Memory-Aware Attentive Control for Community Question Answering With Knowledge-Based Dual Refinement

DOI:
10.1109/TSMC.2023.3234297

Document Version
Accepted author manuscript

Memory-aware Attentive Control for Community Question Answering with Knowledge-based Dual Refinement

Jinmeng Wu, Tingting Mu, *Member, IEEE,* Jeyarajan Thiyagalingam, John Y. Goulermas, *Senior Member, IEEE*

Abstract—Question answering system in open domain enables a machine to automatically select and generate the answer for questions posed by humans in a natural language form on the website. Previous approaches seek effective ways of extracting the semantic features between question and answer, but the contextual information effects in semantic matching is still limited by short-term memory. As an alternative, we propose an internal knowledge-based end-to-end model, enhanced by an attentive memory network for both answer selection and answer generation tasks by considering the full advantages of the semantics and multi-facts (ie. timescales, topics and context). In detail, we design a long-term memory to learn the top-k fine-grained similarity representations, where two memory-aware mechanisms aggregate the series of semantic word-level and sentence-level similarities to support the coarse contextual information. Furthermore, we propose a novel memory refinement mechanism with the two-dimensional of writing heads that offer an efficient approach to multi-view selection of the salient word pairs. In the training stage, we adopt the transformer-based transfer learning skill to effectively pre-train the model. Experimentally, we compare the state-of-the-art approaches on four public datasets, the experimental results show the proposed model achieves competitive performance.

Index Terms—Information retrieval, memory architecture, knowledge based systems, distributed memories, attention mechanism.

1 INTRODUCTION

LONG-TERM semantic matching is one of the major challenges in the question answering (QA) system. The task is to automatically answer the question based on a full understanding of the human-level language combined with context [1]. Two question-answer examples are shown in Fig.1 over a span of text in context, each with three candidates in the answer pool. Regarding the object "cat" in two questions, the keywords "Garden" and "Sandra" in the context respectively indicate the location, and personal information related to "where" and "who" in the questions, these keywords are necessary to judge the correct answer. However, similar keywords in the context may determine the answer, such as "bathroom", and "mat". The semantic matching is not robust based on independent keywords.

Although recurrent neural network (RNN) models encode the relationship between adjacent words in a sentence, the memory capacity is limited by the length of context. Recently, the pre-trained transformer-based models [2], [3] derived by neural language models (e.g., BERT [4], ELMo [5] and RoBERTa [6]) achieve the impressive matching accuracy results. Specifically, the works [2], [3] build the transformer-based model to learn the semantic dependencies between words and sentences on a large dataset. However, they focus on activating specific masked words but ignore the effect of background information, which can enrich the interactive and global information in QA system.

Context: C

- C: Katie saw that Jane took the cat before. Jane walked to the bathroom with the cat. Then, Katie asked Jane to put the cat on the mat, and went to the kitchen with her. After that, Jane gave the cat to Sandra. Sandra went to the garden, and took the cat there.

Question Q_1

- Where is the cat?

Candidate Answers: A_1, A_2 and A_3

- A_{11}: bathroom
- A_{12}: mat
- A_{13}: **garden**

Question Q_2

- Who with the cat?

Candidate Answers: A_1, A_2 and A_3

- A_{21}: Jane with the cat in the kitchen before.
- A_{22}: Jane with the cat now.
- A_{23}: **Sandra with the cat in the garden.**

Fig. 1. Example scenario for question answering based on long-term matching. The groundtruth answers are marked in bold.

Typically, memory networks [7], [8] generally adopt long-term memory structure to capture the semantic dependencies from sequential data. The memory-based neural networks [7], [9] recently have shown promising improvement in QA matching. The advantage of the existing net-

J. Wu is with the Department of Electronical and Information Engineering, HuBei Key Laboratory of Optical Information and Pattern Recognition, Wuhan Institute of Technology, Wuhan 430205, China. Email: jinmeng2004910@outlook.com.
T. Mu is with the School of Computer Science, The University of Manchester, Kilburn Building, Oxford Road, Manchester, UK, M13 9PL. Email: tingting.mu@manchester.ac.uk.
J. Thiyagalingam is with the Science and Technologies Facilities Council, Rutherford Appleton Laboratory, Harwell Campus, Oxon, UK. Email:t.jeyan@stfc.ac.uk
J. Y. Goulermas is with the Department of Computer Science, The University of Liverpool, Brownlow Hill, Liverpool, UK, L69 3GJ. Email: j.y.goulermas@liverpool.ac.uk.

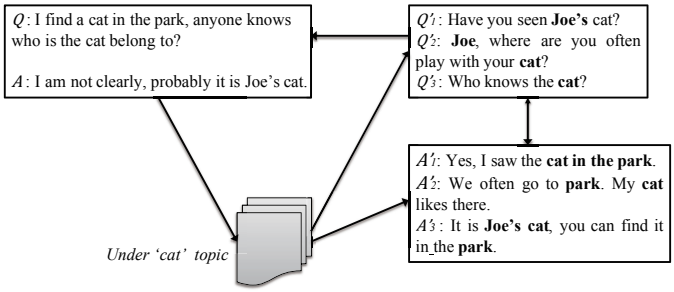


Fig. 2. The complementary QA structure under the topic "cat". The keywords are marked in bold.

works with external memory units is that the memorization performance varies over time scales and the memory cell has a certain data throughput, so as to realize the functions of content update, storage and forgetting. However, the current state-of-the-art works are trapped in two aspects for community semantic matching: 1) The fixed refinement mechanism is not adaptive to record the interactive components. The previous memory-based networks [9], [10] either initialize a random memory vector to record a small amount of contextual information (e.g., to the end of word portions "...bathroom with the cat") or use a large-size memory matrix to encode entire sentences for each iteration process. 2) Limited size of the memory cell causes less external knowledge related to the question and the knowledge is compressed during propagation.

To analyze the semantic interaction with the effective knowledge base of the memory network is a challenge for QA matching. The proposed method designs the dynamic memory network to focus on addressing the above problems with a novel memory refinement, it benefits to analyze the tokens matching process and strengthen data storage capability. To address the problem of the fixed refinement mechanism and the size limitation of the memory cell, the model designing a dynamic memory network that also 1) learns the similarity representation between positional words (e.g.,"in hand" and "hand in") in the sequence, 2) memory the prior long-term semantic knowledge, 3) extracts the vital similarity elements from the memory system.

In most cases, the answer selection task without insufficient context would result in low matching accuracy. For instance, the short question Q in Fig.2 only has a few words. It is difficult to infer the semantic relationship with a small number of word pairs, such as "cat in park" and "Joe's cat". To explore the mutually supportive sentences of the QA pair, we design a topic approach to mining the relevant sentence pairs for enriching the content in memory, so the answer A is judged depending on the keywords in related Q'-A' pairs under the topic. Different from the existing work [9] of the randomly initialized memory module, the proposed model uses a correlation similarity matrix to act on the memory initialization process by collecting a certain number of top-level related question-answer pairs in the same category from the given corpus.

We address the aspects that have been highlighted above, particularly on the refinement mechanism and initialization method of the memory network, and propose a novel memory network approach to enrich the contextual memories in semantic matching. We aim to model a dynamic attention memory network by studying the effectiveness of the hierarchical refinement mechanism. In particular, we make the following key contributions:

- We propose a distributed memory-based system suitable for question answering, in which a dual refinement mechanism adopts a step-wise method, that is, the semantic similarities of the question and answer are selected separately and iteratively, which can better emphasize the impact of word-level semantic matching on the model.
- The paper seeks a new internal knowledge-based mining approach for the supplementary QA system. A three-dimensional quantity in the knowledge-based source for memory initialization, called memory pool is proposed, it composes of high-ranking relevant sentences under the topic from the database.
- The memory-aware attention mechanism is represented by a matrix that constructs an interactive aggregation with multiple inputs, which can be adapted to memory dual refinement.
- We perform the empirical comparisons of the proposed model and various state-of-the-art works using four public datasets for multi-tasks.

Overall, the proposed memory network is designed to learn word-level semantic similarity bilaterally for robust interactive learning in sequential behaviors. Specially, we provide a new memory module initialization method, which is beneficial to complement the lack of information in the short sentences of community data.

2 Related Work

2.1 Deep Neural Networks in QA

Deep learning models have been proven to be effective for generating distributed embedding representations of text objects (e.g., words, phrases and sentences) and characterizing the latent relationships between them. The multiple hidden representations returned by a bi-directional long short-term memory (LSTM) at different states are used to compute a similarity matrix between the question and answer sentences [11]. By characterizing a sentence as a set of word embedding vectors stored in a sentence matrix [12], a CNN is typically employed to compute a vector representation for the sentence from its input matrix [13]. The convolution feature map enables the modeling of the semantic information between words within the same sentence. Overall, the above techniques formulate sentence representation without consideration of the contextual information between sentence pairs.

Another approaches work on learning the deep contextualized word representations, where the self-training bidirectional language model called ELMo [5], the pre-training of deep bidirectional transformer derived by BERT [4]. The variant models of BERT are widely applied to answer selection [2], [3] and machine reading comprehension [14]. To examine the performance of pre-training strategy, [6] proposes RoBERTa model that focuses on optimizing the hyper-parameters of BERT method. Moreover, the BERT-based transfer learning technique is adapted to enhance the amount of QA training datasets to achieve impressive matching accuracy results [3].

