The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

# A Unified Model for Document-Based Question Answering Based on Human-Like Reading Strategy

**Weikang Li, Wei Li, Yunfang Wu***
Key Laboratory of Computational Linguistics, Peking University, MOE, China
{wavejkd,liweitj47,wuyf}@pku.edu.cn

## Abstract

Document-based Question Answering (DBQA) in Natural Language Processing (NLP) is important but difficult because of the long document and the complex question. Most of previous deep learning methods mainly focus on the similarity computation between two sentences. However, DBQA stems from the reading comprehension in some degree, which is originally used to train and test people's ability of reading and logical thinking. Inspired by the strategy of doing reading comprehension tests, we propose a unified model based on the human-like reading strategy. The unified model contains three major encoding layers that are consistent to different steps of the reading strategy, including the basic encoder, combined encoder and hierarchical encoder. We conduct extensive experiments on both the English WikiQA dataset and the Chinese dataset, and the experimental results show that our unified model is effective and yields state-of-the-art results on WikiQA dataset.

## Introduction

Document-based Question Answering (DBQA) is an important issue in natural language processing(NLP). Given a document and a question related to the document, the system is required to give an answer for the question. The answer could be a word, a text span or a sentence extracted from the document. Table 1 gives an example of DBQA. Recently, more and more researches have focused on this challenging problem.

A lot of achievements have been achieved via deep learning models, which obtain better performances than traditional machine learning methods. Inspired by the great success of deep learning methods in voice and image recognition, researchers have adopted various ways to solve the problem of DBQA, including convolutional neural network (CNN) (Feng et al. 2015), recurrent neural network (RNN) (Tan et al. 2015), Attention-Way (Seo et al. 2016) and generative adversarial networks (GAN) (Wang et al. 2017). Many other ways have emerged to dig out more information to solve the problem of DBQA. Document summary could also be seen as an effective information in many NLP tasks. Choi et al. (2017) and Miller et al. (2016) used the most related sentences from document as document summary for the answer selection task.

However, the simple transfer between deep learning and DBQA is not so logical. In our opinion, DBQA is similar to the reading comprehension test, which is defined to test people's comprehension of a document. In their school years, students would do lots of reading comprehension tests. In this paper, we provide a solution for the problem to stimulate men's reading strategy of doing the tests. With the assumption, the detailed reading strategy is as follow:

- 1. Go over the document quickly to get a general understanding of the document;
- 2. Read the question carefully equipped with the general understanding of the document;
- 3. Go back to the document with the prior knowledge of question and get the right answer.

Such a reading strategy could be implemented by neural network models.

As we know, the document in reading comprehension tests usually has a title, which has an important impact on doing reading comprehension tests for people. Unfortunately, the title information is neglected by most researches on DBQA. In this paper, we use the title information (a natural document summary) as the general understanding of a document. As for the document without title, we make many attempts to get the general understanding of the document, by using the first sentence, the last sentence and training a LDA or LSA model to get the topic of a document. In addition, we have tried many ways to understand questions well given the general understanding of the document.

At the end, we propose a unified neural network model according to the human-like reading strategy above.

Our contributions in this paper can be summarized as follows:

- We propose a human-like reading strategy for DBQA task which is similar to the logic of students when they do the test of reading comprehension.
- Based on the reading strategy, we make a good combination of general understanding of both document and question.

| | |
|---|---|
| **Document Title** | *Uncle Sam* |
| **Document** | *J. M. Flagg 's 1917 poster, based on the original British Lord Kitchener poster of three years earlier, was used to recruit soldiers for both World War I and World War II. ...... Uncle Sam (initials U.S.) is a common national personification of the American government that, according to legend, came into use during the War of 1812 and was supposedly named for Samuel Wilson. It is not clear whether this reference is to Uncle Sam as a metaphor for the United States. ......* |
| **Question** | *what does uncle sam represent to the American people ?* |
| **Answer(Sentence)** | *Uncle Sam (initials U.S.) is a common national personification of the American government that, according to legend, came into use during the War of 1812 and was supposedly named for Samuel Wilson.* |
| **Answer(Span)** | *a common national personification of the American government.* |
| **Answer(Word)** | *national personification.* |

Table 1: An Outline of DBQA

- We propose a unified neural network model which is suitable for our reading strategy to tackle the problem of DBQA.

We conduct experiments on the English WikiQA dataset (Yang, Yih, and Meek 2015) and the Chinese DBQA dataset (Duan 2016). On the WikiQA dataset, our model obtains a MAP of 0.754, which outperforms the best previous method by 1.1 MAP points. On the Chinese DBQA dataset, our model gets comparable results without using any features.

## Related Work

Reading documents and being able to answer related questions by machines is a useful and meaningful issue. However, it is still an unsolved challenge. DBQA has several different answer types, as outlined in Table 1. Our work mainly focuses on the form in which answer is a whole sentence.

As we know, many NLP problems involve matching two or more sequences to make a decision. For DBQA, some researches also see this problem as matching two sequences (question and candidate answer) to decide whether a sentence from the document could answer the question.

In the field of sentence pairs matching, there have been various deep neural network models proposed. Two levels of matching strategies are considered: the first is converting the whole source and target sentence into embedding vectors of latent semantic spaces respectively, and then calculating similarity score between them; the second is calculating the similarity score among all possible local positions of source and target sentences, and then summarizing the local scores into the final similarity score.

