EA Reader: Enhance Attentive Reader for Cloze-Style Question Answering via Multi-Space Context Fusion

Chengzhen Fu, Yan Zhang
Department of Machine Intelligence
Peking University
fuchengzhen@pku.edu.cn and zhy.cis@pku.edu.cn

Abstract

Query-document semantic interactions are essential for the success of many cloze-style question answering models. Recently, researchers have proposed several attention-based methods to predict the answer by focusing on appropriate subparts of the context document. In this paper, we design a novel module to produce the query-aware context vector, named Multi-Space based Context Fusion (MSCF), with the following considerations: (1) interactions are applied across multiple latent semantic spaces; (2) attention is measured at bit level, not at token level. Moreover, we extend MSCF to the multi-hop architecture. This unified model is called Enhanced Attentive Reader (EA Reader). During the iterative inference process, the reader is equipped with a novel memory update rule and maintains the understanding of documents through *read*, *update* and *write* operations. We conduct extensive experiments on four real-world datasets. Our results demonstrate that EA Reader outperforms state-of-the-art models.

Introduction

How to enable machines to answer questions with a document as the context has been a long-term goal of natural language processing and artificial intelligence. Towards this goal, several large-scale datasets of cloze-style questions over a context document have been released (Hermann et al. 2015; Hill et al. 2015), which contribute to training supervised question answering (QA) models.

Deep learning techniques, such as end-to-end neural networks, have achieved promising results on cloze-style question answering tasks. A common structure of these recent works can be summarized as an "encoder-interaction-prediction" framework (Chen, Bolton, and Manning 2016; Sordoni et al. 2016; Kadlec et al. 2016; Dhingra et al. 2017; Cui et al. 2017; Cheng, Dong, and Lapata 2016; Wang and Jiang 2016; Tan et al. 2017). In the *encoder phase*, all the words are transformed into low-dimensional dense vector representations and the recurrent neural network (RNN) is applied to encode contextual embeddings. In the *interaction phase*, the attention mechanism (Bahdanau, Cho, and Bengio 2015) allows the query to directly interact with the document at the token level. It then computes a weight distribution over document words according to their relevance to the query. In the *answer prediction phase*, the candidate token with maximum probability, estimated by either the joint query-document representation or the normalized attention vector, is selected as the predicted answer.

Most research focus on the interaction layer which is responsible for semantic interactions between documents and queries. Attention mechanisms, borrowed from the machine translation literature, are introduced to aggregate document representations with a weighted sum. Thus, the attention scoring function is essential for the quality of the obtained query-aware context vector. Existing attention scoring functions, such as *multiplicative attention* (Luong, Pham, and Manning 2015), *additive attention* (Bahdanau, Cho, and Bengio 2015) and *multi-head attention* (Vaswani et al. 2017), are mostly applied token-wise (Hermann et al. 2015; Kadlec et al. 2016; Dhingra et al. 2017; Huang et al. 2017; Paulus, Xiong, and Socher 2017). Therefore, the attention-based pooling can be formularized as $\mathbf{c} = \sum_{i=1}^{i=n} a_i \mathbf{v}_i$, where a_i is a scalar, $\mathbf{v}_i$ denotes the i-th token embedding and $\mathbf{c}$ is the query-aware context vector for prediction.

In this paper we use the term *bit* or *feature* to denote an element (such as v_{i1}) in a vector. A major downside for the token-wise attention mechanism is that, different bits in v_i are assigned to the same weight, which makes it hard to preserve only the highly related semantic features while discard those unrelated parts in the attended context vector. Bit-wise mechanism has been introduced to self-attention and its effectiveness has been verified through (Shen et al. 2018) in many sequence modeling tasks. It is also promising to learn query-document sophisticated semantic interactions by applying bit-wise to the inter-attention scoring function.

Another important issue is that the QA task consists of heterogeneous queries and various document topics. Recent work in (Xiong, Zhong, and Socher 2017) uses Highway Maxout Network to pool across multiple model variations. However, it brings much more parameters compared to a simple feed-forward network. We take the inspiration from multi-head attention proposed in (Vaswani et al. 2017). A simple dot product for modeling relevance may reduce the effective resolution of attention. To alleviate it, the multi-space structure is proposed. It has the advantage of jointly attending to document information from different representation subspaces. This mechanism has become successful in computing the context-aware features inside the en-

coder/decoder on the neural machine translation task.

Given that modeling semantic interactions within a single space sometimes fails to handle complex relations between documents and queries, it is promising to exploit the multi-space module to perform attention, with the purpose of extracting different aspects of the document into different attended context vector representations. In this way, multi-space attention can act as a mixture of experts to summarize useful clues for answer prediction. Notably, to control the model complexity, we constrain the product of the number of spaces and the dimension of each projected space equivalent to the dimension of original inputs.

