be too coarse or too subtle, we propose SMR, a progressive model based on sentence-level reasoning. It is naturally inspired by the reading comprehension strategy of human. When human deal reading comprehension, one usually finds the keywords from the question firstly, and then searches for a sentence semantically related to the keywords in the supporting documents. Next, based on the knowledge of the current sentence, she reasons for the subsequent logical sentence to locate it, which is considered to be a hop. Finally, all the sentences extracted from the supporting documents make up a reasoning chain of information clue, and the answer can be finally derived.

To imitate the aforementioned process, SMR finds a sentence existing in the supporting documents according to the main entity in the question, to start the reasoning. Then, it employs a Sentence Selector, which iteratively selects a relevant sentence as an intermediate reasoning node, resulting in a complete chain. In this way, SMR will construct multiple reasoning chains, and in the end, it leverages an Answer Predictor to infer the answer, which integrates the information of the reasoning chains, as well as the question to derive a probability distribution of answers.

Further, sentences in human language often contain pronouns, and accurate resolution of pronouns and the nouns they refer to are essential for guiding reasoning, e.g., to link the pronoun ‘it’ in *sent*₂ of Fig. 1 with the noun ‘the Johannesburg Zoo’ in *sent*₁. Although existing co-reference resolution methods may help, it is practically non-trivial to conduct it without mistakes, in which case mistakes will be propagated to MRC. To alleviate the issue, we propose to concatenate two sentences (e.g., *sent*₁ and *sent*₂) into one concatenation sentence (e.g., *sent*₃). Hence, when the model needs to reason from *sent*₁ to *sent*₂, it will choose *sent*₃ as a node instead to avoid extra hopping, which substantially reduces the difficulty in overly long reasoning.



Fig. 1. Representation enrichment via concatenation of adjacent sentences.

Contributions. In summary, the proposed model SMR consists of three modules: Sentence Represent, Sentence Selector and Answer Predictor. And we make the following contributions in this paper:

- We proposed to leverage sentence-based reasoning for MRC, which constructs multiple chains that connect sentences relevant to the question;
- We introduce sentence concatenation to handle the potential issue of co-reference in context for effective sentence-based reasoning;
- We achieve competitive accuracy results on popular multi-hop datasets, and SMR is demonstrated to be able to explain the reasoning process.

Organization. We discuss related work in Sect. 2. Section 3 introduces the model SMR in details, including sentence representation, sentence selector and answer predictor. Then, we report the experimental study with in-depth analysis in Sect. 4, and conclude the paper in Sect. 5.

Note that existing multi-document MRC datasets have different formats, corresponding to various types of multi-document MRC. This research mainly focuses on the popular multi-hop datasets WikiHop and MedHop [8], where one needs to choose the correct answer from the given candidate set to the given question, based on a collection of documents.

2 Related Work

In recent years, various multi-hop MRC datasets have been developed, and these datasets all demand models to understand the semantics of texts and find the internal relationship between texts. However, their questions have different forms. For example, HotpotQA [9] and TriviaQA [17] contain {question, document set, answer}, where the answer must be generated, and the question is a natural language text. On the other hand, QAngaroo WikiHop and MedHop [8] contain {question, document set, answer, candidates}, where the answer is an entity presented in the given candidate set, and the question consists of an entity and a relationship. Some others such as Who Did What [18] and Children’s Book Test [19] provide cloze-style MRC datasets, on which the models need to predict the missing word/entity in questions.

According to the characteristics of these data sets, researchers have developed various models to handle the tasks. For example, [8] fuses multiple documents into a long one, and then uses the single-hop MRC model with bidirectional attention mechanism to deduce the answer. However, because the documents after fusion are too long and the

model has no information skipping capability, the performance of the model is far less accurate than that in the single-hop task.

With the assistance of knowledge guidance, [2] enables the model to integrate the semantics of documents, but the approach is difficult to apply due to the fact that external knowledge tends to be limited to a specific field.

Focus on reasoning, [4] gathers all possible reasoning paths according to the entities contained in the documents, and then scores each path to select the correct reasoning path. However, the method extracts many invalid paths that are apt to bring in interference and waste computing resources.

