A Sentence-Based Circular Reasoning Model in Multi-Hop Reading Comprehension

LIJUN HUO AND XIANG ZHAO, (Member, IEEE)
National University of Defense Technology, Changsha 410072, China
Corresponding author: Xiang Zhao (xiangzhao@nudt.edu.cn)

This work was supported in part by the NSFC under Grant 61872446, Grant 61902417, Grant 71690233, and Grant 71971212; and in part by the NSF of Hunan Province under Grant 2019JJ20024.

ABSTRACT Multi-hop Machine Reading Comprehension (MRC) requires models to mine and utilize relevant information from multiple documents to predict the answer to a semantics-related query. The existing researches resort to either document-level or entity-level inferences among relevant information, but such practices may be too coarse or too subtle to result less accurate understanding of the text. To address this issue, this research proposes a Sentence-based Circular Reasoning (SCR) approach, which starts with sentence representation and then unites the query to establish a reasoning path based on a loop inference unit. Further, the model synthesizes the information existing in the reasoning path and receives a probability distribution for selecting the correct answer. In addition, this study proposes a nested mechanism to extend the probability distribution for weighting. And it is proven that this mechanism can assist the model to perform better. Some experiments evaluate SCR on two popular multi-hop MRC benchmark datasets, WikiHop and MedHop, achieving 71.6 and 63.2 in terms of accuracy, respectively, and thus exhibiting competitive results compared with the state-of-the-art model. Additionally, qualitative analyses also demonstrate the validity and interpretability of SCR.

INDEX TERMS Machine reading comprehension, multi-hop question answering, sentence representation, text understanding.

I. INTRODUCTION

As an important technique in the field of Natural Language Processing (NLP), machine reading comprehension can measure how well machines understand complex text. Compared with such basic tasks as named entity recognition, relationship extraction and so on, MRC is a higher-level NLP task, requiring the models to have a deeper ability in semantic understanding, more complete information integration and even reasoning. Generally, MRC is purposed to predict the correct answer to a text-related query based on a given document set. So it demands selecting, extracting and fusing massive text information, thus imposing a great challenge to researchers.

Over recent years, researchers have paid great attention to MRC, while numerous high-quality data sets have been proposed to evaluate the progress of MRC, such as SQuAD [1] and NarrativeQA [2]. But most of the data sets proposed are for single-hop tasks, meaning that only one supporting document for each query is used to find the answer. In real scenarios, however, any single query always corresponds to multiple documents. Therefore, the models are required to jump reasonably among multiple documents to search valid information related to the query and actively utilize them to predict the answer. Obviously, multiple documents increase the difficulty of the task and impose higher requirements on the models. In terms of the number of supporting documents, MRC tasks can be divided into single-hop MRC and multi-hop MRC. For single-hop MRC, researchers have made tremendous progress and proposed many excellent models [3]–[5], some of which are even comparable to humans in performance. In contrast, progress in multi-hop MRC is limited, with much room for improvement, which is just the focus of this study.

As an especially challenging task, multi-hop MRC shows its difficulties mainly in the following three aspects: Firstly, because each query corresponds to a supporting document set, most models are difficult to directly deal with such large
scales of supporting documents. In addition, only a small part of the document set contains valid information, while the rest is irrelevant, causing serious interference in answer prediction. Secondly, due to wide distribution of information, models have to continuously jump among multiple documents to extract useful information and form an information flow for raising an answer. This process is a big test for reasoning abilities, which have always been a bottleneck urgently needed to break through in the MRC field. Thirdly, multi-hop MRC requires performing repeated information discoveries and fusions, which are multiple progressive information coagulations. And the errors generated in any coagulation will be propagated, eventually leading to an accumulation of errors and great uncertainty.

In view of these limitations, this study abandons the current mainstream multi-hop inference methods based on document levels [6]–[8] or entity levels [9], [10]. Rather, it will choose to hop through sentences and gradually build an explicit path based on sentences, so as to deliver the jumps among semantics. Then it will evaluate the path to obtain the final answer. This approach can alleviate irrelevant information interference due to the conciseness of the sentences, while reducing errors generated from appropriate granularities of the path. And this study believes that treating sentences as inference nodes is more reasonable than treating documents or entities. Fig.1 shows the three paths to treat documents (path2), sentences (path1) and entities (path3) as nodes, respectively. All these paths are created to reply the same query. It can be seen that when using documents as nodes, although the path may completely contain the key inference information, it still has a lot of unrelated information, which can produce a great degree of redundancy. Such an inaccurate path is so rough that it will cause strong interference. On the other hand, when using entities as nodes, although the path is precise and concise, it has no original logic due to too much lost information, thus insufficient to support the models to complete the task. However, if the path uses sentences as nodes, it has obvious logic and less information redundancy, not only delivering great assistance to acquiring answers, but also explaining the reasoning process well.