To enrich the textual information in the corpus, knowledge-based models as the external sources to be able to provide the information in question answering [15]. The word co-occurrence information obtained from large text corpora (e.g., Wikipedia, newswire) or the hierarchy information drawn from semantic networks (e.g., Wordnet) can be utilized to formulate semantic similarity between words.Recent works [16], [17] leverage the external knowledge to explore the relational reasoning for answer retrieval, a question over knowledge base normally consists of a subject entity and a single relation.

2.2 Attention-based Models

The attention mechanism was first proposed in the machine translation task [18], and has been widely used in various fields [19], [20]. The use of the attention technique is to enable a neural network to identify the salient components of a sentence, it tends to rely on a weighted sum of a set of component representations, where the attention weights control the contributions of the compositional components. Different ways of designing attention mechanisms correspond to different strategies of defining the components and formulating their importance scores.

A common way of incorporating an attention mechanism in an RNN- or LSTM-based QA system, is to relate the different components to the different hidden states of the network, which correspond to the different word positions in a sentence. An alternative way to set the cross attention mechanism is to examine the importance of the word pairs that appear in the given sentence pair. The importance score of each word pair can be computed from their corresponding word embeddings [19] or the hidden representations at the corresponding word positions returned by an LSTM [21], through the use of Euclidean distance or dot product. [11] measures the semantic interactions of word pairs from the similarity matrix between the encoded sentence representations, which come from bi-directional LSTM.

The self-attention has emerged as an attention mechanism aimed at aligning the multiple positions of a sequence, which has been widely used in recent transformer-based works [2], [7]. The multi-heads self attention mechanism of the transformer concatenates multiple result representations of the three linear transformation matrices of the input sentence [4]. Variations of attention mechanisms are developed in a bespoke manner to suit a specific task, for instance, by joining the context into a given question using co-attention attention in machine reading comprehension task [21], etc.

2.3 Memory-augmented Networks

A classic memory network is made of an array of cells, which records part of the mapped input feature representation, and outputs the new required data through a long-term update of its mechanism. General neural network memory models such as Neural Turing Machine (NTM) consists of the differentiable memory and controller that reads and writes to specific locations [22]. Memory network has been employed for QA task with internal resources, such as supporting contexts or facts, in most cases. Whereas in machine comprehension answers are inferred from a given

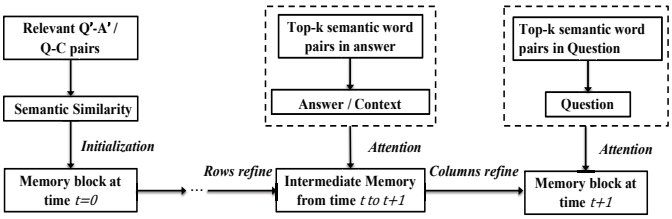


Fig. 3. An illustration of the proposed memory-based QA system.

text, the knowledge base of external domain is typically required to answer questions, this external knowledge consumes the cost of manual feature engineering. In fact, 93% of the questions pruned as Freebase could not answered [23], [24]. A Memory network called MemNN has been proposed in [10], first introducing the concept of a long-term memory component for QA. As a strongly supervised model, MemNN generates a single word to answer the question relying on the supporting facts.

A series of attentive memory networks [2], [7] are recently proposed to strengthen memory by storing the relevant information. Dynamic Memory network (DMN) [25] improves over it by employing an end-to-end trainable network with an attention mechanism. The memory iteratively produces a vector to store the relevant input information, which is used in answer generation. End-to-end memory networks (MemN2N) [26] encodes sentence to a continuous vector representation depending on recurrent attention mechanism instead of sequence aligned recurrence, it has been shown to perform well on simple-language machine comprehension and language modeling tasks. Different from the MemNN and DMN architectures, MemN2N uses an end-to-end mechanism with weakly supervised training. However, the disadvantage of MemN2N is that it only generates answers with one single word. Multi-layer Embedding with Memory Network (MEMEN) [27] provides a hierarchical attentive memory to learn an alignment memory matrix, which contains the syntactic and semantic information of words returned by the skip-gram model.

Recent works [9], [28] focus on optimizing the memory network by changing the refinement technique, they adopt the reinforcement learning agent to update the memory vector in order to compress the written content and remove insufficient information, but this cannot guarantee the integrity of the recorded content, and the scalability of the memory module is still worthy of improvement.

3 METHODOLOGY

The proposed system aims to solve the two specific QA tasks: (1) answer selection (AS) targets on ranking the answer from a pool of candidate sentences, and (2) machine reading comprehension (MRC) generates a span of answer text according to the context. To summarize, we design a memory network based long-term semantic matching system referred to as MMN illustrated in Fig. 3.

3.1 AS Task Overview

Given a question $q = \{w_i^q\}_{i=1}^m$, and an answer candidate $a = \{w_j^a\}_{j=1}^n$, it is reasonable to assume that the answer's relevance depends on the semantic similarity between the

words they contain. A distributed vector representation is employed to model the semantic similarity between words—each word is represented by d-dimensional vector $\boldsymbol{w} = [w_1, w_2, \ldots, w_d]$. Defining the integers m and n as the maximal lengths of a question and an answer, variable length sentences can then be characterized by fixed-size matrices by adding zero rows to fill up empty positions for shorter sentences: the $m \times d$ matrix $\mathbf{X}$ denotes a question and the $n \times d$ matrix $\mathbf{Y}$ denotes an answer candidate.

3.1.1 Mnemonic Matching

To describe the semantic matching dependencies along the time, the element-wise multiplication function is used to aggregate the memory matrix $\mathbf{M}_T$ and the similarity matrix $\mathbf{S}$, given as

$$\mathbf{F} = \mathbf{A}(\mathbf{M}_T, \mathbf{S}) = \mathbf{M}_T \odot \mathbf{S}, \tag{1}$$

where the $\mathbf{M}_T$ and $\mathbf{S}$ are $m \times n$ metric. The similarity matrix $\mathbf{S}$ builds the semantic interaction between the question and answer representations. The memory network produces the episodic memory matrix $\mathbf{M}_t$ at current time t based on the contextual representations and memory matrix at previous time $t-1$. Assuming that the number of iterations ('hops') of the memory network is T times ($0 \leq t \leq T$), the final episodic memory $\mathbf{M}_T$ is able to include significant information required to semantic matching between question and answer. The details of refined memory network will be introduced in the section 3.4.

3.1.2 Bi-linear Similarity Construction

Working with the matrix representations of sentences, the question has m words that is represented as a $m \times d$ matrix $\mathbf{X}$, likewise given the answer has n words it is represented by an $n \times d$ matrix $\mathbf{Y}$, where the rows of each matrix correspond to the vector representations of the words appearing in the sentence. To explore the original word-level relationship, we use a type of bi-linear model [29] with restricted parameters to formulate the semantic relatedness function of a sentence pair (q, a), given as

$$\mathbf{S} = f\left(\mathbf{X}\mathbf{p}_1^{\mathrm{T}}\mathbf{p}_2\mathbf{Y}^{\mathrm{T}} + \mathbf{b}_s\right), \tag{2}$$

where weights $\mathbf{p}_1$ and $\mathbf{p}_2$ are two d-dimensional column vectors, and the $m \times n$ matrix $\mathbf{b}_s$ are network variables to be optimized, and $f(\cdot)$ is a hyperbolic tangent function [30] that operates on each element of the input matrix.

3.1.3 Memory Pools Pre-processing

The relevant sentences of the corpus have the assistant information for the similarity between the original question and answer. In the primary step, we collect a number of the relevant question-question (q', q) and relevant answer-answer (a', a) pairs that have the top matching scores.

Assuming the embedding representations of relevant question and answer are set as matrices $\mathbf{X}'$, $\mathbf{Y}'$. During the pre-processing step, we convert the sentence representation to a vector by normalizing the columns of the matrix of each word in the sentence in terms of l_2-norm function:

$$z_l = ||\mathbf{Z}^{(l)}|| = \left(\sum_{i=1}^{k} \left(\mathbf{Z}_i^{(l)}\right)^2\right)^{\frac{1}{2}}; \forall l \in [1, 2, \ldots, d], \tag{3}$$

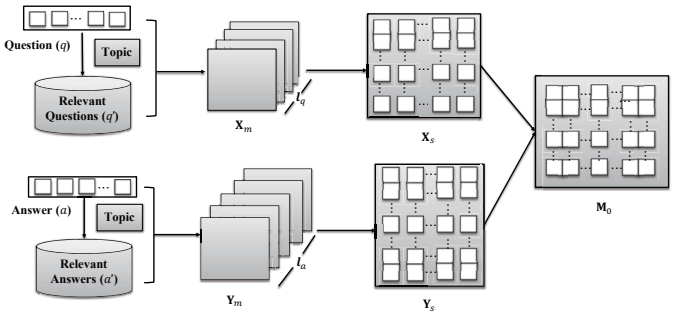


Fig. 4. An illustration of memory initialization for answer selection.

where k denotes the column size of the matrix. The normalized question representation $\mathbf{x}$, the relevant question representation $\mathbf{x}'$, the answer sentence representation $\mathbf{y}$ and the relevant answer representation $\mathbf{y}'$ are d-dimensional vectors.

With regards to the choice of the transformation function, we adopt the l_2-norm function to explore the embedding semantic relationship between elements of different words. The other common methods used in option are to set weighted averaged function [31], and a bi-linear function. Subsequently, we compare and discuss these three methods in section 5.2.