Inspired by the above mentioned models, we propose a new module, named Multi-Space based Context Fusion (MSCF). It offers the following improvements to the previously popular attention paradigms. First, we project encoded embeddings for the query and the document into multiple latent spaces respectively. We hypothesize that the representation for each space will attend to a specific aspect of the semantics. Second, a novel bit-wise attention mechanism, is performed on each of these projected versions of representations in parallel. Different attentions from multiple subspaces are concatenated and once again projected, yielding a fused query-aware context vector.

As a shallow architecture may fail to understand the document and make inferences, the effectiveness of multi-hop reasoning has been explored so far in the literature (Sukhbaatar et al. 2015; Sordoni et al. 2016; Dhingra et al. 2017; Hu, Peng, and Qiu 2017; Xiong, Zhong, and Socher 2017; Kumar et al. 2016; Hu, Peng, and Qiu 2017). It allows the query to reason in a sequential way, based on the information that has been gathered previously from the documen,t. An iterative inference process usually involves alternate operations of *read* and *update*: (1) *read*: the query triggers an attention process that searches the memory (i.e., the whole document) and retrieves relevant facts to produce an "evidence" vector; (2) *update*: the query representation is renewed while taking into account the query of last hop as well as the newly formed evidence vector, which enables the model to attend to different inputs during each pass. After a fixed number of iterations, the model uses a summary of its inference process to predict.

In this paper, our multi-hop strategies tightly integrates previous ideas related to iterative attention processes. Moreover, we equip it with a novel *write* operation, unlike existing models (Sukhbaatar et al. 2015; Sordoni et al. 2016), do not maintain the memory constant, but instead evolve the memory over hops according to a gating writing rule. Specifically, a fusion gate is generated to control the degree to which the previous memory is exposed. The design of the fusion gate is very similar to long short-term memory network (LSTM) (Hochreiter and Schmidhuber 1997) and gated recurrent units (GRU) (Chung et al. 2014), wh,ich controls the information that flows into or out of memory, acting as a fine-grained information filter. In a word, our multi-hop architecture reasons about different parts of the document through *read*, *update* and *write* operations.

We combine MSCF with multi-step inference, and name the joint model **Enhanced Attentive Reader (EA Reader)**.

To summarize, our main contributions are three folds:

- We propose a novel module, named MSCF, to model fine-grained query-document interactions. It combines the multi-space mechanism with a bit-wise scoring function in a complementary manner.
- EA Reader allows multi-hop inference. It is equipped with a novel memory update rule and has an encoding memory that evolves over time and maintains the understanding of documents through *read*, *update* and *write* operations.
- We conduct extensive experiments on four benchmark datasets, and the results demonstrate that EA Reader outperforms several state-of-the-art models significantly.

Related Work

Cloze-style question answering tasks can be defined as: given a query and a document as the context, extract $a \in \mathbb{C}$. Note that, $\mathbb{C}$ is the set of candidate answers which appear in the document, and the answer a is a single word or entity.

According to the way to predict the answer, those methods generally fall into two categories: LSTM with Attention and Pointer-Style Attention sum.

LSTM with Attention

The first aims at computing a **joint query-document representation**, which is used to rank the candidate answers. **DeepLSTM Reader** (Hermann et al. 2015), the most strightforword way, processes the concatenated document-query pair by employing a deep LSTM cell with skip connections to obtain the joint representation.

To learn more complex query-document interactions, some models exploit the attention mechanism. This includes the **Attentive Reader** (Hermann et al. 2015; Chen, Bolton, and Manning 2016) which computes the query-aware document vector as the weighted sum of the token embeddings based on aligning scores computed by the attention scoring function; and the **Impatient Reader** (Hermann et al. 2015) which allows the model to recurrently accumulate information from the document as it sees each query token and thus builds document representation incrementally.

Attention mechanisms in previous works typically have one or more of the following characteristics. First, interactions are modeled in a token-wise fashion. Different features of the token embedding share the same inportance score, which makes it hard to distinguish the highly related semantic features of a word. Second, interactions are performed within a single space, which limits the ability of dealing with heterogeneous, highly flexible queries. Our proposed MSCF is closely related to Stanford Attentive Reader (Chen, Bolton, and Manning 2016), but has two special properties in comparison: (1) MSCF performs multiple spaces of attention, each of which reveals some aspects of semantic interactions, providing a better understanding of the query intents; (2) MSCF learns query-document interactions at the bit-wise level. The attention score map can select the features that best match the query, and incorporate this information into the query-aware context vector.

Pointer-Style Attention Sum

The other category consists of models motivated by **Pointer Network** (Vinyals, Fortunato, and Jaitly 2015). Unlike the classical softmax classifier, the document attention weights are directly used to predict the probability of the answer given the document. The **AS reader** (Kadlec et al. 2016) obtains an attention over the document by computing dot products between the query embeddings and contextual embeddings. An aggregation module named *pointer-sum attention* is further applied to sum the word's attention across all the occurrences.