Taking these requirements into account, this study proposes a Sentence-based Circular Reasoning model, named SCR, which uses sentence inference to construct an information path and consists of three modules: Sentence Encoder (SE), Path Generator (PG) and Path Evaluator (PE). Specifically, this study uses the SE to obtain the sentence representation, and then use the PG to iteratively infer among sentences of multiple documents under the guidance of the query, so as to continuously produce nodes, which are combined to form a path. Finally, the PE evaluates the obtained path and predicts an answer. In this process, the PG imitates the procedure of human inference and constructs a path which integrates the important information related to the query. Crucial for the whole model, it is the core component of SCR. The PE integrates all the necessary information from the bottom layer, thus directly affecting the correctness of answers. So, it is indispensable as well.

The goal of the SE is to get sentence representation, accomplished in previous study by using the self-attention mechanism [11]. The mechanism converts a sentence into a vector by weighting all word vectors (can be seen as a matrix) in the sentence. That usually only has a probability distribution.
However, given that the sentence has multiple aspects of semantic characteristics and the sentence vector obtained from a distribution cannot fully express the semantics of all aspects, so, this study proposes a Nested Mechanism (NM) to assign multiple distributions to the sentence, with each distribution representing one aspect. In this way, the weighted vector can reasonably express as many semantic features as possible. This study introduces the implementation details of the NM in III-D and, further explains the necessity and effectiveness of the NM in IV-D. In order to extend the advantages of multiple distributions, we also apply the NM to the weighting operations of PG and PE.

In summary, the proposed model SCR consists of three modules: SE, PG and PE, as well as a mechanism: NM. This study makes the following contributions:

- It proposes to leverage sentence-based reasoning for multi-hop MRC, which builds an information path that can assist the model to accomplish the task.
- It proposes a novel mechanism, named NM, and proves its improvement to the model performance.
- It achieves the state-of-the-art performance on two popular multi-hop datasets and explains the inference process of SCR.

II. RELATED WORK

Over recent years, various multi-hop MRC datasets have been released publicly, and they all demand models to understand text semantics and find their internal relationships. However, their forms are different, with their tasks mainly divided into the following three types: 1) Generative tasks, such as HotpotQA [12], TriviaQA [13] and CMRC [14]. They usually give \{query, document set, answer\}, where the query is natural language text. Furthermore, the models need to extract a span from the document set or generate a span from the vocabulary as the answer. 2) Multi-choice tasks, such as QAngaroo WikiHop and MedHop [15] and RACE [16]. They usually give \{query, document set, answer, candidate set\}, where the answer is an entity presented in the given candidate set and the query consists of an entity and a relationship. The goal is to choose the correct answer based on the given supporting information. 3) Complementary tasks, such as Who What [17] and Children’s Book Test [18]. They usually ask the models to predict missing words/entities in queries.

Based on the characteristics of these data sets, researchers have developed various models to handle the tasks. For example, [15] fuses multiple documents into a long one, and then uses the typical single-hop MRC model [3] with the bidirectional attention mechanism to deduce the answer. However, the result of the model is far less accurate than that in the single-hop task, because the documents after fusion are too long and the model has no information jumping capability. With the assistance of knowledge guidance, [19] enables the model to integrate the semantics of documents, but the approach is difficult to extend, since the external knowledge tends to be limited to a specific field. Focused on reasoning, [10] gathers all possible reasoning paths based on the entities contained in the documents, and then scores all paths to identify the correct one. However, this method would extract many invalid paths, thus bringing in interference and wasting the computing resources. [9], [20], [21] use graph neural networks [22] to obtain the relationship between entities, or add self-attention mechanism [11] into the model, so as to obtain a gain in the result. These three methods provide good interpretability by constructing non-explicit paths or extracting some relevant supporting facts. Nevertheless, as the number of inferences increases, the complexity of models will rise sharply due to the iteration of cumbersome message passing algorithm, resulting in low efficiency.

This research focuses on the multi-choice task and utilizes WikiHop and MedHop [15] datasets, while adopting sentence-based reasoning to construct a precise inference path. It also proposes and applies the NM to optimize the sentence representation as well as expressions of some other modules. By making a dynamic inference framework, this study reasonably combines the three modules of SCR. As a result, these novel designs and reasonable granularity selection of the proposed model can address the problems of the above models to a certain degree.

III. MODEL

This section introduces the three-module multi-hop MRC model and the Nested Mechanism. The overall architecture of SCR is as Fig.2.

Before delving into the details, the task to be investigated in this study is formally defined as follows.

Task Definition: In the WikiHop and MedHop [15] data sets, each sample has a supporting document set \(Z\) and a related query \(q\). In particular, the form of the query is \((l_e, r, ?)\), where \(l_e\) is the left entity and \(r\) is the relation between the left entity and the unknown right entity, i.e., the answer. In addition, it also provides a candidate set \(C = \{c_i\}^H_{i=1}\), which contains the correct answer. The purpose of this task is to choose the correct answer from the candidate set based on the given query and the supporting document set.