[5, 6] uses graph neural networks [20] to obtain the relationship between entities, and adds self-attention mechanism [7] into the model, which obtains a gain in the result. However, the model has poor interpretability owing to lack explicit reasoning, and meanwhile it is of high complexity and low efficiency.

The research in this paper was inspired by the research by EPAR [3], which creates a document explorer to select documents to build an inference tree. We follow the same framework to establish SMR, but substantially differ by incorporating sentence-based reasoning, explicit paths and sentence concatenation (to be introduced in Sect. 3). The innovative design implements a MRC model with higher accuracy and better interpretability (to be detailed in Sect. 4) (Fig. 2).

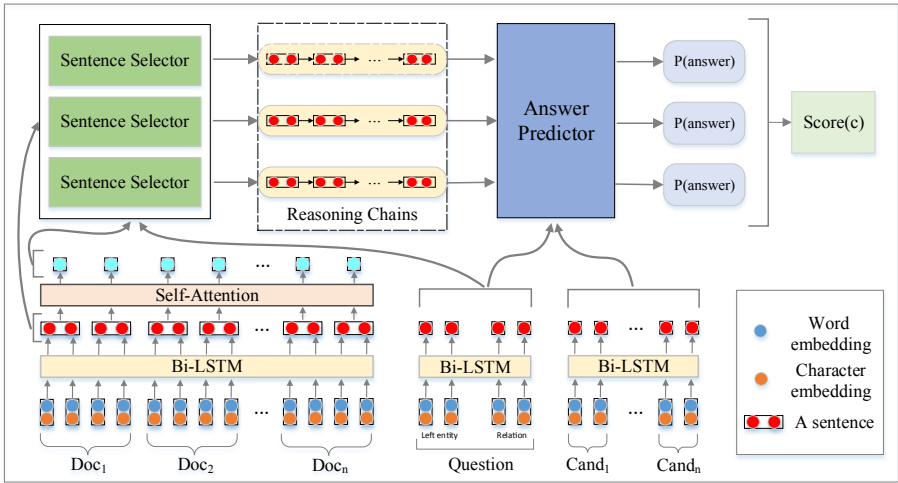


Fig. 2. Framework of sentence-based MRC.

3 Model

In the section, we introduce our proposed model for multi-hop MRC, which comprises three modules.

Before delving into the details, we first formally define the task that is investigated in this paper.

Task Definition. In the task of multi-hop MRC [8], there is a question q and a set of supporting documents T' . In particular, the question q is provided in the form of a tuple $(l_e, r, ?)$, where l_e is the left entity, and r represents the relation between l_e and the unknown right entity, which is the answer. In addition, there is also a candidate set $C' = \{c'_\eta\}_{\eta=1}^H$ containing the correct answer. The purpose is to predict the unknown right entity from C' .

In the sequel, we explain our proposed model, which first performs sentence segmentation and semantic encoding (Sect. 3.1), then inferences to build the multi-hop chains based on the encoded semantics (Sect. 3.2), and finally mines the evidence of the multi-hop chains to rank the candidates for finding the answer (Sect. 3.3).

3.1 Sentence Representation

We first conduct text preprocessing and word encoding methods. Then, we divide the supporting documents into single sentences and concatenation sentences. Subsequently, we explain the encoding methods of these steps.

Word Encoding. The goal of word encoding is to characterize the question and supporting documents as vectors for inputting into neural networks.

We first filter documents to reduce the number of interfering documents and the GPU memory occupied by the model. In practice, we use the TF-IDF algorithm to calculate and rank the cosine similarity between the question and each supporting document.

Then, we intercept the top- N supporting documents with the least similarity as the new supporting document set $T = \{t_n\}_{n=1}^N$. We apply the same word embedding and semantic encoding for l_e, r and T .

For word embedding, we combine character embedding and pre-trained Glove word embedding [12] as the initial word embedding and input them into a Highway Network [21] to obtain the final word representation. We use \mathbf{L}' , \mathbf{R}' and \mathbf{X}' to denote the word embedding of l_e, r and T respectively.