Inspired by it, the **Attention-over-Attention (AoA) Reader** (Cui et al. 2017) exploit mutual information between the document and query based on query-to-document attention and document-to-query attention. The pointer-style attention mechanism to perform the final answer prediction has also been proposed by some mutli-hop models in the prediction phase (Sordoni et al. 2016; Dhingra et al. 2017). In contrast, our model predicts the answer based on the context vector rather than the pointer-sum attention.

Multi-Hop Architecture

The above mentioned models use a single-hop architecture. The benefit of multi-hop reasoning has also been explored.

Memory networks (Sukhbaatar et al. 2015; Kumar et al. 2016) focus on maintaining a memory component for QA tasks. Generally, the document representations are stored as the memory, and each token embedding can be regarded as a memory slot. **MemN2N** (Sukhbaatar et al. 2015) modifies it into an end-to-end recurrent architecture. The query representation is updated iteratively from hop to hop, which enables the model to attend to different inputs during each pass. **Iterative Attentive Reader** (Sordoni et al. 2016) also scan the document and the query iteratively to perform multi-hop reasoning. The major difference between them, is that MemNets embeds the query to obtain an internal state, whereas Iterative Attentive Reader performs an attentive read on the query encodings, resulting in a query glimpse during each iteration. **ReasoNet** (Shen et al. 2017) also combines dynamic reasoning steps with reinforcement learning. **GA Reader** (Dhingra et al. 2017) designs a novel module, called *gated-attention*, and is applied per hop to act as fine-grained information filters during the multi-step reasoning.

These extensions of memory networks have two things in common: (1) the multi-hop architecture is equipped with an explicit memory and a recurrent attention mechanism for reading the memory; (2) the encoded memory remains the same during each pass, i.e., can be only read but not written to. In contrast,we extends MemN2N by designing a *gating writing* operation which evolves the memory over iterations. The repeated, tight integration between queries and documents allows the model to store and filter context information dynamically, which is helpful to distinguish the parts of the document that are most salient to the answer prediction.

Enhanced Attentive Reader

In this section, we will give a detailed introduction to the proposed Enhanced Attentive Reader (EA Reader).

Contextual Encoding Representation

First, all the discrete tokens are mapped to a sequence of k-dimensional dense vectors via an embedding matrix $\boldsymbol{E} \in \mathbb{R}^{k\times|V|}$; therefore we have $\mathbf{x}_1^d, \ldots, \mathbf{x}_m^d \in \mathbb{R}^k$ for the context document and $\mathbf{x}_1^q, \ldots, \mathbf{x}_n^q \in \mathbb{R}^k$ for the query.

To incorporate some contextual information into the embedding of each word, we use a bidirectional GRU (Chung et al. 2014) with hidden size h to encode the context,

$$\begin{aligned}\vec{\mathbf{h}}_i &= \overrightarrow{\mathrm{GRU}}(\vec{\mathbf{h}}_{i-1}, \mathbf{x}_i^d), i = 1, \ldots, m \\ \overleftarrow{\mathbf{h}}_i &= \overleftarrow{\mathrm{GRU}}(\overleftarrow{\mathbf{h}}_{i+1}, \mathbf{x}_i^d), i = m, \ldots, 1\end{aligned} \tag{1}$$

and we concatenate each $\vec{\mathbf{h}}_i$ with $\overleftarrow{\mathbf{h}}_i$ to obtain $\mathbf{d}_i \in \mathbb{R}^{2h}$ for the i-th document word. Meanwhile, another BiGRU is applied to process the query and the last hidden state is picked up as the query embedding. Finally, we obtain two contextual encoded representations: $\boldsymbol{D} = \{\mathbf{d}_i\}_{i=1}^m \in \mathbb{R}^{2h\times m}$ for the document and $\mathbf{q} \in \mathbb{R}^{2h}$ for the query.

Multi-Space based Context Fusion

Question answering (QA) requires modeling complex interactions between the document and the query. Previous works, such as Stanford Attentive Reader (Chen, Bolton, and Manning 2016), use token-wise attention to perform semantic interactions and summarize the document into a fixed-size query-aware context vector.

We design a new module to generate the query-aware context vector, named Multi-Space based Context Fusion (MSCF), with the following considerations: (1) interactions are applied across multiple latent semantic spaces; (2) attention is measured at bit level, not at token level. Figure 1 provides an overview of the architecture of MSCF.

Latent Semantic Spaces The intuition behind using multiple spaces is that the QA task consists of heterogeneous queies and various document topics. These variations may require different spaces to perform attention-based pooling.

The projected embeddings in each space are derived from the contextual encoded representations by linear projection,

$$\begin{aligned}\boldsymbol{D}^l &= \boldsymbol{W}_d^l \cdot \boldsymbol{D} \\ \mathbf{q}^l &= \boldsymbol{W}_q^l \cdot \mathbf{q}\end{aligned} \tag{2}$$

with $\boldsymbol{W}_d^l \in \mathbb{R}^{2h/L\times 2h}$ and $\boldsymbol{W}_q^l \in \mathbb{R}^{2h/L\times 2h}$ being learned transformation matrices in the l-th space for the document and query respectively. L indicates the number of subspaces. To control the model complexity, we constrain that the dimension for each space is the same and the total dimensions of all spaces equivalent to that of original embeddings.