Next, the proposed model is explained, which first (III-A) encodes the semantic information of the text to get the vector representation required for the subsequent operations, then (III-B) constructs an inference path based on these encoded texts. Finally (III-C) SCR interacts the path information with the query encoding, and joins the candidate set to compute a probability distribution which can be used to select the answer. In addition, the proposed NM is illustrated at the end.

A. SENTENCE ENCODER

Here, the text preprocessing and semantic encoding methods first are introduced. Then, the documents are split into sentences and converted into vectors to obtain the sentence representations.
1) PREPROCESSING AND BASIC-SEMANTIC LAYER

This layer has two main objectives: 1). Filtering the supporting document set. It can reduce the number of irrelevant documents and decrease interference; 2). Importing word encoding. It can convert all texts into vectors and perform semantic encoding to build the underlying foundation for the model.

Specifically, this study first applies a two-layer TF-IDF algorithm to filter the supporting document set. In the first layer, this study calculates the TF-IDF cosine similarity of each document and the query to take out a document with the largest similarity. In the second layer, this study calculates the TF-IDF cosine similarity of all remaining documents and the one selected by the last layer, and takes out the top $N - 1$ documents with the largest similarity. All selected documents form a new supporting document set $Z = \{z_n\}_{n=1}^N$.

Then, this study combine the pre-trained Glove vector [23] and character n-gram vector [24] as the initial word embedding, orderly input into a Highway Network [25], a bidirectional LSTM network with $v$ hidden units [26] to obtain the word encoding of all the texts.

$\mathbf{L} \in \mathbb{R}^{Q_l \times v}$, $\mathbf{R} \in \mathbb{R}^{Q_r \times v}$, $\mathbf{X} \in \mathbb{R}^{N \times J \times v}$ is used as the word encoding of $l_e$, $r$, $Z$ respectively, where $Q_l$, $Q_r$, $J$ are the word-level lengths of $l_e$, $r$, $Z$ respectively. In addition, since each candidate can be found in the supporting document set, this study takes out the encoding corresponding to $c'_{\kappa}$ in $\mathbf{X}$, averages it at the word-level and then gets $\mathbf{e}_c \in \mathbb{R}^v$ as the semantic encoding of $c'_{\kappa}$.
2) SELF-ATTENTION LAYER
The encoding $X$, obtained in the last layer, is based on word. But in order to simulate human multi-hop reasoning more realistically, this study needs to compute sentence-level semantic encoding. Therefore, the purpose of this layer is to obtain sentence-level encoding. Specifically, this study first splits all the documents into sentences to get the sentence set $D = \{d_i\}_{i=1}^j$ s.t. $d_i \in \mathbb{R}^{K \times y}$, where $j$ is the number of sentences contained in $Z, K$ is the number of words that make up a sentence and $d_{ik}$ is the corresponding word encoding in $X$. Then, a self-attention mechanism [11] is applied to implement the vector representation of sentences, and the representation is denoted as $S = \{s_i\}_{i=1}^j$, where $s_i \in \mathbb{R}^y$ can be given by the following formulas:

$$a_{ik} = \tanh(W_2 \tanh(W_1 d_{ik} + b_1) + b_2);$$

$$w_i = \text{softmax}(a_i),$$

$$s_i = \sum_{k=1}^K w_{ik} d_{ik}.$$  

Similarly, we apply a self-attention to $L$ and $R$ to complete the encoding of the query. For the result, this study uses $q_f \in \mathbb{R}^y, q_r \in \mathbb{R}^y$ as the vector representation of the left entity and the relation.

1) ATTENTION MODULE
The module consists of two sub-layers: a multi-head attention layer [27] and a position-wise feed-forward network. Each sub-layer is placed inside a residual block [28].

The multi-head attention layer is an implementation of co-attention mechanism, which can be formally described as:

$$Q_i, K_i, V_i = \text{split}_{\text{head}}(QW^q_i, KW^k_i, VW^v_i),$$

$$D_i = \text{softmax} \left( \frac{Q_i K_i^\top}{\sqrt{V}} \right) V_i;$$

$$H = D_1 \| D_2 \| \ldots \| D_{\text{head}},$$

where $Q, K, V$ are its three input and this study feeds them the same variable. Also split$_{\text{head}}$ indicates splitting equally into head shares in the final dimension, $\|$ indicates concatenation and $W^q_i, W^k_i, W^v_i$ are trainable weight matrices.

The feed-forward network consists of two linear transformations, with a GELU [29] activation function in between. Then the output is acquired through a layer normalization [30]. The whole sub-layer is described as FFN for simplicity and its input is $H$:

$$E = \text{FFN}(H)$$

At step $t (t > 1)$ of the PG, the input of the sentence attention module is the output of its step $t-1$, and $S$ is used as the first step’s input. Its output is represented by $\hat{S}_t = \{s_{ti}\}_{i=1}^j$. The module updates the sentence representation by encoding the deeper semantic information.