In order to search for the most relevant (q', a') pairs, we pick a number of questions and answers from the dataset corpus under the same category with the input (q, a) pair. We employ the cosine similarity function is formulated by $cos[\mathbf{z}, \mathbf{z}'] = \frac{\mathbf{z}^T\mathbf{z}'}{||\mathbf{z}||||\mathbf{z}'||}$ to compute the similarity scores $cos[\mathbf{x}, \mathbf{x}'_u]$ between question and its relevant question and $cos[\mathbf{y}, \mathbf{y}'_v]$ between answer and its relevant answer, where u, v denote the number of related questions and answers. A certain number of relevant (q', a') pairs are collected into memory pools. We rank a top number l_q, l_a of relevant questions and answers depending on the magnitudes of similarity scores. Then we feed these specific relevant question and answer embeddings into three-dimensional memory pools $\mathbf{X}_m$, $\mathbf{Y}_m$, where the entire number of sentences in the memory pools is l_q, l_a, separately.

3.1.4 Memory Network Initialization

A series of relevant question and answer pairs contain enriched information that relates to the original sentence pairs. The architecture of initializing the proposed memory network is illustrated in Figure 4. Assuming a distributed memory representation $\mathbf{M}_0$ set as a $m \times n$ matrix at time $t = 0$. We use the memory pools of relevant question and answer sentences $\mathbf{X}_m$, $\mathbf{Y}_m$ to initialize the memory.

Memory pools store a number of related questions and answers. Prior to initializing memory matrix, we individually integrate the related l_q questions and l_a answers from the memory pools. The two-dimensional matrices of memory slots are defined by the weighted sum functions:

$$\mathbf{X}_s = \sum_{i}^{l_q} \mathbf{V}_i \mathbf{X}_m^{(i)}, \tag{4}$$

with

$$\mathbf{Y}_s = \sum_{j}^{l_a} \mathbf{U}_j \mathbf{Y}_m^{(j)}, \tag{5}$$

where $\mathbf{V}_i$ denotes $m \times m$ matrix, and $\mathbf{U}_j$ is $n \times n$ matrix. $\mathbf{X}_m^{(i)}$ denotes the i-th element $m \times d$ dimensional matrix of the memory pool $\mathbf{X}_m$. $\mathbf{Y}_m^{(j)}$ denotes the j-th element $n \times d$ dimensional matrix of the memory pool $\mathbf{Y}_m$. In the initialization process, two memory slots $\mathbf{X}_s$, $\mathbf{Y}_s$ are used in Eq. (2) as the inputs. The similarity output is equal to the initialized memory $\mathbf{M}_0$, it contains interactive information between the relevant question and answer pairs. In the subsequent section 3.4, we use the initialized memory matrix to refine and update the memory matrix.

3.2 MRC Task Overview

Given a question $q = \{w_i^q\}_i^m$ and context $c = \{w_j^c\}_j^n$ pair, where the m, n is the number of words in question, and context. In general, $n \geq m$. Suppose the distributed context representation $\mathbf{C}$ is denoted as $n \times d$ matrix. The goal of this task is to predict an answer that is constrained as a segmented text of context. Subsequently, we set the question and context pair (q, c) as inputs into the proposed model.

3.2.1 Pointer Matching in MRC task

In the matching layer, the model explores a segment of sequence spans of the context to answer the question. A pointer networks [32] is a popular positional decoding approach to predict the start and end position of the answer. We follow the approach in [33] to compute the start and end distribution of words in context using a bi-linear semantic matching between the context and final episodic memory. The probability distribution of the start index is defined as

$$\mathbf{p}_\mathrm{s} = softmax\left((\mathbf{C}\mathbf{w}_s)^\mathrm{T}\mathbf{M}_T\right), \tag{6}$$

with the probability distribution of the end index as

$$\mathbf{p}_\mathrm{e} = softmax\left((\mathbf{C}\mathbf{w}_e)^\mathrm{T}\mathbf{M}_T\right), \tag{7}$$

where the trainable weights $\mathbf{w}_s$, $\mathbf{w}_e$ both are d-dimensional row vector. The notation $softmax(\cdot)$ represents the softmax function shown in Eq.(13). The n-dimensional distributed outputs $\mathbf{p}_\mathrm{s}$, $\mathbf{p}_\mathrm{e}$ are calculated by the matching functions.

3.2.2 Semantic Memory Network Initialization

Regards the similarity matrix $\mathbf{S}$, we apply the question, context embeddings $\mathbf{X}$, $\mathbf{C}$ to the bi-linear function Eq. (2). In particular, to initialize the memory network, we replace the memory pools module due to the context contains a number of sequences related to the question, and the length of the context is much longer than the one of a candidate answer. Thus, we directly set the initialized matrix $\mathbf{M}_0$ equals to $\mathbf{S}$. Subsequently, the initialized matrix is the input into the memory refinement mechanism.

3.3 Top-k Max Pooling

To aggregate significant information and to reduce the size of the similarity representation, we use a pooling process to select the number of top-ranked word pairs in the similarity matrix $\mathbf{S}$ in row and column directions individually, corresponding to word pairs importance between the question and answer. The pooling function focuses on each column of the similarity matrix that is defined as

$$\mathbf{P}^{(q)} = \text{top-}k \text{ max pooling}\left(\mathbf{S}[:, j]\right); \forall j \in [1, 2, \ldots, n], \tag{8}$$

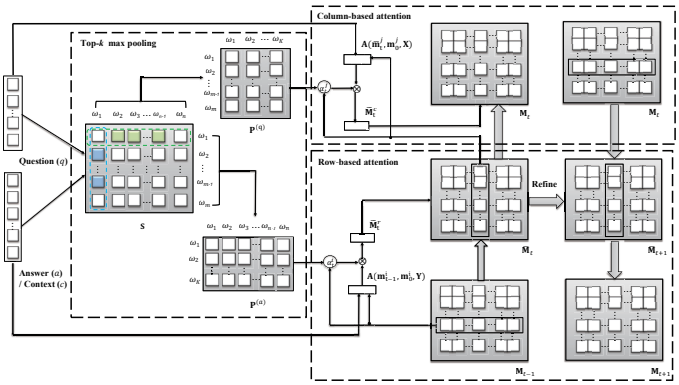


Fig. 5. An illustration of memory refinement for QA system.

with the pooling in each row of the similarity matrix:

$$\mathbf{P}^{(a)} = \text{top-}k \text{ max pooling}\left(\mathbf{S}[i, :]\right), \forall i \in [1, 2, \ldots, m], \tag{9}$$

where $\mathbf{P}^{(q)}$ is $m \times k$ matrix and $\mathbf{P}^{(a)}$ is $k \times n$ matrix. The $\mathbf{S}[:, j]$ is a m dimensional vector, and the $\mathbf{S}[i, :]$ is a n dimensional vector. The pooling operation compares the semantic similarities between a word in the sentence and all the words in the other corresponding sentence and returns an aggregated similarity measure for that word. This results in the m-th element and n-th element of the similarity matrix $\mathbf{P}^{(q)}$, $\mathbf{P}^{(a)}$ for the question and answer, respectively. The top-ranked number k can be arbitrarily set to different values for questions and answers. In general, the value k of answer is bigger than the one of question, since the answer contains more words.

3.4 Memory-aware Attentive Refinement

In this section, we build the semantic memory refinement structure. A $m \times n$ dimensional memory matrix $\mathbf{M}_t$ is set at the current time t in order to store not only the local input information but also the relevant questions and answers. We design two writing attention heads to extract the salient word-level similarities from the answer and question, respectively. Memory refinement is a continuous mechanism, where the memory matrix renews its column elements once the rows are updated. The memory refinement structure of the proposed model is illustrated in Fig.5.

Assuming the $m \times n$ memory matrix at current episode t denotes $\mathbf{M}_t$. To simplify the notations convenient for computation, we define the row vector $\mathbf{M}_t[i, :] = \mathbf{m}_t^i$ and the column vector $\mathbf{M}_t[:, j] = \mathbf{m}_t^j$ at the t-th time.