For semantic encoding, we pass \mathbf{L}' , \mathbf{R}' , and \mathbf{X}' through a bidirectional LSTM network [22] with v hidden units and concatenate the bidirectional output of LSTM as the word-level semantic representation. We use $\mathbf{L} \in \mathbb{R}^{Q_l \times v}$, $\mathbf{R} \in \mathbb{R}^{Q_r \times v}$, $\mathbf{X} \in \mathbb{R}^{N \times J \times v}$ as the word encoding of l_e, r and T , respectively, where Q_l, Q_r, J are the word-level lengths of l_e, r and T respectively.

Since each candidate c'_η can be found in the supporting document set T , we take out the word encoding corresponding to c'_η in \mathbf{X} , average it at the word-level and then get $\mathbf{c}_\eta \in \mathbb{R}^v$ as the semantic encoding of c'_η .

Sentence Encoding. The Sentence Encoding mainly divides each document into several sentences and converts each sentence to a vector.

We first cut a documents t into multiple sentences to obtain the single sentence set $\mathbf{D}^o = \{\mathbf{d}_i^o\}_{i=1}^I$ s.t. $\mathbf{d}_i^o \in \mathbb{R}^{K \times v}$ where I is the number of single sentences contained in t , K is the number of words that make up a single sentence and \mathbf{d}_{ik}^o is the corresponding

word encoding in X . We then connect all two adjacent single sentences in the document to obtain the concatenation sentence set $\mathbf{D}^b = \{\mathbf{d}_i^b\}_{i=1}^{I-1}$, \mathbf{d}_i^b can be given as

$$\mathbf{d}_i^b = \mathbf{d}_i^o \parallel \mathbf{d}_{i+1}^o \text{ and } 1 \leq i < I, \quad (1)$$

where \parallel is used to indicate concatenation. Next, we joint \mathbf{D}^o and \mathbf{D}^b to complete the sentence division of t and get the sentence set \mathbf{D} ; that is,

$$\mathbf{D} = \mathbf{D}^o \cup \mathbf{D}^b, \quad (2)$$

where \cup refers to union.

We adopt the same operation for all supporting documents and get the word-level sentence encoding \mathbf{S} of T ; that is,

$$\mathbf{S} = \mathbf{D}_1 \cup \mathbf{D}_2 \cup \dots \cup \mathbf{D}_N = \{s_1, \dots, s_{I'}\}, \quad (3)$$

where I' is the number of total sentences including single sentence and concatenation sentence of T . We apply a self-attention mechanism [7] to implement vector representation of sentences and get the sentence-level sentence encoding set \mathbf{E} of T . Specifically, the formula we use to transform a sentence s_i into a vector representation $\mathbf{e}_i \in \mathbb{R}^v$ is as follows (K is considered as the length of all sentences for simplicity); that is,

$$\begin{aligned} a_{ik} &= \tanh(\mathbf{W}_2 \tanh(\mathbf{W}_1 s_{ik} + \mathbf{b}_1) + \mathbf{b}_2), \\ \hat{a}_i &= \text{softmax}(a_i), \\ \mathbf{e}_i &= \sum_{k=1}^K \hat{a}_{ik} s_{ik} \end{aligned} \quad (4)$$

3.2 Sentence Selector

In the section, we utilize a hierarchical memory network [23] to construct sentence-based reasoning chains.

We define two phases for Sentence Selector: selecting a node and establishing a hop edge. In the selecting phase, the model extracts a sentence that is most relevant to the network memory state \mathbf{m} as the starting node of the current hop. During the establishing phase, the model updates \mathbf{m} to prepare for jumping the next node, which can be compared to generating the current jump edge.

We choose to use the left entity as the starting node of the inference chain, so the model initializes \mathbf{m} with the last state of \mathbf{L} and updates it with a Gated Recurrent Unit (GRU) [14].