The path attention module is used to update the path representation in each loop. Its input is the path generated by step $t-1$ and $l_t$ at the first step. $P_t = \{p_1, p_2, \ldots, p_t\}$ is used as the output of the module.

2) GENERATION MODULE
The generation module is an inference network based on an LSTM [26] unit. At step $t$, it integrates the raw path and the query to calculate the probability $\epsilon_t$ of each sentence as the next hop node, then $\epsilon_t$ is used to weighted average the sentence set. Thereby a sentence information fusion vector $\eta_t$ can be got. Further, this study puts $\eta_t$ through a position-wise feed-forward network and splices it at the end of the current raw path to form a new path, thus completing the inference of step $t$. There is:

$$\eta_t = \sum_{i=1}^j \epsilon_t \hat{s}_{ti},$$

$$n_t = \text{FFN}(\eta_t),$$

$$P_t = P_{t-1} \| n_t.$$  

For $\epsilon_t$, the raw path $P_{t-1}$ is first spliced with each sentence $\hat{s}_{ti}$ to form $I$ hypothetical paths $G = \{g_1, g_2, \ldots, g_I\}$. Then this study inputs them into a LSTM unit respectively, and takes the final hidden state $\gamma_{hi}$ as the judgment vector of $\hat{s}_{ti}$, $\gamma_{hi}$ is a summary of the hypothetical path $g_{ti}$, which can be used to measure the reasonableness of the current hypothesis. Therefore, this study adopts a two-layer fully connected network to convert $\gamma_{hi}$ to the reasonableness. It also

![Figure 3. The interconnection of the three sub-modules of PG.](image-url)
incorporates the influence of the query by calculating the \( \alpha \)-similarity between \( y_i \) and \( q, y_i \) and \( q \). At the same time, in order to explicitly reflect the continuity of the path, the influence of the last node on the current node selection is also added. The above process can be formally described as:

\[
g_i = P_{i-1} \| \tilde{s}_i, \\
y_i = \text{LSTM}(g_i), \\
\theta_i = W_2 \tanh(W_1 y_i + b_1) + b_2, \\
m_i = \alpha(y_i, q) + \alpha(y_i, q_i) + \alpha(y_i, n_{i-1}) + \theta_i; \\
\varepsilon_t = \text{softmax}(m_i),
\]

where \( \alpha \) can be defined as:

\[
\alpha(x, y) = W_{\alpha 2}(W_{\alpha 1}x + b) \circ y,
\]

where \( \circ \) represents the element-wise multiplication.

C. PATH EVALUATOR

The path evaluator is a recurrent structure based on the nodes of the path created by the PG. Specifically, this study first uses a GRU [31] cell to encode the path information needed for the current step. Considering the order and integrity of the path, it concatenates the \( k \) step node \( p_{k-1} \), the \( k \)-th node \( p_k \), the hidden state of the GRU at \( k \)-th step and the weighted average of the first \( k = 1 \) nodes as the input of the GRU at \( k \)-th step. Then, the \( \alpha \)-similarity between the GRU’s output and the query is calculated to obtain the weight of the \( k \)-th node. After finishing all the iterations, this study uses \( \varphi_T \) to condense the path into a scoring vector \( \lambda \), which integrates the information on the path and query. Finally, \( \lambda \) is used to score each candidate and propose an answer that has the highest score. The above process can be formally described as (The length of \( \varphi \) is equal to \( k \)):

\[
r_k = \sum_{i=1}^{k} \varphi_{ki}p_i, \\
u_k = \text{GRU}(p_{k-1}, p_k, u_{k-1}, r_{k-1}), \\
\phi_k = \alpha(u_k, q) + \alpha(u_k, q_i), \\
\varphi_k = \text{softmax}(\phi); \\
\lambda = \sum_{i=1}^{T} \varphi_i p_i, \\
\text{Score}(c_k) = \beta(\lambda, c_k),
\]

where \( u_k \) is the GRU’s hidden state at \( k \)-th step. And \( \beta \) can be defined as:

\[
\beta(o, u) = W_{\beta 2} \text{relu}(W_{\beta 1}[o; u \circ u] + b_{\beta 1}) + b_{\beta 2},
\]

where \( \text{relu} \) is the activation, and \( W_{\beta 1}, W_{\beta 2}, b_{\beta 1} \) and \( b_{\beta 2} \) are trainable weight matrices.