3.4.1 Row-based Memory

Each row vector in the previous episode memory matrix is represented as $\mathbf{m}_{t-1}^i$, the activation function Rectified Linear Unit ($ReLU$) [34] is applied to convert the element to be sparse, we update the memory by using the following defined function:

$$\bar{\mathbf{M}}_t = ReLU(\mathbf{W}_r\tilde{\mathbf{M}}_t^r + \mathbf{b}_r), \tag{10}$$

where the $\mathbf{W}_r$ denotes a $m \times 4$ dimensional matrix, $\mathbf{b}_r$ is $m \times n$ dimensional matrix. Soft attention as given by a contextual matrix through a weighted summation of vectors

$\mathbf{A}(\mathbf{m}_{t-1}^i, \mathbf{m}_0^i, \mathbf{Y})$ and attention weight α_t^i. The contextual memory is computed by the attention function, given as

$$\tilde{\mathbf{M}}_t^r = \sum_{i=1}^{m} \alpha_t^i \mathbf{A}(\mathbf{m}_{t-1}^i, \mathbf{m}_0^i, \mathbf{Y}), \tag{11}$$

where the contextual memory matrix $\tilde{\mathbf{M}}_t^r$ represents a $4 \times n$ matrix. The attention mechanism is responsible for generating the contextual memory matrix $\tilde{\mathbf{M}}_t^r$ based on the previous episode memory vector $\mathbf{m}_{t-1}^i$, the initialized memory vector $\mathbf{m}_0^i$ and the answer sentence representation $\mathbf{Y}$. The attention function $\mathbf{A}(\mathbf{m}_{t-1}^i, \mathbf{m}_0^i, \mathbf{Y})$ for each row vector of memory is defined as

$$\mathbf{A}(\mathbf{m}_{t-1}^i, \mathbf{m}_0^i, \mathbf{Y}) = \begin{bmatrix} \mathbf{m}_{t-1}^i \odot \mathbf{Y}\mathbf{w}_y \\ |\mathbf{m}_{t-1}^i - \mathbf{Y}\mathbf{w}_y| \\ \mathbf{m}_{t-1}^i \odot \mathbf{m}_0^i \\ |\mathbf{m}_{t-1}^i - \mathbf{m}_0^i| \end{bmatrix}, \tag{12}$$

where $\mathbf{w}_y$ is a d-dimensional row vector. The symbol $\odot$ is an element-wise product. The symbol $|\cdot|$ is defined as the element-wise absolute value. The feature set $\mathbf{A}(\mathbf{m}_{t-1}^i, \mathbf{m}_0^i, \mathbf{Y})$ aggregates four vectors of n elements to a $4 \times n$ matrix. The feature vector captures a variety of similarities between input sentence and memory [35]. The feature set $\mathbf{A}(\mathbf{m}_{t-1}^i, \mathbf{m}_0^i, \mathbf{Y})$ is composed of four different similarity vectors between answer sentence representation, memory and initialized memory. The similarity between related question and answer sentence representations is used as an initialized memory to generate a feature vector instead of a single input sentence. The attention weight is computed depending on the softmax function

$$\alpha_t^i = \frac{exp(z_t^i)}{\sum_{l=1}^{m} exp(z_t^l)}. \tag{13}$$

The attention weight is computed by the relational value z_t^i, which depends on the previous memory vector and the pooled similarity matrix of answer. The relational function is defined as

$$z_t^i = f((\mathbf{m}_{t-1}^i + \mathbf{w}_{p_1}^t \mathbf{P}^{(a)})\mathbf{w}_a + b_a), \tag{14}$$

where the weight $\mathbf{w}_a$ denotes a n-dimensional row vector, $\mathbf{w}_{p_1}^t$ is a k-dimensional column vector at the t-th time, and bias b_a is a scalar.

3.4.2 Column-based Memory

After updating the rows of the episode memory matrix, we adopt the same refinement mechanism in the section 3.4.1 to refine the columns of memory matrix $\bar{\mathbf{M}}_t^c$ based on the updated memory matrix $\tilde{\mathbf{M}}_t^r$, where the parameter represents $\theta_c = \{\mathbf{w}_{p_2}^t, \mathbf{w}_q, b_q\}$. Subsequently, the memory matrix at the current time t is computed by

$$\mathbf{M}_t = ReLU(\tilde{\mathbf{M}}_t^c \mathbf{W}_c + \mathbf{b}_c), \tag{15}$$

where the weight $\mathbf{W}_c$ denotes a $4 \times n$ dimensional matrix, $\mathbf{b}_c$ is $m \times n$ matrix. Next, we follow the same steps as updating the row-based memory, but with different inputs. The previous memory $\bar{\mathbf{m}}_t^j$, initial memory vector $\mathbf{m}_0^j$ and question sentence representation are used to compute the column-based contextual memory $\tilde{\mathbf{M}}_t^c$ as the inputs in Eq.(11).

3.5 Model Training and Initialization

3.5.1 Prediction Layer

To aggregate significant information of the similarity representation, we apply a pooling process to the computed similarity matrix $\mathbf{S}$. The pooling function returns a similarity vector that contains the most important pairs between question and answer, is defined as

$$\mathbf{s}^{(l)} = \text{max-pooling}\left(\mathbf{F}^{(l1)}, \mathbf{F}^{(l2)}, \cdots, \mathbf{F}^{(ln)}\right), \tag{16}$$

where $\mathbf{s}^{(l)}$ denotes the l-th element, $\forall l \in [1, 2, \ldots, m]$ of the similarity vector $\mathbf{s}$, and $\mathbf{F}^{(lk)}$ denotes the lk-th element of the full $m \times n$ matrix $\mathbf{F}$. The max-pooling operation compares the semantic similarities between a word in the question sentence and all the words in the answer candidate, and returns an aggregated similarity measure for that word. This results in a length-m similarity vector $\mathbf{s}$ for each question.

Given the Q-A pair in answer selection task, the probability that an answer candidate a is related to q can be modeled using two-way softmax based on the encoding of the similarity representation $\mathbf{h}$, given as

$$p(t = 1|\mathbf{s}) = \frac{\exp\left(\mathbf{s}^{\mathrm{T}}\boldsymbol{\alpha}_1\right)}{\exp\left(\mathbf{s}^{\mathrm{T}}\boldsymbol{\alpha}_0\right) + \exp\left(\mathbf{s}^{\mathrm{T}}\boldsymbol{\alpha}_1\right)}, \tag{17}$$

where the two column vectors $\boldsymbol{\alpha}_0$ and $\boldsymbol{\alpha}_1$ are softmax parameters with the same dimensionality as the similarity vector $\mathbf{s}$. The matching prediction task is formulated as a binary classification problem.

Given a collection of question and answer candidate sentences with available ground truth knowledge of whether they are related, the traditional training approach optimizes the model variables by minimizing the regularized cross-entropy cost function as shown below

$$\begin{aligned} L_{as}(\boldsymbol{\theta}) = -\sum_{(i,j)\in I} & [t_{ij} \log p(t_{ij} = k|\mathbf{s}_{ij}) \\ & + (1 - t_{ij}) \log\left(1 - p(t_{ij} = k|\mathbf{s}_{ij})\right)] + \frac{\lambda}{2}\left\|\boldsymbol{\theta}\right\|_2^2, \end{aligned} \tag{18}$$

where the index set I denotes the used training sentence pairs, and $\lambda > 0$ is the regularization parameter set by the user. The training of the proposed text matching model involves the bilinear similarity weights and biases $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{b}_s\}$, a set of memory network parameters $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{W}_r, \mathbf{W}_c, \mathbf{w}_{p_1}^t, \mathbf{w}_{p_2}^t \mathbf{w}_a, \mathbf{w}_q\}$, as well as the softmax parameters $\boldsymbol{\alpha}_0$ and $\boldsymbol{\alpha}_1$.

3.5.2 Positional Decoder

In the machine comprehension task, the boundary detecting method [36] is adopted for training process, it minimizes the sum of the negative log probabilities of the true start and end position by the predicted distributions, the loss for start and the end position is defined as

$$L_{ag}(\boldsymbol{\theta}) = -\sum_{i \in D} \log \mathbf{p}_{\mathrm{s}}(y_i^{\mathrm{s}}) + \log \mathbf{p}_{\mathrm{e}}(y_i^{\mathrm{e}}), \tag{19}$$

where the index set D denotes the training question and context pairs. y_i^{s} and y_i^{e} are the ground-truth start and end position indices of i-th pair, respectively. The model overall variable $\boldsymbol{\theta}$ is the set of entire trainable weights and bias, including bi-linear similarity and multi-dimensional memory network parameters.

TABLE 1
Dataset content statistics.

Parameter	TREC	WikiQA	SQuAD	TriviaQA
No. of Questions	1,505	3,047	97,000	77,400
No. of Answers	60,800	29,258	-	-
No. of Contexts	-	-	20,800	138,538
Avg. Question Length	11.39	7.26	11.0	15.0
Avg. Answer Length	24.63	24.94	-	-
Avg. Context Length	-	-	122	495

4 Evaluation Setup

4.1 Datasets

In answer selection, we focus two benchmark datasets: TREC and WikiQA datasets. TREC[1] [37], is generated from TREC QA tracks 8-13, which each contain a set of factoid questions and candidate answers [37]. The correct answers for each question are manually labeled and ranked in the dataset. WikiQA[2] [24], is the public released QA dataset in which all answers are collected from Wikipedia.

In machine comprehension, we also adopt two authoritative datasets to evaluate the proposed model: Stanford Question Answering Dataset (SQuAD) and TriviaQA datasets. The SQuAD dataset [38] totally consists of more than 100k questions manually annotated by crowd sourcing workers on 536 Wikipedia articles. Each question corresponds to contexts is a paragraph collected from an article. The best answer responses to question is a segment of text to be a span of contexts. The dataset contains 87k question context tuples for training, 10k tuples for validation. We follow the same experimental setting as in [39] by dividing 10% of training samples as the test set, and computing performance when training on subsets of the remaining samples of the entire dataset.

TriviaQA[3] [40] is a recent popular QA dataset consisting of over 650K question-answer-context triples, which consist of 95K Trivia QA pairs with the average six contexts as supporting evidence for each question. TriviaQA web dataset is derived from TriviaQA database with unfiltered strings in each question context pair. The mean length of original contexts contains 2,895 words is much larger than the one in SQuAD dataset. Thus, we truncate the contexts followed with [7], and reduce the average length of contexts to 495 words. We will test the full and verified subsets of TriviaQA web dataset in experimental evaluation, where the verified subsets contains the part of full dataset correctly answered question context pairs. More detailed data information for datasets is shown in Table 1.