Bit-wise Attention Scoring Function For each space, attention is triggered when weighting different parts in the document according to their relevance to the query.

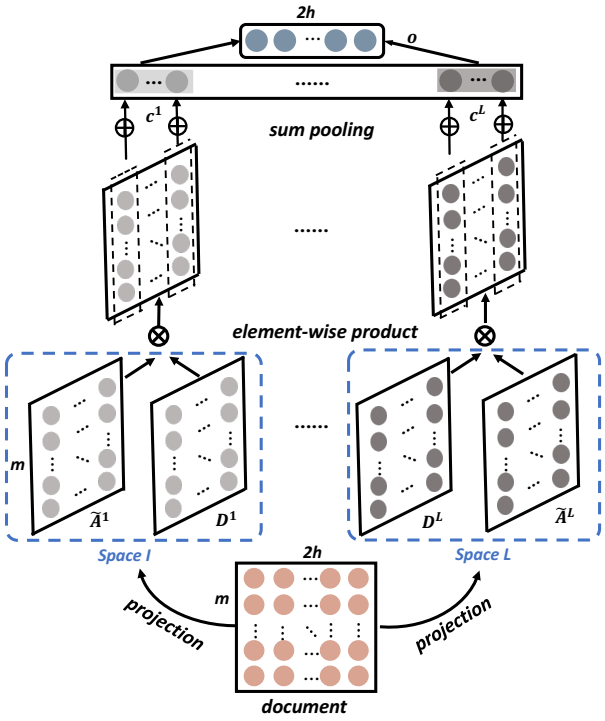

Figure 1: The overview of MSCF. The superscript denotes the serial number of subspaces. $\boldsymbol{D}$ denotes contextual encoded representations for documents. $\boldsymbol{D}^i$ and $\tilde{\boldsymbol{A}}^i$ are the projected document representations and attention score map for the i-th space. $\mathbf{o}$ denotes the query-aware context vector.

Previous attention mechanisms have one characteristic, is that weights are computed token-wise. *Multiplicative attention* uses dot product, and Stanford Attentive Reader (Chen, Bolton, and Manning 2016) chooses a more flexible bilinear term as in (Luong, Pham, and Manning 2015), i.e.,

$$r_i^l = (\mathbf{d}_i^l)^\intercal \boldsymbol{W}_b^l \, \mathbf{q}^l \tag{3}$$

Additive attention is associated with

$$r_i^l = (\mathbf{w}^l)^\intercal \tanh(\boldsymbol{W}_1^l \mathbf{d}_i^l + \boldsymbol{W}_2^l \mathbf{q}^l) \tag{4}$$

where the scalar score r_i^l indicates relevance between the i-th context word and its corresponding query in the l-th space.

In contrast, we propose a novel attention scoring function to learn relevance in a bit-wise fashion. Varied weights are assigned to different bits of an token embedding, which enable the query to attend to document semantics at fine-grained feature level rather than word level.

The bit-wise compatibility function is defined as,

$$\begin{aligned} \boldsymbol{A}_{:,i}^l &= \boldsymbol{W}^l \tanh(\mathbf{d}_i^l \odot \mathbf{q}^l)) \\ \tilde{\boldsymbol{A}}^l &= \mathrm{softmax}(\boldsymbol{A}^l) \end{aligned} \tag{5}$$

where $W^l \in \mathbb{R}^{2h/L \times 2h/L}$ is the inter-attention parameter matrix for the l-th space, and $\odot$ denotes Hadamard product.

Similar to a *feature map* in computer vision, we also define the normalized weight matrix $\tilde{\boldsymbol{A}}^l$ as an *attention score map*. The element $\tilde{A}_{ij}^l$ indicates the importance weight for the i-th feature over the j-th token in the l-th space.

Let $\mathbf{c}^l$ denotes the corresponding attended context vector in the l-th space. The computation is defined as,

$$\mathbf{c}^l = \sum_{i=1}^{i=m} \tilde{\boldsymbol{A}}_{:,i}^l \odot \mathbf{d}_i^l \tag{6}$$

Multi-Space Context Fusion As shown in Figure 1, we combine different inter-attentions from multiple subspaces and successively perform concatenation and non-linear transformation to refine the aggregated query-aware context representation as,

$$\mathbf{o} = \mathrm{relu}(\boldsymbol{W}_c[\mathbf{c}^1, \ldots, \mathbf{c}^L]) \tag{7}$$

where $W_c \in \mathbb{R}^{2h \times 2h}$ ensures that the output keeps the same shape (i.e., $2h$-dimensional) as the input.