D. NESTED MECHANISM

Nested mechanism is mainly suitable for mapping sequences into a vector. It can provide multiple probability distributions for sequences in different feature spaces when sequences need to be weighted average. It works by nesting the applied module and plays a role by operating the input (one vector sequence) and output (one vector). Specifically, the NM first maps the input sequence of the Y into \( h \) feature subsequences through \( h \) one-layer Full Connected networks (FCa) where Y is a module adopting the NM. Then, it replaces the original input with these subsequences respectively, and uses the Y to calculate a probability distribution for each feature subsequence \( \rho_i \), and weight to obtain a summary \( \mu_i \) of each subsequence (output). Finally, it compresses the concatenation of these outputs to a \( v \)-dimension vector by another one-layer Full Connected network (FCb) and a layer normalization [30]. Thus, this study can replace the original output of Y with the obtained vector. It derives the process as follows:

\[
\rho_i = \tanh(FCa(input)), \\
\mu_i = Y(\rho_i); \\
v = FCb(\mu_1 \| \mu_2 \| \ldots \| \mu_k), \\
output = \text{LN}(v),
\]

where FCa and FCb have different parameter sizes. It is worth noting that the dimension of output remains unchanged for \( v \) after using the NM.

In this study, the NM is applied for Self-Attention Layer, Generation Module and Path Evaluator. When these modules act as Y, the parameters of the NM are as follows:

|       | input | output | \( h \) | \( v \) |
|-------|-------|--------|--------|--------|
| Self-Attention Layer | D | S | 12 | 100 |
| Generation Module | G | \( \eta \) | 8 | 100 |
| Path Evaluator | \( P_T \) | \( \lambda \) | 8 | 100 |

IV. EXPERIMENT

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.

A. DATASETS

We use WikiHop and MedHop [15] data sets to evaluate our proposed model; in particular, we exploit the unmasked version of them. WikiHop, a massive multi-hop MRC data set, provides about 43.8k samples for the training set and 5.1k samples for the development set. Each sample contains an average of 13.7 supporting documents, which can be divided into about 50 sentences. And the documents are collected from Wikipedia. The query of each sample contains an entity and a relationship. They form a triple of the WikiData knowledge base with the unknown answer, which is contained in the candidate set provided. MedHop is a smaller dataset, consisting of 1.6K samples for the training set and 342 samples for the test set. Mainly focused on the domain of molecular biology, each sample of it includes a query, a document set and a candidate set, with the same structure as the samples in WikiHop. And the difference between them 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.

B. EXPERIMENTAL SETTINGS
We use NLTK [33] to divide the supporting document set into word tokens and sentence tokens in different granularities, and the candidate set and the query are into word tokens.

We use the 300-dimensional Glove [23] pre-trained word embedding (with 840B tokens and 2.2M vocabulary size) to represent initial word tokens and use random array embedding for out of vocabulary. The number of hidden units of LSTM and GRU in SE is 100 and others are 200. We set the dimension of the character n-gram embedding [24] to 100. We use dropout [34] with probability 0.1 for every trainable layer. We select top-8 documents which contain an average of 30 sentences after filtering by using the TF-IDF algorithm in each sample. We set the value of head to 8 for multi-head attention layer. We set hop to 6 and receive a 6-hop inference path.

We use cross entropy loss to measure the level of model training and use the Adam [35] optimizer to train our model. We set the initial learning rate at 0.001 and decrease to the original 0.8 every 2k steps. We train 30k steps using two P100 GPUs and the batch size is fixed at 64. We use accuracy as the indicator for the multi-hop MRC task.

C. RESULT AND ANALYSIS
Table 1 presents the results of our proposed multi-hop MRC model on development set of WikiHop, and we compared them with the latest results reported in other original papers.

We can observe that our proposed model achieves the highest accuracy of 71.6 on the development set for all models in the table. Compared to the best previous result whose accuracy is 70.1, our model gets a significant improvement on development set. It’s worth noting that our model doesn’t use pre-trained language models such as ELMO [39] and Bert [40] which has been well known to provide a significant gain for machine reading comprehension and question answering. Despite this, our model still performs better than many used. But for fairness, we don’t compare SCR with those using pre-trained language models.

Next, we show the results on MedHop in Table 2. We have a noticeable improvement on MedHop test set.

In addition, we also provide an instance (Fig.4) to explain the reasoning process of SCR. It can be found that these extracted sentences have at least one of the following characteristics: 1). The sentences are explicitly related to the relation: located in the administrative territorial entity left entity: zoo lake candidates: Johannesburg, south, gauteng, africa answer: gauteng

FIGURE 4. A sample whose answer is predicted correctly by SCR. The sentences of the path are all with a large probability in the distribution $\epsilon$ of per cyclic. In fact, the path also contains some other sentences with small probability, but these showed sentences are the main components and have a decisive influence on the choice of the answer.
left entity \((\text{sent}_1, \text{sent}_2)\); 2). The sentences are semantically related to the relation (The semantics of all sentences are related to the location information expressed by the relation); 3). Sentences are explicitly related to each other (\(\text{sent}_3, \text{sent}_4, \text{sent}_5, \text{sent}_6, \text{sent}_7, \text{sent}_8\)). We proposed model can accurately extract these sentences with such characteristics, which is essential for correct prediction of the answer. Meanwhile, SCR connects them logically to construct a weak information flow, which can guide the path evaluator to find the exact answer.