Selecting a Node. At each hop h , the model calculates the correlation between each sentence encoding \mathbf{e}_i in \mathbf{E} and current memory state \mathbf{m}^h based on the bilinear-similarity and obtains a node selection distribution P_{sent} , which can be described as

$$p_i = \mathbf{e}_i^T \mathbf{W}_p \mathbf{m}^h,$$

$$P_{sent} = \text{softmax}(p). \quad (5)$$

Then, we choose the sentence $\mathbf{s}_i \in \mathbf{S}$ as the starting node of the current hop, where i satisfies

$$P_{sent}(i) = \max(P_{sent}). \quad (6)$$

Establishing a Hopping Edge. After selecting the starting node of h hop, the model calculates the bilinear-similarity of \mathbf{m}^h and each word \mathbf{s}_{ik} in \mathbf{s}_i and normalizes it to obtain a weight μ ; that is,

$$\begin{aligned} v_k &= \mathbf{s}_{ik}^T \mathbf{W}_m \mathbf{m}^h, \\ \mu &= \text{softmax}(v). \end{aligned} \quad (7)$$

Now, we use μ to calculate the weighted average $\bar{\mathbf{s}}_i$ of all the words in \mathbf{s}_i and then input it into a GRU cell to update \mathbf{m}^h , which can be described as

$$\begin{aligned} \bar{\mathbf{s}}_i &= \sum_{k=1}^K \mathbf{s}_{ik} \mu_k, \\ \mathbf{m}^{h+1} &= \text{GRU}(\bar{\mathbf{s}}_i, \mathbf{m}^h). \end{aligned} \quad (8)$$

Afterwards, we can combine the two sections together as a recurrent unit U ,

$$\left(\mathbf{s}_{h+1}, \mathbf{m}^{h+1} \right) = U(\mathbf{m}^h). \quad (9)$$

U can continuously select nodes by updating \mathbf{m} . Looping for U H times, we can get a H -hop reasoning chain $\mathbf{S}_{chain} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_H\}$ where each sentence \mathbf{s}_h is selected iteratively as a node by U in \mathbf{S} . To reduce the fortuity of reasoning chain generation, we repeat Sentence Selector M times to generate M possible H -hop reasoning chains for the model.

3.3 Answer Predictor

In the section, the model mainly predicts the probability of each candidate as the answer based on the H -hop reasoning chains obtained in Sentence Selector. Each chain may be a logical reasoning path from one entity to another.

Therefore, the model also introduces the question as auxiliary evidence to select the answer that meets the requirements of the question. Answer Predictor consists of two parts: reasoning chain information integration and calculating the probability distribution of answers.

Information Integration. Since the predicted answer exists in the last hop \mathbf{s}_H of a reasoning chain, we calculate the attention σ between the first $H - 1$ hop of chain and the question for each word in \mathbf{s}_H . Then, σ is used to compute the weighted average $\mathbf{x} \in \mathbb{R}^v$ of \mathbf{s}_H . The formulas can be expressed as

$$\mathbf{x} = \sum_{k=1}^K \mathbf{s}_{Hk} \sigma_k. \quad (10)$$

For calculating σ , we first horizontally stitch the top $H - 1$ hop of \mathbf{S}_{chain} to obtain \mathbf{s}_{fore} ; that is,

$$\mathbf{s}_{fore} = \mathbf{s}_1 \| \mathbf{s}_2 \| \dots \| \mathbf{s}_{H-1}. \quad (11)$$

Then we calculate an information vector δ^k though adopting a LSTM with an attention mechanism [24] to encode \mathbf{s}_{fore} and the top $k - 1$ words of \mathbf{s}_H . In the meanwhile, considering the impact of the question on σ , we calculate the α -correlation [3] ε^k of δ^k with the left entity and relationship, mathematically,

$$\begin{aligned} a_i^k &= \omega^T \tanh(\mathbf{W}_a \mathbf{s}_{fore}^i + \mathbf{W}_b \mathbf{v}^k + \mathbf{b}), \\ c^k &= \text{softmax}(a^k), \\ \mathbf{g}^k &= \sum_i c_i^k \mathbf{s}_{fore}^i, \\ \delta^k &= \text{LSTM}(\mathbf{s}_H^{k-1}, \mathbf{v}^{k-1}, \mathbf{g}^{k-1}), \\ \varepsilon^k &= \alpha(\delta^k, \mathbf{l}) + \alpha(\delta^k, \mathbf{r}) \end{aligned} \quad (12)$$

where \mathbf{v}^k is the hidden states of LSTM at the k th step, \mathbf{l} and \mathbf{r} are the final state of \mathbf{L} and \mathbf{R} respectively. In addition, α can be defined as

$$\alpha(x, y) = \mathbf{W}_{\alpha 1}^T ((\mathbf{W}_{\alpha 2} x + \mathbf{b}) \circ y), \quad (13)$$

where \circ represents element-wise multiplication.