Our MSCF module shares some similarities with the well-known multi-head attention (Vaswani et al. 2017), but with two differences. First, due to the property of QA tasks, we simply use projected representations of queries and documents as inputs to multiple subspaces of inter-attention. There is no extra derivation from inputs to sets of queries, keys and values. Second, we replace the simple scaled dot product with a fine-grained bit-wise attention. A sophisticated attention score map instead of a score vector is futher applied to guide the extraction of answers.

Multi-Hop Context Generator

EA Reader can also be extended to handle T hop operations, analogous to end-to-end memory network (Sukhbaatar et al. 2015). The document representation is regarded as a memory component that can be read and written to. Each document word vector is a memory slot.

Query Update Mechanism Our model renews the query representation during each pass, which allows the model to condition its attention on the result of previous iterations.

More specifically, multiple memory modules are stacked together by taking the output from last hop as input to the current hop. Similar to (Sukhbaatar et al. 2015), we apply a simple linear update across hops:

$$\begin{aligned} \mathbf{s}_0 &= \mathbf{q} \\ \mathbf{s}_{t+1} &= \boldsymbol{W}_s \, \mathbf{s}_t + \mathbf{e}_t \end{aligned} \tag{8}$$

where the linear mapping combines query $\mathbf{s}_t$ with "evidence" vector $\mathbf{e}_t$ retrieved from the previous hop. After T hops, the final state $\mathbf{s}_T$ is passed to the answer module.

Memory Read and Write Mechanism Figure 2 shows the workflow of multi-step reasoning. At step t, the query triggers a MSCF process over the memory slots to compute an overall memory, and obtains an "evidence" vector $\mathbf{e}_t$.

Prior work, such as the Iterative Attention (Sordoni et al. 2016), does not take the *write* operation into consideration. Our reader is equipped with a novel *write* operation that evolves the memory over multiple iterations.

At time step $t+1$, it writes the evidence gathered from *read* operation at the previous hop into slots of the memory.

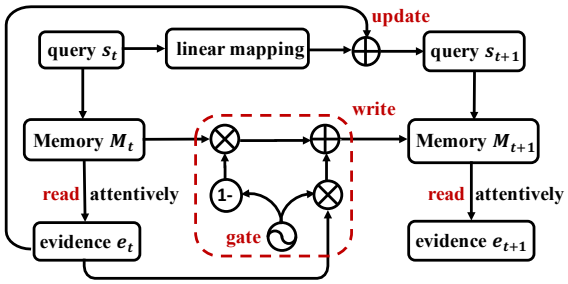

Figure 2: The workflow of multi-hop reasoning.

To efficiently fuse attended information $\mathbf{e}_t$ into memory $\boldsymbol{M}_t$, we take a simple gating operation to do information integration. The fusion gate is computed as,

$$\boldsymbol{G}_t = \sigma(\boldsymbol{W}_e[\mathbf{e}_t \otimes r_m] + \boldsymbol{W}_m \boldsymbol{M}_t) \tag{9}$$

where $\otimes$ denotes the outer product which duplicates its left vector m times to form a matrix. We use $\boldsymbol{G}_t$ and $\mathbf{1} - \boldsymbol{G}_t$ as the gated weights to assemble $\boldsymbol{M}_t$ and $\mathbf{e}_t$. The integrated information is computed by a weighted sum as:

$$\boldsymbol{M}_{t+1} = \boldsymbol{G}_t \odot (\mathbf{e}_t \otimes r_m) + (\mathbf{1} - \boldsymbol{G}_t) \odot \boldsymbol{M}_t \tag{10}$$

where $\mathbf{1}$ is a matrix of ones. Notably, we set the weight matrix $\boldsymbol{W}_s$, $\boldsymbol{W}_e$ and $\boldsymbol{W}_m$ for *update* and *write* operations to be the same across hops.

This memory update mechanism further encourages information to flow from query to document. Note that, eq.(10) has strong connections with the *write* module adopted in NSE (Munkhdalai and Yu 2017). In NSE, the attention weight vector emitted by the read module is reused to guide the update of memory. In contrast, we use a gate to dynamically determine how much of the past information needs to be erased and how much of the newly collected evidence needs to be passed along to the future.

Answer Prediction Module

Using the final query-aware context vector $\mathbf{s}_T$, the model computes the probability distribution over tokens as:

$$\mathbf{p} = \text{softmax}(\boldsymbol{W}_a \mathbf{s}_T) \tag{11}$$

The entity with maximum probability which appears in the passage is predicted as the answer. Model parameters are updated w.r.t. a negative log-likelihood objective.

Experiments

In this section, we conduct extensive experiments to answer the following questions:

- **(Q1)** How does our proposed MSCF perform in query-document interaction learning?
- **(Q2)** Is it necessary to extend EA Reader to the multi-hop architecture for question answering?
- **(Q3)** How does the settings of networks influence the performance of EA Reader?

We will answer these questions after presenting some fundamental experimental settings.