Of course, the success of SCR also benefits from the precise sentence representation. We add the NM to the self-attention mechanism, so that the sentence vectors can represent the semantics more comprehensively and reasonably. Compared with [6], [11], which only use the self-attention mechanism, we have the advantages in this regard. And we also make a detailed analysis of the NM in the next section.

Furthermore, the proposed model constructs an effective information path, each constituent node of which is selected based on the query and the current path. However, EEPath [10] catches all possible entity paths that include many invalid paths and waste computing resources. Also, compared with the models using GNN [22] such as [9], [20], the path can provide better interpretability.

What’s more, our novel model structure design is the basis of the strong performance. SCR can dynamically encode the sentences and the raw path in the reasoning progresses. In this way, the constructed path will notice the deeper level of semantic information as the cycle increases, and the semantic-based reasoning will gradually deepen. Compared with the static structures of encoding-decoding, such as [27] which is one encoding for all cycles, our dynamic structure is more suitable for continuous inference and more satisfied the needs of multi-hop MRC tasks.

**D. ABLATION STUDY**

In order to better understand the contributions of different modules to the performance of SCR, we design several ablation studies (Table 3) on the WikiHop development set.

| Model                        | Accuracy (%) |
|------------------------------|--------------|
|                              | Dev | Δ    |
| Full model                   | 71.6 | -    |
| - TF-IDF algorithm           | 69.4 | 2.2  |
| - Sentence-level reasoning   | 66.9 | 4.7  |
| - Dynamic encoding           | 69.3 | 2.3  |
| - Full NM                    | 68.9 | 2.7  |
| - A part of NM               | 70.5 | 1.1  |

We first remove the filtering of the TF-IDF algorithm on documents; as a result, the accuracy of SCR is reduced by 2.2. This proves that culling some irrelevant documents may help decrease disturbance, while the model consumes fewer computing resources and delivers higher training efficiency due to the streamlining of supporting documents.

We then replace sentence-level reasoning with the document-level reasoning and use a self-attention with the NM at the document-level. The accuracy of SCR is thus reduced by 4.7, illustrating the rationality of sentence reasoning. At the document-level, the granularity is too large, and the path is ambiguous and contains too many possibilities. Of course, we also try the word-level reasoning, but the granularity is too subtle and, each word’s information is limited, resulting in poor performance.

In addition, we replace dynamic encoding with static encoding, that is, to encode the sentence \(T\) times at once instead of following the inference step by step. In this way, the accuracy of SCR is reduced by 2.3. This result proves that dynamic encoding is more adaptable to the cyclic process, which can continuously inject new blood into the model and make it better respond to the changing reasoning environment. This also confirms our analysis in the last section.

Finally, we discuss the nested mechanism. First, we remove the NM of all modules, and the accuracy of SCR is reduced by 2.7, proving the effectiveness of the NM. Then, we only use the NM for the self-attention of the sentences. Compared with not using it at all, the accuracy of SCR is improved by 1.6, showing that the NM can better characterize the semantics of a sentence by a vector.

To illustrate the working principle of the NM, we depict a heat map about the self-attention using the HM in Fig.5. It is well known that all word encoding come from the same variable space. And the same dimension of different word encoding represents a similar semantic space, for instance, Dimension 1 represents shapes, Dimension 2 represents colors, and so on. The weighted combination of some dimensions can be used to represent the features of words in some aspects. For example, the combination of shapes and colors can characterize the category to some extent. Words have many features. When weighting average a word encoding sequence into a vector, we hope to highlight the strong features with obvious semantics in each word, and weaken those that are not obvious. If only one distribution is used for whole sequence, we cannot distinguish between strong and weak features. Obviously, this is unreasonable. However, the NM assigns a distribution to each feature, thus alleviating this problem.

In Fig.5, we use a linear layer to extract the 12 sub-semantic (horizontal) features for each word encoding, and calculate a distribution for each feature sequence (vertical). Based on it, we can consider the Feature 3 as a location feature, because Distribution 3 gives the greatest weight to ‘located’ with obvious geographical information. Also, the Feature 9 can be considered as a category feature, because the Distribution 9 gives the greatest weight to ‘biomedical’ with obvious species information. However, ‘located’, which is not clearly related to category, has a small weight. For the self-attention mechanism, the NM assists it to combine the strong features of all words into a vector, which can reasonably represent
sentence semantics from multiple feature spaces. For others, the NM can consolidate their expressions through the similar principle.

V. CONCLUSION

In this paper, we propose a multi-hop MRC model based on sentence reasoning, named SCR, in which sentences play a pivotal role in constructing an information path. Besides, we innovatively propose the nested mechanism to systematically represent semantics, which has been proved by experiments to be able to improve the model performance significantly. The high accuracy rates on both WikiHop and MedHop data sets verify the effectiveness of SCR. Through an instance, we also present that SCR can illustrate its inference process.