Finally, ε integrating the information of \mathbf{S}_{chain} and the question can be used to calculate attention σ ,

$$\sigma = \text{softmax}(\varepsilon). \quad (14)$$

Probability Distribution Evaluation. After the above, we get a vector \mathbf{x} of highly integrated reasoning chains and problem information. Thus, we can use \mathbf{x} to calculate a probability distribution P_{answer} of candidate \mathbf{c}_i as the answer; that is,

$$\begin{aligned} \theta_i &= \mathbf{W}_{\theta 1} \text{Relu}(\mathbf{W}_{\theta 2} [\mathbf{c}_i; \mathbf{x}; \mathbf{c}_i \circ \mathbf{x}] + \mathbf{b}_{\theta 2}) + \mathbf{b}_{\theta 1}, \\ P_{answer} &= \text{softmax}(\theta), \end{aligned} \quad (15)$$

where Relu is the activation.

We calculate P_{answer} for all reasoning chains and get the answer probability distribution set $\tilde{P}_{answer} = \{P_{answer}^i\}_{i=1}^M$. Aggregating the results of all reasoning chains, the score of the candidate \mathbf{c}_η as the answer can be given as

$$\text{score}(\mathbf{c}_\eta) = \sum_{i=1}^M P_{answer}^i(\mathbf{c}_\eta). \quad (16)$$

4 Experiments

In the section, we describe the data sets used to evaluate the model, parameter settings, and experimental configurations firstly; additionally, we demonstrate the results and ablation studies of the proposed model.

4.1 Datasets

We use WikiHop and MedHop [8] data sets to evaluate our proposed model; in particular, we exploit the unmasked version of them.

WikiHop is a massive multi-hop MRC data set which provides about 43.8k samples for training set and 5.1k samples for development set. Each sample contains an average of 13.7 supporting documents, which can be divided into about 50 sentences and documents are collected from Wikipedia. The question of each sample contains an entity and a relationship. They form a triple of the WikiData knowledge base with the unknown answer that is contained in the provided candidate set.

MedHop is smaller dataset which consists of 1.6K samples for training set and 342 samples for development set. It mainly focuses on the domain of molecular biology and its each sample including a question, a document set and a candidate set has the same structure as the samples of WikiHop. And the difference is that each document set includes an average of 9.6 supporting documents, and can be divided into about 40 sentences.

In experiments, we use all samples in the training set to train our proposed model and all samples in the development set to adjust the hyper-parameters of the model.

4.2 Experimental Settings

We use NLTK [15] to divide the supporting document set into word tokens and sentence tokens in different granularity and the candidate set and the question into word tokens.

We use the 300-dimensional Glove pre-trained word embedding (with 840B tokens and 2.2 M vocabulary size) [12] to represent initial word tokens. The number of hidden units of all LSTM-RNN [22] is 100. We use dropout [25] with probability 0.5 for every trainable layer. We select top-10 documents which contains an average of 30 single sentences and 20 concatenation sentences after filtering by using the TF-IDF algorithm in each sample.

We use cross entropy loss to measure the level of model training, and use the Adam optimizer to train our model and set the learning rate at 0.001. We train 20k steps using four Nvidia 1080Ti GPUs. On each GPU, the batch size is fixed at 4, and the total batch size is 20. We use accuracy as an indicator for the multi-hop MRC task.

4.3 Result and Analysis

Table 1 presents the results of our proposed multi-hop MRC model on development set and test set¹ of WikiHop, and we compare it with the results that were reported in their original papers.

¹ We are in the process of obtaining the results on the hidden test set.

Table 1. Accuracy on the WikiHop development set and test set, where “-” denotes that the values are unavailable currently.