4.2 Performance Measures

In this paper, we use two performance metrics to measure the performance in answer selection, namely mean reciprocal rank (MRR) and mean average precision (MAP), as in [41]. The MRR metric focuses on the order of the correct answers. While MAP accumulates the mean ranking of all the correct answers in each question.

In machine comprehension, we use two different performance metrics to evaluate the model's accuracy. Exact match (EM) [36] measures the precision of predicted answer that match any one of the groundtruth answers exactly overall test question context pairs. The exact match score is equal to 1 when the prediction is exactly the same as groundtruth or 0 otherwise. Alternative metric is F1 score [42] that indicates the average of word overlap between the prediction and ground truth answer, where the predicted answer and groundtruth are treated as bags of words. In evaluation, we adopt the maximal F1 score over all of the possible groundtruth answers for a given question, and then average it over all of test questions.

4.3 Experimental Configuration

Experimental platform and recordings: All the training and testing were carried on a workstation equipped with Nvidia GeForce RTX 2080 graphics card, 8G RAM, and running the new version of the Tensor Flow Framework (v1.6) supported by Nvidia CUDA library (v.8.0).
Neural network configurations: In answer selection task, considering the top-k max pooling process, the number of top ranked word pairs k is set to 10 for question, and 30 for answer. In pre-processing step, the number of relevant questions l_q=10, and the number of relevant answers l_a=50. The total number of iterations or hops is defined as T=3 for memory update mechanism. In machine comprehension, the number k is set to 15 for question, and 150 for context. The total number of iterations is T=4 for memory update.
Training preparation and initialization: In preparing the dataset process, we followed the same text pre-processing procedures described in [43]. To initialize the word embedding of MMN, the basic pre-training model is Glove [44] using a corpus containing 27B words from 2B Tweets. The Tweets have been filtered by removing infrequent words, resulting in 1.2M words from the English vocabulary. The dimensionality of each word embedding vector is set as 100. In this work, we also adopted the pre-trained RoBERTa-Large model as in [3]. For words appearing in each dataset, but out of vocabulary, a random value uniformly sampled from the interval of $[-0.3, 0.3]$ is assigned to each embedding dimension. For model variables to be initialized, a normal distribution $\mathcal{N}(0, 0.1)$ is used.
Training/testing process: For model optimization, a root mean square propagation (RMSProp) optimizer is used. The process includes a mini-batch containing 50 training examples, a learning rate of 0.01, the regularization parameter of 0.9 and a dropout rate of 0.5 [45]. The learning rate is halved after 10 epochs. Gradient clipping [46] is used to scale the gradient when the norm of gradient exceeds a threshold of five. The overall datasets have been split for training, testing and development purposes as suggested by the original datasets [24]. These are given in Table 2. The parameters adopted above were selected using the development set based on coarse manual tuning. For the model fine-tuning process for answer selection task, we used the same transfer learning technique with [2], it first pre-train the model on the StackExchangeQA dataset, then fine-tune the pre-trained model on the datasets.

4.4 Baselines

In this work, we divided the compared models into three different categories according to the technology they used.

1. http://trec.nist.gov/data/qa/t8qa_data.html
2. https://aka.ms/WikiQA
3. http://http://nlp.cs.washington.edu/triviaqa/

TABLE 2
Benchmark data splits.

Data Set	Q/A Pairs	Development	Training	Testing
TREC [37]	8,997	1,148	4,718	1,517
WikiQA [24]	29,258	2,733	20,360	6,165
SQuAD [38]	100,000	10,000	78,300	8,700
TriviaQA [40]	78,582	7,900	61,800	7,700

Several well-known works are introduced as follows:

Models without pre-training:

1) CAM [13]: The model performs different comparison matching functions to match the sentences based on word-level, where the similarity outputs from the function are aggregated into a vector by a convolution layer. The convoluted vector is fed into the prediction layer to compute the matching score.
2) SLQA [33]: The model provides the fusion functions to combine self-attention and similarity matrix to complete the machine comprehension.

Models with pre-training:

1) BERT [4]: The model demonstrates the deep transformers for pre-training the bidirectional word representations, which are used in the matching layer followed by fine- tuning the parameters of the model. The advanced RoBERTa model [6] is proposed by optimizing the parameters of basic BERT.
2) GSAMN [2]: It adopts the input self-attention mechanism to update memory cell using the compare aggregation operation.
3) TANDA [3]: The recent model builds transfer learning architecture to transfer and pre-trains the model from a large QNLI corpus, then applies it on the target datasets.

Memory networks:

1) KV-MemNNs [23]: The model stores facts in a key-value structure, where the key is used to locate the question and the corresponding value is returned as the answer.
2) MEMEN [27]: The model designs an attentive memory to learn an alignment memory matrix, which contains the syntactic and semantic information of the words returned by skip-gram model.
3) M-Reader [7]: The reinforced memory model uses re-attention mechanism to refine current attentions for generating answer words.
4) EMR [9]: By training a reinforcement learning agent, it decides which memory vector to update when the memory module is full.

5 Results and Analysis

5.1 Comparison with State-of-the-art Methods

5.1.1 Answer Selection

We first evaluate the performance of proposed MMN model in AS task, the proposed method is compared with several state-of-the-art approaches using the benchmark TREC and WikiQA datasets. For evaluation, we compare the performance of the proposed model across a number of techniques using the MAP and MRR metrics. The top two sub-tables in Table 3 report the MRR and MAP metrics for different models with and without pre-training. It can be seen from table that the pre-training MMN with transfer learning technique ("MMN+Tr") provides the competitive performance under the evaluation setups. We report a number of more specific observations:

- When compared against the TREC dataset, the proposed model gains a competitive performance for both BERT-L and RoBERTa-L. BERT based MMN outperforms TANDA on MRR and MAP performance by 1.38%, and 1.7% respectively.
- Among the pre-trained models, because the TANDA model adopts an external library database, which may cause the MRR and MAP performance of our model are 0.88%, and 2.5% lower than that of TANDA based on the RoBERTa-L combined transfer learning for TREC dataset. While BERT plays a higher utility in our proposed model, the BERT-L based MMN with transfer learning outperforms the TANDA approach on MRR and MAP performance on TREC dataset, by 0.92%, and 2.91% respectively.
- When evaluated against the WikiQA dataset, the RoBERTa-L based proposed approach with transfer learning outperforms all models with respect to MRR and MAP results. In particular, the proposed model outperforms the second best performing TANDA+Tr, by 1.3%, and 1.23% respectively.
- Considering the impact of the pre-training scheme, the MMN with RoBERTa model provides a big improvement of MRR and MAP results under the WikiQA dataset, by 13.28%, and 13.88% respectively.

Although most recent works are likely to use structured CNN, RNN or transformer-based model, there is a memory network called KV-MemNNs in the Table offers a good performance on both two datasets. With respect to KV-MemNNs model using the external knowledge database, our proposed memory network focuses on utilizing the related pairs to a question-answer pair, it aims to reduce the larger computation and manual data creation caused by the external information resource. Additionally, different with KV-MemNNs model discards the original input information to refine memory network, we prefer to explore the memory-aware attention approach to involve the inputs in order to prevent the information loss after several iterations. Overall, the proposed model outperforms the most of the existing works in answer selection task.

5.1.2 Machine reading comprehension

To analyze the proposed model effectiveness in machine comprehension task, we test the model performance using TriviaQA and SQuAD datasets. The sub-table at bottom left in Table 3 illustrates that the EM and F1 scores for different models evaluated on two types datasets of TriviaQA dataset: Full, Verified. A longer length of context increases the complexity to memory the sentences information and search the answer spans of context for the TriviaQA dataset. From the results in Table, the proposed MMN model shows the state-of-the-art performance among the comparison models on more complex dataset. In the following, we report a number of specific points from the table:

- With respect to both EM and F1 scores, the proposed approach outperforms all models when evaluated

TABLE 3
Performance comparison of different models across a range of datasets. The memory networks are marked in bold. The best results are highlighted and the second best results are underlined.

Models	TREC		WikiQA	
	MRR	MAP	MRR	MAP
Random Guess [47]	0.4905	0.4335	0.4733	0.4253
WordEmbed [48]	0.6107	0.5537	0.5697	0.5065
AP-CNN [49]	0.8511	0.7530	0.6957	0.6886
Ab-CNN [19]	0.8539	0.7741	0.7108	0.6921
KV-MemNNs [23]	0.8523	0.7857	0.7265	0.7069
IARNN [50]	0.8208	0.7369	0.7418	0.7341
BiMPM [51]	0.8750	0.8020	0.7310	0.7180
IWAN [11]	0.8890	0.8220	0.7500	0.7330
CAM [13]	0.8659	0.8145	0.7545	0.7433
MMN	0.8865	0.8390	0.7847	0.7659

Pre-trained Models		TREC		WikiQA	
		MRR	MAP	MRR	MAP
BERT-L	GSAMN [2]	0.9490	0.9060	0.8320	0.8210
	TANDA [3]	0.9460	0.9040	0.8530	0.8360
	MMN	0.9598	0.9210	0.8742	0.8635
	GSAMN+Tr [2]	0.9570	0.9140	0.8720	0.8570
	TANDA+Tr [3]	0.9670	0.9120	0.9120	0.9040
	MMN+Tr	**0.9762**	**0.9411**	0.8890	0.8731
RoBERTa-L	TANDA [3]	0.9280	0.8800	0.9190	0.9100
	MMN	0.9421	0.8938	0.9175	0.9047
	TANDA+Tr [3]	0.9740	0.9430	0.9330	0.9200
	MMN+Tr	0.9652	0.9180	**0.9460**	**0.9323**