In the future, we will extend our model to more data sets, e.g., the newly proposed benchmark HotpotQA [12]. We also plan to focus on the generative models incorporating sentence-based inferring like Masque [42].

REFERENCES

[1] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ questions for machine comprehension of text,” in Proc. Conf. Empirical Methods Natural Lang. Process., Austin, Texas, USA, 2016, pp. 2383–2392, doi: 10.18653/v1/d16-1264.

[2] T. Koai ský, J. Schwarz, P. Blunsom, C. Dyer, K. M. Hermann, G. Melis, and E. Grefenstette, “The NarrativeQA reading comprehension challenge,” Trans. Assoc. Comput. Linguistics, vol. 6, pp. 317–328, Dec. 2018. [Online]. Available: https://transacl.org/ojs/index.php/tacl/article/view/1197

[3] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, “Bidirectional attention flow for machine comprehension,” in Proc. 5th Int. Conf. Learn. Represent., Toulon, France, Apr. 2017, pp. 1–4. [Online]. Available: https://openreview.net/forum?id=HJ0UKP9ge

[4] S. Wang and J. Jiang, “Machine comprehension using match-lstm and answer pointer,” in Proc. 5th Int. Conf. Learn. Represent., Toulon, France, Apr. 2017, pp. 1–4. [Online]. Available: https://openreview.net/forum?id=B1-q5Pqxl

[5] M. Hu, Y. Peng, F. Wei, Z. Huang, D. Li, N. Yang, and M. Zhou, “Attention-guided answer distillation for machine reading comprehension,” in Proc. Conf. Empirical Methods Natural Lang. Process., Brussels, Belgium, 2018, pp. 2077–2086, doi: 10.18653/v1/d18-1232.

[6] Y. Jiang, N. Joshi, Y.-C. Chen, and M. Bansal, “Explore, propose, and assemble: An interpretable model for multi-hop reading comprehension,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Florence, Italy, 2019, pp. 2714–2725, doi: 10.18653/v1/p19-1261.

[7] M. Tu, K. Huang, G. Wang, J. Huang, X. He, and B. Zhou, “Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents,” 2019, arXiv:1911.00484. [Online]. Available: http://arxiv.org/abs/1911.00484

[8] L. Song, Z. Wang, M. Yu, Y. Zhang, R. Florian, and D. Gildea, “Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks,” 2018, arXiv:1809.02040. [Online]. Available: http://arxiv.org/abs/1809.02040

[9] M. Tu, G. Wang, J. Huang, Y. Tang, X. He, and B. Zhou, “Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Florence, Italy, 2019, pp. 2704–2713, doi: 10.18653/v1/p19-1260.

[10] S. Kundu, T. Khot, A. Sahbaniwal, and P. Clark, “Exploiting explicit paths for multi-hop reading comprehension,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Florence, Italy, 2019, pp. 2737–2747, doi: 10.18653/v1/p19-1263.

[11] V. Zhong, C. Xiong, N. S. Keskar, and R. Socher, “Coarse-grain fine-grain coattention network for multi-evidence question answering,” in Proc. 7th Int. Conf. Learn. Represent. (ICLR), New Orleans, LA, USA, May 2019, pp. 1–4. [Online]. Available: https://openreview.net/forum?id=Syl17OoRqK7
[12] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning, “HotpotQA: A dataset for diverse, explainable multi-hop question answering,” in Proc. Conf. Empirical Methods Natural Lang. Process., Brussels, Belgium, 2018, pp. 2369–2380, doi: 10.18653/v1/d18-1419.

[13] M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer, “TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension,” in Proc. 55th Annu. Meeting Assoc. for Comput. Linguistics, Vancouver, BC, Canada, Jul. 2017, pp. 1601–1611, doi: 10.18653/v1/p17-1147.

[14] Y. Cui, T. Liu, W. Che, L. Xiao, Z. Chen, W. Ma, S. Wang, and G. Hu, “A span-extraction database for Chinese machine reading comprehension,” in Proc. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP), Hong Kong, 2019, pp. 5882–5888, doi: 10.18653/v1/d19-1060.

[15] J. Welbl, P. Stenetorp, and S. Riedel, “Constructing datasets for multi-hop reading comprehension across documents,” Trans. Assoc. Comput. Linguistics, vol. 6, pp. 287–302, Dec. 2018. [Online]. Available: https://transacl.org/ojs/index.php/tacl/article/view/1325

[16] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. H. Hovy, “RACE: Large-scale reading comprehension dataset from examinations,” in Proc. 2017 Conf. Empirical Methods Natural Lang. Process., Copenhagen, Denmark, Sep. 2017, pp. 785–794, doi: 10.18653/v1/d17-1082.