Model	Accuracy (%)	
	Dev	Test
BiDAF [26]	–	42.9
Entity-GCN [13]	64.8	67.6
CFC [7]	66.4	70.6
BAG [27]	66.5	69.0
EEPath [4]	67.1	–
EPAr [3]	67.2	69.1
SMR (ours)	68.3	–

We can observe that our proposed model achieves the highest accuracy of 68.3 on the development set for all the models in the table. Compared to the best previous result whose accuracy is 67.2, it is a 1.1 improvement on development set. It’s worth noting that our model no use pre-trained language models such as ELMO [16] and Bert [11] which has been shown to give MRC models a significant gain. Therefore, to be fair, the result of the proposed model doesn’t compare with those of the pre-trained language model.

We also show the results on MedHop in Table 2. We have a noticeable improvement on MedHop test set. In addition, our proposed model is more interpretable because the sentence-level reasoning chain it generates can be regarded as an explicit path for human reasoning.

Table 2. Accuracy on the MedHop test set, where the results marked “*” were originally reported by [8].

Model	Accuracy (%)
Max -mention*	9.5
Document -cue*	44.9
BiDAF [26]	47.8
Majority -candidate -per -query -type*	58.4
EPAr [3]	60.3
SMR (ours)	62.9

In order to reveal how SMR model based on sentence reasoning can realize reasoning and find the answer, we illustrate an example in Fig. 3 to visualize this process. In SMR, relevant supporting documents are screened out, and the sentence sets containing single

and concatenation sentences are obtained by sentence division. Relying on the sentences set, SMR constructs two different reasoning chains: $chain_1$ and $chain_2$. Through the SR, SS and AP modules, our model predicts the answer: ‘loon op zand’. It can be seen from Fig. 3 that the process of SMR predicting the answer constructs a reasoning path consistent with human cognition.

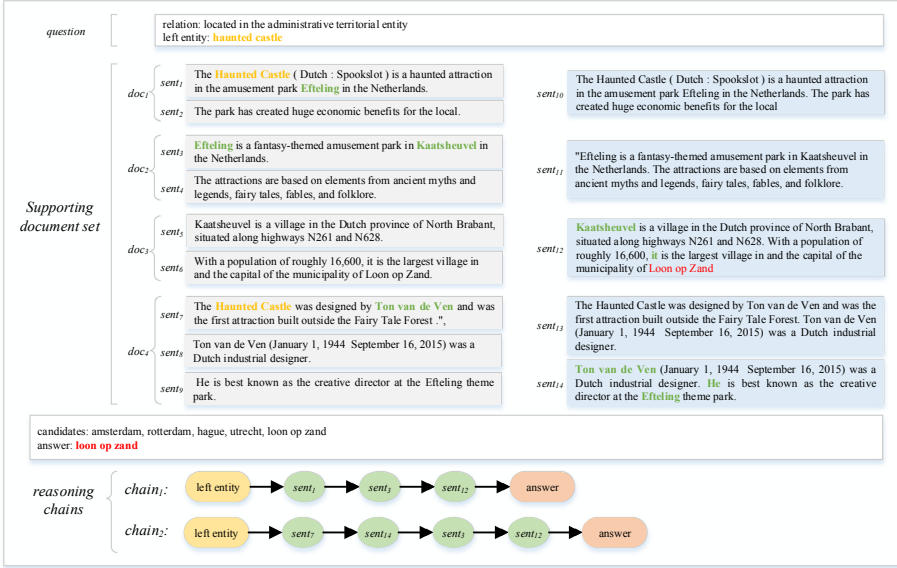


Fig. 3. Sample case of SMR reasoning process.

In the process of constructing the reasoning chains, our model uses self-attention [7] to integrate all the words in a sentence into a vector which represents the semantics of the sentence. EPAr [3] does the same at the document level as well. Because sentences have fewer words, less information is lost in the process than documents, which is the advantage of SMR compared to EPAr. EEPATH [4] takes out all possible paths as the basis for predicting the answer. Our model builds valid reasoning path by integrating sentence information and the obtained path has some logic, so our model has more accurate path and higher efficiency compared with EEPATH.