Experiment Setup

Datasets We evaluate the EA Reader on four large-scale datasets. The first two, **CNN and Daily Mail datasets**[1] are constructed with web-crawled CNN and Daily Mail news data (Hermann et al. 2015). One entity word is replaced with a special placeholder to indicate the missing token. Further, entities within each article are anonymized.

The next two datasets are formed from two different subsets of the **Children's Book Test**[2] (Hill et al. 2015). Each document consists of 20 continuous sentences in the story, and the 21st sentence is regarded as the query, where one word is blanked with a special symbol. We choose subsets whose answers belong to Named Entities (CBT-NE) or Common Nouns (CBT-CN).

Reproducibility We implement our method using Tensorflow[3]. Hyper-parameters of each model are tuned by grid-searching on the validation set. All tokens are initialized with the 100-dimensional pre-trained GloVe word embeddings (Pennington, Socher, and Manning 2014). The embedding matrix are updated during training. The hidden size h is 240. The number of latent semantic subspaces L is 6. We adopt Adam for optimization (Kingma and Ba 2014), with an initial learning rate of 0.001 and mini-batches of 32. We set GRU-dropout probability to 0.1 (Srivastava et al. 2014).

Performance Comparison among Single-Hop Models (Q1)

Table 1 shows a comparison of the performance of MSCF with previously published single-hop models. Note that, Attentive Reader and Impatient Reader are highly related to MSCF. They weight the document based on a token-level attention scoring function, whereas MSCF performs bit-wise semantic aligning across multiple spaces. As shown in Table 1, MSCF brings a substantial boost in performance, with nearly 6% absolute improvements. Compared with other prior work, MSCF outperforms the state-of-the-art single-hop systems BIDAF, with 2.4% and 3.1% absolute improvements on the CNN and Daily Mail testsets. Our model also could stay on par with the second-best baseline NSE when evaluated on the CBT datasets.

MSCF integrates the bit-wise attention into multi-space qeury-document interactions. While MSCF covers two distinct properties, we want to know whether it is indeed necessary and effective to combine them for jointly interaction learning. Table 1 also shows accuracy by removing one component at a time. The steepest reduction is observed when removing multi-space projections. Next, we observe a substantial drop when replacing the bit-wise attention with a bilinear term applied in Stanford Attentive Reader, which demonstrates the effectiveness of modelling interactions at fine-grained feature level. It also indicates multi-space projections are essential, which can be verified through the fact that it still provides 4% absolute improvements over Stanford AR on CNN datasets.

[1] http://cs.nyu.edu/~kcho/DMQA/

[2] http://www.thespermwhale.com/jaseweston/babi/CBTest.tgz

[3] https://www.tensorflow.org/

Table 1: Validation / Test accuracy (%) on four benchmark datasets for single-hop models. Results marked with "w/o" are obtained by removing a component. Best performance is in bold.

Model	Acc (%)							
	CNN		Daily Mail		CBT-NE		CBT-CN	
	Valid	Test	Valid	Test	Valid	Test	Valid	Test
Deep LSTM Reader (Hermann et al. 2015)	55.0	57.0	63.3	62.2	-	-	-	-
Attentive Reader (Hermann et al. 2015)	61.6	63.0	70.5	69.0	-	-	-	-
Impatient Reader (Hermann et al. 2015)	61.8	63.8	69.0	68.0	-	-	-	-
Stanford AR (Chen, Bolton, and Manning 2016)	72.4	72.4	-	-	-	-	-	-
BiDAF (Seo et al. 2017)	76.3	76.9	80.3	79.6	-	-	-	-
AoA Reader (Cui et al. 2017)	73.1	74.4	-	-	77.8	72.0	72.2	69.4
AoA Reader + Reranking (Cui et al. 2017)	-	-	-	-	**79.6**	**74.4**	**75.7**	**73.1**
NSE (Munkhdalai and Yu 2017)	-	-	-	-	78.2	73.2	74.3	71.9
MSCF (w/o multi-space projections)	72.9	73.4	76.3	71.7	74.2	70.1	70.2	68.5
MSCF (w/o bit-wise, + token-wise)	76.1	76.5	80.3	80.4	75.7	71.1	71.2	69.5
MSCF	**78.8**	**79.3**	**82.3**	**82.7**	78.1	73.1	73.9	71.5

Table 2: Validation / Test accuracy (%) on four benchmark datasets for multi-hop models. Results marked with "w/o" are obtained by removing a component. Best performance is in bold.