Models	Full		Verified	
	EM(%)	F1(%)	EM(%)	F1(%)
Classifier [40]	23.40	27.70	23.60	27.90
BiDAF [52]	40.26	45.74	47.47	53.70
MEMEN [27]	43.16	46.90	49.28	55.83
M-Reader [7]	46.94	52.85	54.45	59.46
QANet [20]	51.10	56.60	53.30	59.20
document-qa [53]	63.99	68.93	67.98	72.88
BiDAF+SA [39]	-	-	69.03	74.61
SLQA [33]	66.56	71.39	74.83	78.74
EMR-biGRU [9]	52.50	57.57	-	-
EMR-Transformer [9]	48.43	53.81	-	-
MMN	**68.69**	**73.57**	**75.95**	**79.67**

Models	Dev Set		Test Set	
	EM(%)	F1(%)	EM(%)	F1(%)
LR Baseline [38]	40.0	51.0	40.4	51.0
Match-LSTM [36]	64.1	73.9	64.7	73.7
DCN+ [54]	74.5	83.1	75.1	83.1
Interactive AoA Reader [55]	-	-	73.6	81.9
FusionNet [56]	-	-	76.0	83.9
SAN [57]	76.2	84.0	76.8	84.4
BiDAF + SE [58]	-	-	78.6	85.8
MEMEN [27]	-	-	75.4	82.7
R-Net+ [59]	-	-	79.9	86.5
QANet [20]	-	-	76.2	84.6
M-Reader [7]	78.9	86.3	79.5	86.6
MMN	**79.8**	**87.1**	**80.2**	**87.3**

against the Full and Verified datasets. When considering the EM performance, the proposed outperforms the second best performing hierarchical attentive model SLQA, by 2.13%, and 1.12% respectively.

- When considering the F1 performance, the proposed approach outperforms the SLQA model, by 2.18%, and 0.93% respectively, on Full and Verified datasets.
- the proposed approach performs much better than the Classifier model, where EM performance is improved by 45.29%, and 52.35% respectively, and F1 performance is improved by 45.87%, and 51.77% respectively, on Full and Verified datasets.

The above results verify the proposed model not only offers good performance on a small subset of dataset, e.g. Verified datset. It also shows the proposed is capable of performing a robust performance with a large scale dataset, e.g. the Full TriviaQA dataset. The second best model SLQA stacks the intermediate representations of question and context pair using multiple attention functions, without considering the order of words in a long context situation. In summary, the result shows that the proposed model offers the best performance among other published results.

Further, we conduct the evaluation using the community SQuAD dataset. The proposed model is compared with ten state-of-art neural network models for this dataset, including model QANet [20], BiDAF+SE [58], and M-Reader [7]. A LR baseline model based on linear regression is given to provide a standard view among all results. For evaluation, we use two different datasets to test model: Dev and Test sets. The performance for different models along with the proposed model is reported at the bottom right sub-table in Table 3. It can be seen that the proposed model performs better than the competing models on both Dev and Test sets, demonstrating the superiority of the proposed model and its memory network strategy.

5.2 Ablation Study

In this section, we investigate different components of the proposed model with alternative design options, to analyze the effectiveness of the proposed model. Table 4 summarizes performance of the compared settings over WikiQA and SQuAD datasets for two tasks.

To assess the absolute advantage over the proposed version, we set the performance of MMN as the benchmark, and define the percentage gain on MRR and MAP performance as:

$$\delta_{\mathbf{g}}(\mathbf{x}) = \frac{\mathbf{g}_{\mathbf{x}} - \mathbf{g}_{(\mathrm{MMN})}}{\mathbf{g}_{\mathbf{x}}} \tag{20}$$

where the $\mathbf{g} \in \{(\mathrm{MRR}), (\mathrm{MAP}), (\mathrm{EM}), (\mathrm{F1})\}$, and $\mathbf{x}$ denotes the variations of the proposed models.

Memory refinement structure: An attentive memory refinement mechanism in the proposed model based on the multi-dimension strategy, which refines the memory matrix from two heads individually. We analyze the memory network under the four compared conditions: 1) MMN-Mr: Memory only updates the column side but without updating the row side; 2) MMN-Mc: Memory only updates the row side but without updating the column side; 3) MMN-Md: An generally choice of almost works is to update the two sides in a meanwhile by refining a memory matrix directly; 4) MMN-M: The proposed model without the memory network.

It can be seen from the table that the proposed memory network results in the best performance, followed by MMN-Mc. Such a result represents that refining the memory with two dimensions is beneficial for the proposed model, where

TABLE 4
Effects of multiple experiment settings used in AS and MRC tasks. The smallest gain values of each setting are marked in bold.

Models	WikiQA				SQuAD			
	MRR	δ_{MRR}	MAP	δ_{MAP}	EM	δ_{EM}	F1	δ_{F1}
MMN-Mc	75.96	-3.20	73.90	-3.64	78.60	-2.04	85.80	-1.75
MMN-Mr	75.72	-3.63	73.78	-3.81	78.10	-2.69	85.30	-2.34
MMN-M	73.00	**-7.49**	71.05	**-7.80**	77.50	**-3.48**	84.10	**-3.80**
MMN-Md	75.63	-3.76	73.40	-4.35	78.40	-2.30	85.60	-1.99
MMN-sim	77.12	-1.75	75.71	-1.16	-	-	-	-
MMN-norm	75.37	**-4.11**	73.80	**-3.78**	-	-	-	-
MMN-avg	76.90	-2.06	74.85	-2.32	-	-	-	-
MMN-bilinear	76.72	**-2.28**	74.67	**-2.57**	-	-	-	-
MMN	78.47	base	76.59	base	80.20	base	87.30	base

TABLE 5
Memory-aware attention refinement with three QA sentence pairs and the supported relevant Q'A' pairs for memory refinement operation with three hops ($T = 3$). In (q', a') pairs ranking list, the top three relevant questions Q'_t, the relevant, irrelevant top three answers $P_{A'_t}$ and $N_{A'_t}$ are ranked; In the candidate answers ranking list with each hop, a correct answer is labeled as A_+, an incorrect one is marked as A_-. The subject words in question are marked in orange colour, the object words in $P_{A'_1}$, $P_{A'_2}$ and $P_{A'_3}$ are marked in purple, cyan and blue colours, respectively.

	Question Q**:** how does interlibrary loan work?	**Supported** (q', a') **pairs ranking list (topic="loan")**
Hop-1	A_{11}: The lending library usually sets the due date and overdue fees of the material borrowed. (A_-)	Q'_1: what would be the deliverables?
	A_{12}: The user makes a request with their local library, which, acting as an intermediary, identifies owners of the desired item, places the request, receives the item, makes it available to the user, and arranges for its return. (A_+)	Q'_2: what are points on a mortgage?
	A_{13}:In many cases, nominal fees accompany interlibrary loan services. (A_-)	Q'_3: what kind of school is MIT?
Hop-2	A_{21}: The user makes a request with their local library, which, acting as an intermediary, identifies owners of the desired item, places the request, receives the item, makes it available to the user, and arranges for its return. (A_+)	$P_{A'_1}$: Interlibrary loan (abbreviated ILL, and sometimes called interloan, document delivery, or document supply) is a service whereby a user of one library can borrow books or receive photocopies of documents that are owned by another.
	A_{22}: The lending library usually sets the due date and overdue fees of the material borrowed. (A_-)	$P_{A'_2}$: A deliverable could be a report, a document, a server upgrade or any other building block of an overall project.
	A_{23}: Interlibrary loan (abbreviated ILL, and sometimes called interloan, document delivery, or document supply) is a service whereby a user of one library can borrow books or receive photocopies of documents that are owned by another. (A_+)	$P_{A'_3}$: By charging a borrower points , a lender effectively increases the yield on the loan above the amount of the stated interest rate .
Hop-3	A_{31}: The user makes a request with their local library, which, acting as an intermediary, identifies owners of the desired item, places the request, receives the item, makes it available to the user, and arranges for its return. (A_+)	$N_{A'_1}$: Borrowers can offer to pay a lender points as a method to reduce the interest rate on the loan, thus obtaining a lower monthly payment in exchange for this up-front payment.
	A_{32}: Interlibrary loan (abbreviated ILL, and sometimes called interloan, document delivery, or document supply) is a service whereby a user of one library can borrow books or receive photocopies of documents that are owned by another. (A_+)	$N_{A'_2}$:Although books and journal articles are the most frequently requested items, some libraries will lend audio recordings, video recordings, maps, sheet music, and microforms of all kinds.
	A_{33}: The lending library usually sets the due date and overdue fees of the material borrowed. (A_-)	$N_{A'_3}$: In many cases, nominal fees accompany interlibrary loan services.

the performance of MMN-Mc is better than the one in column side MMN-Mr. Whereas the proposed model without memory network MMN-M offers the worst performance, leading to the gain values of MRR and MAP results decrease by 7.49% and 7.80%, receptivity. The memory matrix of MMN-Md provides a worse performance than the proposed method. The computational time of MMN approximately costs 15 hours for WikiQA dataset, it is less than 10 hours of MMN-Md due to the specific memory refining structure of MMN, which verifies the network design of the proposed method is more effective. From the empirical observation, a similar performance happens on the SQuAD dataset.