SMR use sentence sets which contain single and concatenation sentences to generate T-hop inference chains, which can deal with the pronouns among the sentences well. As $chain_1$ and $chain_2$ in Fig. 3, $sent_5$ and $sent_6$ are two single sentences from the same document, and $sent_{12}$ is the concatenation of the two sentences. In the reasoning process, $sent_3$ chooses $sent_{12}$ as the node of one hop instead of $sent_5$ or $sent_6$. Although containing the key word: ‘Kaatsheuvel’, $sent_5$ is difficult to reason to $sent_6$ because $sent_6$ used a pronoun ‘it’ to express the keyword but the model does not understand the meaning of the pronoun. And $sent_6$ contains important intermediate information for predicting the answer and must be a node in the chain of reasoning. Jumping from the $sent_3$ to $sent_{12}$

can not only capture the key word information contained in *sent*₅, but also match the pronoun in *sent*₆ with the key word.

Therefore, the existence of concatenation sentence can make the model more suitable for the situation where there are and there are no pronouns in the inference chain. If the concatenation sentence is too long, the process of integrating it into a vector will lose too much semantic information. Therefore, the model combines two adjacent single sentences into a concatenation sentence, which can satisfy most cases. Meanwhile, the existence of a single sentence avoids the model choosing unnecessary concatenation sentences as nodes such as the second node of *chain*₁ choosing *sent*₃ instead of *sent*₁₁.

4.4 Ablation Study

In order to better understand the contributions of different modules to the performance of our proposed model, we designed several ablation studies (Table 2) on the WikiHop development set.

If removing the sentence-based reasoning from the model, we will encode the documents directly using the self-attention mechanism [7] and replace sentence encoding **S**, **E** with the resulting document vectors. Then we carry out multi-hop reasoning at the document level and the accuracy of SMR will reduce by 1.1 absolutely. This proves the validity of our proposed reasoning at the sentence level for the multi-hop MRC task.

If we only use one reasoning chain in the model, that is, we don't repeat SC module, the accuracy of SMR will decrease by 2.2%. This demonstrates that constructing multiple inference chains can reduce the randomness of reasoning path generation indeed. If the TF-IDF algorithm isn't used to filter the documents, the accuracy of the model we obtained will be reduced by 1.9%. This proves that removing some irrelevant articles can help to get more accurate reasoning chains, while the model will occupy fewer computing resources and achieve higher training efficiency due to the reduction of supporting documents (Table 3).

Table 3. Ablation results on the WikiHop development set.

Model	Accuracy (%)	
	Dev	Δ
Full model	68.3	–
- document-based reasoning	67.2	1.1
- one reasoning chain	66.1	2.2
- TF-IDF algorithm	66.4	1.9
- single sentence	65.6	2.7
- concatenation sentence	65.1	3.2

We also investigate the effect of single sentences and concatenation sentences on the model effect. Specifically, we use single-sentence set \mathbf{D}^o instead of all sentence set \mathbf{D} for T-hop reasoning and the accuracy is reduced by 2.7%. At the same time, we also replace \mathbf{D} with the concatenation sentence set \mathbf{D}^b and the accuracy is reduced by 3.2%. According to the ablation, we can infer that using only the single sentence set may prevent the model from understanding the meaning of pronouns that may exist. However, the merely using concatenation sentence set can lead to excessive interference between sentences in the reasoning process. Therefore, the combined using of single sentences and concatenation sentences can better cope with the presence of pronouns in adjacent sentences and reduce the negative influence between sentences to improve the performance of the model.

5 Conclusion

In this paper, we have proposed a multi-hop MRC model sentence-based reasoning named SMR, where sentences play a pivotal role in constructing reasoning chains. Besides, we innovatively use concatenation sentence to deal with the semantic encoding of pronouns in a single sentence, which has been proved by experiments to improve the model effect significantly. We also presented that SMR can illustrate its reasoning through hopping across multiple sentences. The superior performance on WikiHop and MedHop data sets verifies the effectiveness of SMR.