Model	Acc (%)							
	CNN		Daily Mail		CBT-NE		CBT-CN	
	Valid	Test	Valid	Test	Valid	Test	Valid	Test
Iterative Attention (Sordoni et al. 2016)	72.6	73.3	-	-	75.2	68.6	72.1	69.2
EpiReader (Trischler et al. 2016)	73.4	74.0	-	-	75.3	69.7	71.5	67.4
GA Reader (+ feature, fix $L(w)$) (Dhingra et al. 2017)	76.7	77.4	80.0	79.3	78.5	74.9	74.4	70.7
GA Reader (update $L(w)$) (Dhingra et al. 2017)	77.9	77.9	81.5	80.9	76.7	70.1	69.8	67.3
MSCF (i.e., single-hop)	78.8	79.3	82.3	82.7	78.1	73.1	73.9	71.5
EA Reader (w/o write operation)	80.2	80.9	82.9	83.4	78.3	75.1	75.2	72.5
EA Reader	**80.9**	**81.4**	**83.9**	**84.3**	**79.8**	**76.6**	**76.8**	**73.9**

Performance of Multi-hop Architecture (Q2)

By applying the multi-hop architecture to EA Reader, the performance increases by 2.1% and 1.6% again to set a new state of the art on the CNN and Daily Mail testsets respectively. Moreover, on the CBT-NE and CBT-CN test sets, it leads to an improvement of 1.7% and 0.8% over the most competitive model (achieved by GA Reader in Table 2, AoA Reader with the assistance of reranking strategies in Table 1, respectively). Our multi-hop model improves the accuracy to 76.6% and 73.9% for CBT-NE, CBT-CN respectively.

It can be observed that, removing the *write* operation leads to a reduction of about 1% and 1.5% on the Daily Mail and CBT-NE testsets. The gated writing function is responsible for filtering information irrelevant to the prediction.

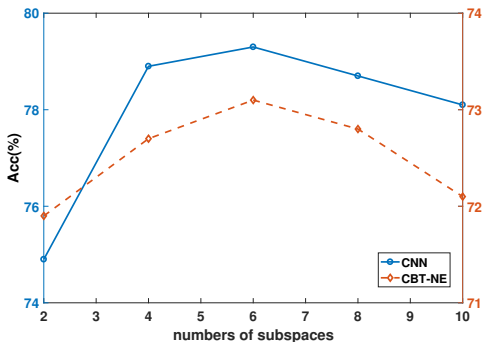

Figure 3: Impact of subspaces on Acc performance.

Hyper-Parameter Study (Q3)

Number of Subspaces We show the effect of varying the number of subspaces on the final validation set performance in Figure 3. A steep and steady rise in accuracy is observed as the number of hops is increased from 2 to 4. However, model performance degrades when the subspaces is set greater than 6, which is caused by overfitting.

Depth of Network As shown in Figure 4, model performance on CBT-NE validation dataset increases steadily when we increase the hops from 1 to 4, while on CNN, 3 is a more suitable setting for the number of hops.

Attention Scoring Function We compare four variants of operations, including *Additive, Multiplicative, Multi-Head* and *Bit-wise*. Results in Table 3 indicate that the feature-wise attention surpasses token-wise attention by large margins, e.g., +1.5% to multi-head attention on CNN testset. It justifies our motivation to interact at fine-grained level.

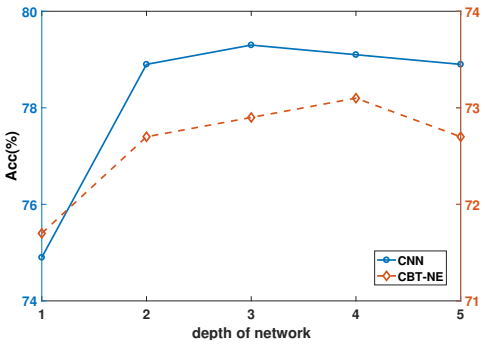

Figure 4: Impact of hops on Acc performance.

Table 3: Results with various attention on single-hop model.

Attention	Acc on Testsets (%)	
	CNN	Daily Mail
Additive	77.5	80.9
Multiplicative	77.9	81.2
Multi-Head	77.8	81.4
Bit-wise	**79.3**	**82.7**

Case Study

We try to interpret the interactive aligning across multiple latent spaces in a straight forward way. We choose the **bilinear scoring function** which computes relevance scores at token level, thus we can visualize **attention weight distribution** over the context tokens with a heat map in Figure 5. Each row represents a specific space and each column represents a context word[4]. Note that, all attention weight vectors are all extracted from the **single-hop** structure.

This way of visualization gives hints on which aspect of semantics is reflected in each subspace. Some latent spaces, e.g., space I, III, V highlight *characters* associated with the event (i.e., entity0 and entity5), while others like II, IV focus on some other aspects of the event, such as *time, location* and *action*. Space VI shows the *average aggregation* of all the aspects of the event and seems messy.

Another way of visualization can be achieved by averaging the attention vectors across different spaces. Note that, there is no need to apply *softmax* to it, since averaging attentions still keep the normalization condition. Figure 6 yields a general view of which token is mostly focused on during the query-context interaction. It can be observed that the interaction phase takes the characters (i.e., entity0 and entity5) more into account, which is consistent with the query type.