Memory initialization method: In the answer selection task, we investigate the effectiveness of the proposed memory initialized matrix $\mathbf{M}_0$ over three variants: 1) MMN-sim: the similarity matrix $\mathbf{S}$ in Eq.(2) is set as the initialized matrix; 2) MMN-norm: the elements of memory matrix are initialized by a norm distribution $\mathcal{N}(-1, 1)$ at time $t = 0$; 3) MMN: the proposed memory pools of relevant question and answer pairs are used to initialize the memory network in the section 3.1.4.

As can be seen from the table, the MRR and MAP performance of the proposed method achieves a gain of 1.75% and 1.16% compared to the second best MMN-sim method on WikiQA dataset. The MMN-norm obtains the worst performance. The proposed method deigns the memory pools to aggregate a number of important relevant sentence pairs for answer selection. The results verify that the related information of corpus could enrich the content of memory block for short sentence's semantic matching.

Aggregated sentence vector function: In the answer selection task, we consider three design options of learning

Web Rank	Hops	Answers	
			Q: The ancient poetess Sappho who wrote emotional verses with other females as subjects was born on what Greek island?
1	Hop-1	poet Sappho	C_1: other females instead. Another explantion for where the meaning of the word lesbian derives, from is the ancient Greek female poet Sappho, who was born in Lesbos.
	Hop-2	Lesbos	
2	Hop-1	poet Sappho	C_2: Little Lesbos is just minutes by boat from the island of ... is the ancient Greek female poet Sappho, who was born in Lesbos and who wrote emotional verses ..."
	Hop-2	Little Lesbos	
	Hop-3	Lesbos	
3	Hop-1	poet Sappho	C_3: lesbian derives from is the ancient Greek female poet Sappho, who was born in Lesbos and who wrote emotional, verses aimed.
	Hop-2	lesbian derives	
	Hop-3	Lesbos	

Fig. 6. Visualized example for web search over TriviaQA dataset.

d-dimensional sentence representation vector in the 3.1.3 section: 1) MMN: non-parametric l_2-norm function Eq.(3) of our proposed model aggregates the word elements in sentence embedding matrix; 2) MMN-bilinear: the function is used to transform the sentence embedding matrix to a vector; 3) MMN-avg: reducing a sentence representation dimension based on averaged sum weighted function by stacking the elements in sentence embedding matrix.

As seen from the table that, by using neither bi-linear nor average weighted sum as a transformation function, a fairly low performance is obtained since the lack of regularization of distributed sentence presentation. In detail, the MRR and MAP performance of MMN-bilinear are less than our proposed design by 2.28% and 2.57%, respectively.

5.3 Case Study

A running example with three QA sentence pairs in Table 5 shows the effect of refinement operation with three hops for WikiQA dataset. It can be seen from the table that memory refinement operation demonstrates the correct answer moves closer to the question in ranking list while incorrect one moves further. The contents of relevant pairs (q', a') are stored in memory pool. It is interesting to observe that, the relevant answer $P_{A'_1}$ has the similar meaning to the original ground truth A_{12}, and the $P_{A'_2}$ and $P_{A'_3}$ indicate the correct answer A_{23}, so it is eventually moved to the second position. The insignificant attention words and phrases are shown in the same color, they are positive for matching, while the irrelevant pairs are marked by the underline.

Fig.6 displays the ranked documents web search for the MRC example. For the question with "who" and "what" words, the proposed model generated the highlighted answer in blue from different contexts within three hops. The rank order of documents is determined by the index of hops, which means, fewer hops have a higher speed search.

6 Conclusions

In summary, we have proposed a novel memory network architecture is capable of completing multi-tasks in the QA field. The key idea of the proposed memory approach is to improve the memory refinement system for long-term semantic matching. To achieve the goal, we propose the row and column based write heads of refined memory matrix to easily extract the semantic information according to the question and candidates. Overall, our proposed model collects a top number of related questions and answers to initialize the memory block. Multiple performance comparisons with the state-of-art approaches also demonstrate the superiority of the proposed model for community data.

Acknowledgment

This work was supported by the Central Guidance on Local Science and Technology Development Fund of HuBei Province (Grant No. 2018ZYYD025), Natural Science Foundation of Hubei Province (Grant No. 2021CFB255).

References

[1] N. Chomsky, *Deep Structure, Surface Structure and Semantic Interpretation: An interdisciplinary reader in philosophy, linguistics, and psychology,*. Cambridge University Press, 1971.

[2] T. Lai, Q. H. Tran, T. Bui, and D. Kihara, "A gated self-attention memory network for answer selection," in *Proceedings of EMNLP-IJCNLP*. Association for Computational Linguistics, 2019.

[3] S. Garg, T. Vu, and A. Moschitti, "Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, Apr 2020, pp. 7780–7788.

[4] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 16th NAACL-HLT Annual Conference*, 2019.

[5] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, p. 2227–2237.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[7] M. Hu, Y. Peng, Z. Huang, X. Qiu, F. Wei, and M. Zhou, "Reinforced mnemonic reader for machine reading comprehension," in *Proceedings of the 27th IJCAI International Joint Conference on Artificial Intelligence*, 2018, pp. 4099–4106.

[8] S. Gao, X. Chen, Z. Ren, D. Zhao, and R. Yan, "Meaningful answer generation of e-commerce question-answering," *ACM Trans. Inf. Syst.*, vol. 39, no. 2, pp. 1–26, 2021.

[9] M. Han, M. Kang, H. Jung, and S. J. Hwang, "Episodic memory reader: Learning what to remember for question answering from streaming data," in *Proceedings of the 57th ACL Conference on Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 4407–4417.

[10] M. Tan, B. Xiang, and B. Zhou, "Memory networks," in *Proceedings of the 3th International Conference on Learning Representations*, 2015.

[11] G. Shen, Y. Yang, and Z.-H. Deng, "Inter-weighted alignment network for sentence pair modeling," in *Proceedings of the 2017 Conference on EMNLP*, 2017, pp. 1179–1189.

[12] B. Jin, E. Chen, H. Zhao, Z. Huang, Q. Liu, H. Zhu, and S. Yu, "Promotion of answer value measurement with domain effects in community question answering systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2019.

[13] S. Wang and J. Jiang, "A compare-aggregate model for matching text sequences," in *Proceedings of the 5th ICLR International Conference on Learning Representations*, 2017.

[14] T. Zhao, Z. Yan, Y. Cao, and Z. Li, "Asking effective and diverse questions: a machine reading comprehension based framework for joint entity-relation extraction," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 3948–3954.

[15] Y. Feng, X. Chen, B. Y. Lin, P. Wang, J. Yan, and X. Ren, "Scalable multi-hop relational reasoning for knowledge-aware question answering," in *the 2020 Conference on EMNLP*, 2020, pp. 2814–2828.

[16] K. Ma, F. Ilievski, J. Francis, Y. Bisk, E. Nyberg, and A. Oltramari, "Knowledge-driven data construction for zero-shot evaluation in commonsense question answering," in *Proceedings of the 33th AAAI Conference on Artificial Intelligence*, 2021, pp. 13 507–13 515.

[17] Q. Zhang, X. Weng, G. Zhou, Y. Zhang, and J. X. Huang, "Arl: An adaptive reinforcement learning framework for complex question answering over knowledge base," vol. 59, no. 3, 2022, p. 102933.