[17] T. Onishi, H. Wang, M. Bansal, K. Gimpel, and D. A. McAllester, “Who did what: A large-scale person-centered cloze dataset,” in Proc. 2016 Conf. Empirical Methods Natural Lang. Process., Austin, TX, USA, Nov. 2016, pp. 2230–2235, doi: 10.18653/v1/d16-1241.

[18] F. D. A. Bordes, S. Chopra, and I. Weston, “The goldilocks principle: Reading Children’s books with explicit memory representations,” 2015, arXiv:1511.02301. [Online]. Available: http://arxiv.org/abs/1511.02301

[19] C. Wang and H. Jiang, “Explicit utilization of general knowledge in machine reading comprehension,” in Proc. 57th Conf. Assoc. for Comput. Linguistics, Florence, Italy, Jul./Aug. 2019, pp. 2263–2272, doi: 10.18653/v1/a19-1219.

[20] M. Ding, C. Zhou, Q. Chen, H. Yang, and J. Tang, “Cognitive graph for multi-hop reading comprehension at scale,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Florence, Italy, 2019, pp. 2694–2703, doi: 10.18653/v1/a19-1259.

[21] Y. Cao, M. Fang, and D. Tao, “BAG: Bi-directional attention entity graph convolutional network for multi-hop reasoning question answering,” in Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics, Minneapolis, MN, USA, Jun. 2019, pp. 357–362, doi: 10.18653/v1/n19-1032.

[22] P. W. Battaglia et al., “Relational inductive biases, deep learning, and graph networks,” 2018, arXiv:1806.01261. [Online]. Available: http://arxiv.org/abs/1806.01261

[23] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), Doha, Qatar, 2014, pp. 1532–1543, doi: 10.3115/v1/d14-1162.

[24] K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher, “Dropout: A simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014, [Online]. Available: http://dl.acm.org/citation.cfm?id=2670313.

[25] P. W. Battaglia, et al. “Interaction networks,” 2016, arXiv:1605.00438. [Online]. Available: http://arxiv.org/abs/1605.00438

[26] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Int. Conf. Artif. Intell. Statist., Fort Lauderdale, Florida, Apr. 2017, pp. 315–323. [Online]. Available: http://proceedings.mlr.press/v15/glorot14a/glorot14a.pdf

[28] S. Bird, “NLTK: The natural language toolkit,” in Proc. 44th Annu. Meeting Assoc. Comput. Linguistics, Sydney, NSW, Australia, Jul. 2006, pp. 17–21. [Online]. Available: https://www.aclweb.org/anthology/P06-4018

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014, [Online]. Available: http://dl.acm.org/citation.cfm?id=2670313.

[30] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980

[31] B. Dhingra, Q. Jin, Z. Yang, W. Cohen, and R. Salakhutdinov, “Neural models for reasoning over multiple mentions using coreference,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., New Orleans, LA, USA, 2018, pp. 42–48, doi: 10.18653/v1/n18-2007.

[32] N. De Cao, W. Aziz, and I. Titov, “Question answering by reasoning across documents with graph convolutional networks,” in Proc. Conf. North, Minneapolis, MN, USA, 2019, pp. 2306–2317, doi: 10.18653/v1/n19-1240.

[33] Y. Zhuang and H. Wang, “Token-level dynamic self-attention network for multi-passage reading comprehension,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Florence, Italy, 2019, pp. 2252–2262, doi: 10.18653/v1/p19-1218.

[34] M. E. Peters, M. Neumann, M. Iyyer, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., New Orleans, LA, USA vol. 1, 2018, pp. 2227–2237.

[35] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, 2019, pp. 4171–4186, doi: 10.18653/v1/n19-1423.

[36] D. Weissborn, G. Wiese, and L. Seifke, “Making neural QA as simple as possible but not simpler,” in Proc. 21st Conf. Comput. Natu- ral Lang. Learn. (CoNLL), Vancouver, BC, Canada, 2017, pp. 271–280, doi: 10.18653/v1/n17-1028.

[37] K. Nishida, I. Saito, K. Nishida, K. Shinoda, A. Otsuka, H. Asano, and J. Tomita, “Multi-style generative reading comprehension,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Florence, Italy, 2019, pp. 2273–2284, doi: 10.18653/v1/p19-1220.

LIJUN HUO received the bachelor’s degree from the National University of Defense Technology (NUDT), China, in 2018, where he is currently pursuing the master’s degree with the Science and Technology on Information Systems Engineering Laboratory. His research interests include natural language processing and machine reading comprehension.

XIANG ZHAO (Member, IEEE) was born in 1986. He received the Ph.D. degree from the University of New South Wales, Australia, in 2014. He is currently an Assistant Professor with the National University of Defense Technology, China. He has published several refereed papers in international conferences and journals, such as SIGMOD, SIGIR, VLDB Journal, and TKDE. His research interests include natural language processing and knowledge graph construction.