Conclusion

In this paper, we propose the Enhanced Attentive Reader (EA Reader) to answer cloze-style questions. The reader has three special virtues: (1) it performs multiple spaces of attention, each of which reveals some aspects of semantic interactions; (2) it measures interactions at fine-grained bit level; (3) it equips the iterative inference process with a novel memory update rule, which is vital to filter out noisy informantion. Our model yields a significant performance gain on several large-scale benchmark datasets over competitive baselines. We demonstrate that both multi-space mechanism and multi-hop architecture are integral parts of EA Reader. We also show empirically that the bit-wise attention outperforms token-wise attention. In the future, we will explore ways to solve span-extractive question answering tasks, which includes designing a novel memory-based answer pointing mechanism.

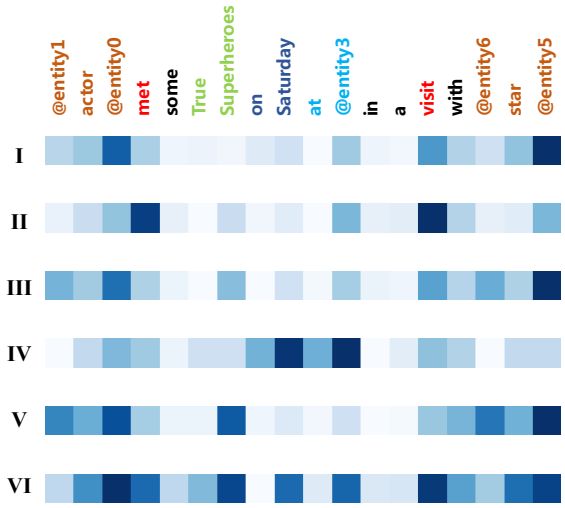

Figure 5: A visualized example of interactive aligning.

Figure 6: A visualized example of average importance score.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and helpful suggestions. This work is supported by NSFC under Grant No. 61532001, and MOE-ChinaMobile program under Grant No. MCM20170503.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. *ICLR 2015*.

[4]**Corresponding query:** *@entity0, @placeholder visit young patients at @entity3*. **Concrete Meanings** for **entity1:** Captain America; **entity0:** Chris Evans; **entity3:** Seattle Children's Hospital; **entity6:** Guardians of the Galaxy; **entity5:** Chris Pratt.

Chen, D.; Bolton, J.; and Manning, C. D. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 2358–2367.

Cheng, J.; Dong, L.; and Lapata, M. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 551–561.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS 2014 Deep Learning and Representation Learning Workshop*.

Cui, Y.; Chen, Z.; Wei, S.; Wang, S.; Liu, T.; and Hu, G. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 593–602.

Dhingra, B.; Liu, H.; Yang, Z.; Cohen, W.; and Salakhutdinov, R. 2017. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 1832–1846.

Hermann, K. M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc. 1693–1701.

Hill, F.; Bordes, A.; Chopra, S.; and Weston, J. 2015. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*.

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

Hu, M.; Peng, Y.; and Qiu, X. 2017. Reinforced mnemonic reader for machine comprehension. *CoRR, abs/1705.02798*.

Huang, H.-Y.; Zhu, C.; Shen, Y.; and Chen, W. 2017. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. *arXiv preprint arXiv:1711.07341*.

Kadlec, R.; Schmid, M.; Bajgar, O.; and Kleindienst, J. 2016. Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 908–918.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kumar, A.; Irsoy, O.; Ondruska, P.; Iyyer, M.; Bradbury, J.; Gulrajani, I.; Zhong, V.; Paulus, R.; and Socher, R. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, 1378–1387.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1412–1421.

Munkhdalai, T., and Yu, H. 2017. Neural semantic encoders. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 1, 397. NIH Public Access.

Paulus, R.; Xiong, C.; and Socher, R. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2017. Bidirectional attention flow for machine comprehension. *ICLR 2017*.

Shen, Y.; Huang, P.-S.; Gao, J.; and Chen, W. 2017. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1047–1055. ACM.

Shen, T.; Zhou, T.; Long, G.; Jiang, J.; Pan, S.; and Zhang, C. 2018. Disan: Directional self-attention network for rnn/cnn-free language understanding. *AAAI 2018*.

Sordoni, A.; Bachman, P.; Trischler, A.; and Bengio, Y. 2016. Iterative alternating neural attention for machine reading. *arXiv preprint arXiv:1606.02245*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.

Sukhbaatar, S.; szlam, a.; Weston, J.; and Fergus, R. 2015. End-to-end memory networks. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc. 2440–2448.

Tan, C.; Wei, F.; Yang, N.; Lv, W.; and Zhou, M. 2017. S-net: From answer extraction to answer generation for machine reading comprehension. *arXiv preprint arXiv:1706.04815*.

Trischler, A.; Ye, Z.; Yuan, X.; Bachman, P.; Sordoni, A.; and Suleman, K. 2016. Natural language comprehension with the epireader. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 128–137.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 6000–6010.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, 2692–2700.

Wang, S., and Jiang, J. 2016. Machine comprehension using match-lstm and answer pointer. *ICLR 2016*.

Xiong, C.; Zhong, V.; and Socher, R. 2017. Dynamic coattention networks for question answering. *ICLR 2017*.