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings the 6th ICLR International Conference on Learning Representations*, 2015.
[19] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "Abcnn: Attention-based convolutional neural network for modeling sentence pairs," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 259–272, 2016.
[20] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, "Qanet: Combining local convolution with global self-attention for reading comprehension," in *Proceedings of the 6th ICLR International Conference on Learning Representations*, 2018.
[21] C. Xiong, V. Zhong, and R. Socher, "Dynamic coattention networks for question answering," in *Proceeding of the 5th ICLR International Conference on Learning Representations*, 2017.
[22] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.
[23] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1400–1409.
[24] Y. Yang, W.-t. Yih, and C. Meek, "Wikiqa: A challenge dataset for open-domain question answering," in *Proceedings of ACL-EMNLP Conference*, 2015, pp. 2013–2018.
[25] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," in *Proceedings of the 33rd ICML Conference*, 2016, pp. 2397–2406.
[26] S. Sukhbaatar, J. Weston, and R. Fergus, "End-to-end memory networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2440–2448.
[27] B. Pan, H. Li, Z. Zhao, B. Cao, D. Cai, and X. He, "Memen: multi-layer embedding with memory networks for machine comprehension," *arXiv preprint arXiv:1707.09098*, 2017.
[28] C.-C. Hung, T. Lillicrap, J. Abramson, Y. Wu, M. Mirza, F. Carnevale, A. Ahuja, and G. Wayne, "Optimizing agent behavior over long time scales by transporting value," *Nature Communications*, vol. 10, p. 5223, 2019.
[29] Z. Lu and H. Li, "A deep architecture for matching short texts," in *Advances in NIPS*, 2013, pp. 1367–1375.
[30] T. M. Mitchell, "Machine learning. WCB," 1997.
[31] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the 3rd ICLR Conference on Learning Representations*, 2015.
[32] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances NIPS*, 2015, pp. 2692–2700.
[33] W. Wang, M. Yan, and C. Wu, "Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering," in *Proceedings of the 56th ACL Conference*, vol. 1, 2018, pp. 1705–1714.
[34] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, p. 947, 2000.
[35] A. Kumar, O. Irsoy, and P. Ondruska, "Ask me anything: Dynamic memory networks for natural language processing," in *Proceedings of the 33rd ICML International Conference on Machine Learning*, 2016, pp. 1378–1387.
[36] S. Wang and J. Jiang, "Machine comprehension using match-lstm and answer pointer," in *Proceedings of the 5th ICLR International Conference on Learning Representations*, 2017.
[37] M. Wang, N. A. Smith, and T. Mitamura, "What is the jeopardy model? a quasi-synchronous grammar for QA," in *Conference on Empirical Methods on Natural Language Processing*, vol. 7. Association for Computational Linguistics, 2007, pp. 22–32.
[38] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," in *The 2016 Conference on EMNLP*, 2016, p. 2383–2392.
[39] B. Dhingra, D. Pruthi, and D. Rajagopal, "Simple and effective semi-supervised question answering," in *Proceedings of the 16th NAACL-HLT Annual Conference*, 2018, pp. 582–587.
[40] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension," in *Proceedings of the 55th ACL Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 1601–1611.
[41] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*. ACM press New York, 1999, vol. 463.
[42] Y. Sasaki *et al.*, "The truth of the f-measure," *Teach Tutor mater*, vol. 1, no. 5, pp. 1–5, 2007.
[43] A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 373–382, 2015.
[44] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *Proceedings of the 2014 Conference on EMNLP*, vol. 14, 2014, pp. 1532–43.
[45] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[46] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks." in *Proceedings of the 30th ICML Conference*, vol. 28, 2013, pp. 1310–1318.
[47] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng, "A deep architecture for semantic matching with multiple positional sentence representations," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, 2016, pp. 2835–2841.
[48] L. Kang, B. Hu, X. Wu, Q. Chen, and Y. He, "A short texts matching method using shallow features and deep features," in *Natural Language Processing and Chinese Computing*. Springer, 2014, pp. 150–159.
[49] C. d. Santos, M. Tan, B. Xiang, and B. Zhou, "Attentive pooling networks," in *CoRR*, vol. abs/1602.03609, 2016.
[50] B. Wang, K. Liu, and J. Zhao, "Inner attention based recurrent neural networks for answer selection," in *Proceedings of the 54th ACL Conference*, vol. 1, 2016, pp. 1288–1297.
[51] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," in *Proceedings of the 26th IJCAI International Joint Conference on Artificial Intelligence*, 2017, pp. 4144–4150.
[52] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," in *Proceedings of the 5th ICLR International Conference on Learning Representations*, 2017.
[53] C. Clark and M. Gardner, "Simple and effective multi-paragraph reading comprehension," in *Proceedings of the 56th ACL Conference*, 2018, pp. 845–855.
[54] C. Xiong, V. Zhong, and R. Socher, "Dcn+: Mixed objective and deep residual coattention for question answering," in *Proceeding of the 6th International Conference on Learning Representations*, 2018.
[55] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, and G. Hu, "Attention-over-attention neural networks for reading comprehension," in *Proceedings of the 55th ACL Conference*, 2017, pp. 593–602.
[56] Y. Shen, P.-S. Huang, J. Gao, and W. Chen, "Reasonet: Learning to stop reading in machine comprehension," in *Proceedings of the 23rd ACM SIGKDD Conference*. ACM, 2017, pp. 1047–1055.
[57] X. Liu, Y. Shen, K. Duh, and J. Gao, "Stochastic answer networks for machine reading comprehension," in *Proceedings of the 56th ACL Conference*, 2018, pp. 1694–1704.
[58] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 56th ACL Conference on the Association for Computational Linguistics*, 2018, pp. 2227–2237.
[59] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, "Gated self-matching networks for reading comprehension and question answering," in *Proceedings of the 55th ACL Conference*, vol. 1, 2017, pp. 189–198.

APPENDIX

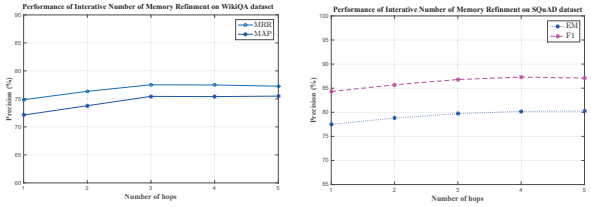
EFFECTS OF MULTIPLE EXPERIMENT SETTINGS

Memory-based attention mechanism: In this experiment, we examine the usage of attention mechanism under four different conditions: 1) MMN-Ar: the proposed model only uses the attention mechanism in column-based memory update, but not in row-based memory update. 2) MMN-Ac: the proposed model applies attention mechanism in column-based memory update. 3) MMN-A: the proposed model removes the attention mechanism in both two sides. 4)MMN: the proposed model contributes the memory-based attention mechanism both on the row and column sides.

We compare various designs of attention mechanism in memory refinement. As can be seen from the Table 6, MMN

TABLE 6
Effects of attention mechanism with MMN model used in AS and MRC tasks. The smallest gain values of each setting are marked in bold.

Models	WikiQA				SQuAD			
	MRR	δ_{MRR}	MAP	δ_{MAP}	EM	δ_{EM}	F1	δ_{F1}
MMN-Ac	76.50	-2.58	74.42	-2.92	79.10	-1.39	86.20	-1.28
MMN-Ar	76.01	-3.24	73.95	-3.57	78.70	-1.91	85.90	-1.63
MMN-A	75.48	**-3.96**	73.43	**-4.29**	78.00	**-2.82**	85.60	**-1.99**



(a) (b)

Fig. 7. The effects of the number of iterations during memory refinement.

TABLE 7
The comparison per-iteration complexity of different memory refinement types.

Refinement types	Complexity per iteration
MMN-Mc	$O(m \times 4 \times n)$
MMN-Mr	$O(m \times 4 \times n)$
MMN-Md	$O(m \times n^2)$
MMN	$O(m \times 8 \times n)$

causes the gain value of the WikiQA dataset to increase by at least 2.5% compared to the MMN-Ac and MMN-Ar for both MRR and MAP performance. It indicates that the attention mechanism on each dimension is important for improving the matching performance, where the contextual information as the input into the attention function.

Iterative number of memory refinement: The effect of memory update is investigated by examining different numbers of iteration T, e.g., $T \in \{1, 2, 3, 4, 5\}$. We compare the number of iterations in the memory refinement mechanism is increased from 1 to 5, the results are shown in Fig.7.

It can be seen from Fig.7(a) that the matching performance MAP and MRR increases until $T = 3$ over the WikiQA dataset, and then starts to be flat when $T = 4$. Differently, Fig.7(b) shows the predictive accuracy EM and F1 score continue to grow until $T = 4$ over the SQuAD dataset. Based on this empirical observation, $T = 3$ for answer selection task and $T = 4$ for machine comprehension task are sufficient to achieve robust performance.

Complexity analysis of memory module: An attentive memory refinement mechanism in the proposed model is based on the multi-dimension strategy, which refines the memory matrix from two heads individually. We analyze the complexity of the proposed memory network under the four compared conditions: 1) MMN- Mr: Memory only updates the column side but without updating the row side; 2) MMN-Mc: Memory only updates the row side but without updating the column side; 3) MMN-Md: An generally choice of almost works is to update the two sides in a meanwhile by refining a memory matrix directly; 4) MMN: The proposed model with the entire memory network.

Here, we calculate the complexity depending on the specific algorithm and parameters, where $m \times n$ denotes the size of memory matrix, in general, $m \leq n$. It can be seen from the Table 7 that the two optional choices of the proposed memory refinement: MMN-Mc and MMN-Mr have the same complexity according to the refinement algorithm. Thus, the complexity proposed method MMN with entire memory refinement is the sum of the MMN-Mc and MMN-Mr. Compared with the traditional method MMN-Md by using bilinear function with a large dimension of parameters, the MMN has a lower complexity due to the specific attention and Top-k pooing operations.

Jinmeng Wu received the B.Eng. and Ph.D. degrees in the department of electrical engineering and electronics from the University of Liverpool, Liverpool, U.K., in 2014 and 2019, respectively. She is currently a Lecturer in the School of Electrical and Information Engineering at Wuhan Institute of Technology. She is also serving in the Key Laboratory of Optical Information and Pattern Recognition. Her research interests include natural language processing and deep learning.

Tingting Mu received the B.Eng. degree in Electronic Engineering and Information Science from the Special Class for the Gifted Young, University of Science and Technology of China, Hefei, China, in 2004, and the Ph.D. degree in Electrical Engineering and Electronics from the University of Liverpool in 2008. She is currently a Lecturer in the School of Computer Science at the University of Manchester. Her research interests include machine learning and its applications to computer vision, natural language pro- cessing and text mining.

Jeyan Thiyagalingam received his Ph.D. degree in Computer Science from Imperial College, London, in 2005. Currently, he is a Lecturer at the University of Liverpool. Before joining the university, he held positions at Math- works, U.K. and at the University of Oxford. His research interests include computationally efficient algorithms and models, specifically for learning systems, target tracking, estimation, filtering and data processing. He is a fellow of the British Computer Society and also a member of IET and IEEE.

John Y. Goulermas obtained the B.Sc. (1st class) degree in computation from the University of Manchester (UMIST) in 1994, and the M.Sc. and Ph.D. degrees from the Control Systems Center, UMIST, in 1996 and 2000, respectively. He is currently a Professor in the Department of Computer Science at the University of Liverpool. His research interests include mathematical modeling, data analytics and machine learning. He has worked with various applications including image analysis, biomedical engineering, industrial monitoring and security.