In the future, we will verify the effect of SMR after adding the pre-trained language model, although it has achieved excellent performance. We also plan to focus on generative models incorporating sentence-based reasoning like Masque [1]. Moreover, it is of interest to investigate other types of multi-hop MRC datasets, e.g., the newly proposed benchmark HotpotQA [9].

Acknowledgement. This work was partially supported by NSFC under grants Nos. 61872446, 61902417 and 71971212, and PNSF of Hunan under grant No. 2019JJ20024.

References

1. Nishida, K., et al.: Multi-style generative reading comprehension. *ACL* (1), 2273–2284 (2019)
2. Wang, C., Jiang, H.: Explicit utilization of general knowledge in machine reading comprehension. *ACL* (1), 2263–2272 (2019)
3. Jiang, Y., Joshi, N., Chen, Y.-C., Bansal, M.: Explore, propose, and assemble: an interpretable model for multi-hop reading comprehension. *ACL* (1), 2714–2725 (2019)
4. Kundu, S., Khot, T., Sabharwal, A., Clark, P.: Exploiting explicit paths for multi-hop reading comprehension. *ACL* (1), 2737–2747 (2019)
5. Tu, M., Wang, G., Huang, J., Tang, Y., He, X., Zhou, B.: Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. *ACL* (1), 2704–2713 (2019)
6. Ding, M., Zhou, C., Chen, Q., Yang, H., Tang, J.: Cognitive graph for multi-hop reading comprehension at scale. *ACL* (1), 2694–2703 (2019)
7. Zhong, V., Xiong, C., Keskar, N.S., Socher, R.: Coarse-grain fine-grain coattention network for multi-evidence question answering. *ICLR* (Poster) (2019)

8. Welbl, J., Stenetorp, P., Riedel, S.: Constructing datasets for multi-hop reading comprehension across documents. *TACL* **6**, 287–302 (2018)
9. Yang, Z., et al.: HotpotQA: a dataset for diverse, explainable multi-hop question answering. *EMNLP*, pp. 2369–2380 (2018)
10. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100, 000+ questions for machine comprehension of text. *EMNLP*, pp. 2383–2392 (2016)
11. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT* (1), 4171–4186 (2019)
12. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. *EMNLP*, 1532–1543 (2014)
13. De Cao, N., Aziz, W., Titov, I.: Question answering by reasoning across documents with graph convolutional networks. *NAACL-HLT* (1), 2306–2317 (2019)
14. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP*, pp. 1724–1734 (2014)
15. Bird, S., Loper, E.: NLTK: the natural language toolkit. *ACL (Poster and Demonstration)* (2004)
16. Peters, M.E., et al.: Deep contextualized word representations. *NAACL-HLT*, pp. 2227–2237 (2018)
17. Joshi, M., Choi, E., Weld, D.S., Zettlemoyer, L.: TriviaQA: a large scale distantly supervised challenge dataset for reading comprehension. *ACL* (1), pp. 1601–1611 (2017)
18. Onishi, T., Wang, H., Bansal, M., Gimpel, K., McAllester, D.A.: Who did what: a large-scale person-centered cloze dataset. *EMNLP*, pp. 2230–2235 (2016)
19. Hill, F., Bordes, A., Chopra, S., Weston, J.: The goldilocks principle: reading children’s books with explicit memory representations. *ICLR* (2016)
20. Battaglia, P.W., et al.: Relational inductive biases, deep learning, and graph networks. *CoRR* abs/1806.01261 (2018)
21. Srivastava, R.K., Greff, K., Schmidhuber, J.: Highway Networks. *CoRR* abs/1505.00387 (2015)
22. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
23. Chandar, S., Ahn, S., Larochelle, H., Vincent, P., Tesauro, G., Bengio, Y.: Hierarchical Memory Networks. *CoRR* abs/1605.07427 (2016)
24. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. *ICLR* (2015)
25. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
26. Seo, M., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional attention flow for machine comprehension. *ICLR* (2017)
27. Cao, Y., Fang, M., Tao, D.: BAG: bi-directional attention entity graph convolutional network for multi-hop reasoning question answering. *NAACL-HLT* (1), 357–362 (2019)