Works using the first strategy include bag of words based methods (Wang et al. 2011) and CNN model (Arc-I) (Hu et al. 2014). Qiu and Huang (2015) applied a tensor transformation layer on CNN based embeddings to capture the interactions between question and answer more effectively. Long short-term memory (LSTM) network model (Palangi et al. 2016) are also explored in this problem. Works using the second strategy include DeepMatch (Lu and Li 2013) which incorporated latent topics to make the local matching structure sparse, Arc-II (Hu et al. 2014) which proposed a two dimensional CNN to extract local matching features. Besides, Pang et al. (2016) built hierarchical convolution layers on the word similarity matrix between sentences, and Yin and Schütze (2015) proposed MultiGranCNN to integrate multiple granularity levels of matching models. The Representation (MPSR) model (Wan et al. 2016) employed LSTM and interactive tensor to capture matching features with positional local context. For both levels of matching strategies, the ways of computing similarity between two sentences are similar. The most popular methods are cosine similarity (Tan et al. 2016), element-wise product (Seo et al. 2016) and tensor computation (Bowman et al. 2015).

As a task to train people's reading and understanding skills, DBQA is more complex, logical and skillful than a simple comparison of the similarity between two sentences. We will imitate people's reading strategy of doing reading comprehension tests via the neural network.

Attempts have also been made to study how people read. Masson (1983) conducted studies on how people answer questions by first skimming the document, identifying relevant parts, and carefully reading these parts to obtain an answer. Inspired by this observation, Golub et al. (2017) proposed a coarse-to-fine model for question answering. It first selects relevant sentences and then generates an answer. Different from their method, our work mainly focuses on people's reading strategy when doing reading comprehension tests.

Titles could be naturally used to obtain a general understanding of documents in people's reading, and summarization offers another way to get the general meaning of a document in the absence of a title. Usually, there are two ways to automatically summarize a document, including extractive summarization and abstractive summarization. There has been an increase interest in document summarization over the years.

Extractive summarization works with the method of finding the salient sentences in a document. The research of IBM laboratory (Luhn 1958) worked on the frequency of words in the text. Edmundson (1969) used the title words, core phrases, key concepts, position method, which are the surface level information. Gillick (2011) employed a classification function to categorize each sentence (sentence extraction) using a Naive-Bayes classifier. Hovy and Lin (1998)

| Steps | Explanations |
|---|---|
| $Step1$ | Using the title or a summary-like method to achieve the general understanding of a document. Using a RNN to encode the question and the summary respectively. |
| $Step2$ | Making a combination of the question's encoding and the summary's encoding to get a new representation of the question. |
| $Step3$ | Using a hierarchical RNN to encode the document, equipped with the new representation of the question and selecting the right answer. |

Table 2: The Reading Strategy

also studied on sentence position and tried to restructure the sentence extraction using the decision tree. Abstractive approach is different from extractive approach, which is more suitable for short documents. Ghosh et al. (2016) proposed contextual LSTM models for large scale NLP tasks, in which they put the summary information achieved by the HTM (Hierarchical Topic Model) into the RNN cell. Latent Semantic analysis (LSA) (Dumais 2004) and Latent Dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003) are two easy and effective ways to get a summary of documents, which are used in our work to achieve a general understanding of the document.

To the best of our knowledge, our neural network model is unique, which is inspired by men's reading strategy and could well incorporate the general meaning of a document. Also, extensive experiments conducted on the open DBQA data demonstrate that our model significantly improves the basic system and helps us get state-of-the-art results.

## Approach Overview

We propose a human-like reading strategy to tackle the DBQA problem, which could be conducted via the neural network. In this section, we firstly make a formalized definition of DBQA; then we detail our reading strategy and explain why it is logical and reasonable in this problem.

### Problem Setting

Given a training set of question-document-answer triples $\left(q^i, d^i, a^i\right)_{i=1}^{N}$, our goal is to learn a model that produces an answer $\widehat{a}$ for each question-document pair $(q, d)$. A document $d$ is a list of sentences $\left(s_1, s_2, ..., s_{|d|}\right)$. Table 1 shows a training example.

### The Reading Strategy

In our school years, we would do a mass of reading comprehension exercises, which could improve the ability of men's logic thinking. Here, we propose a neural network model to imitate human's reading strategy, as illustrated in Table 2.

First, we get the general understanding of a document by reading the title or using a summary-like method. A RNN is applied to encode the question and the title respectively. As we know, the question $q$ about a document $d$ is very close to the document. If we directly apply a RNN layer to encode it without the information of $d$, the understanding of $q$ would be too broad and emanative. Thus in the second step, we incorporate the hidden representation of the title into the question, posing a limitation to the understanding of $q$ and making the meaning more close to the document. We explore several methods including deep learning models and simple computations to combine both information. Thirdly, a document usually consists of many sentences, but the traditional single RNN could not capture the dependencies between sentences. We employ a hierarchical RNN structure to obtain the document level representation, equipped with the new question's encoding vector. At last, the model decides which sentence could answer the question.

## Model

We build a neural network model based on the human-like reading strategy. In this section, We firstly make an explanation about each component in our model, and then detail the unified model, as shown in Figure 1.

### Document Summary

When using deep learning techniques in some NLP tasks, it is important to take advantage of more plain texts in the original input process. Title is a natural summary of the document, which gives us a brief and general introduction to the content. However, not all public datasets of DBQA task have a title for the document, and so we need some other methods to dig out the summary information of a text.

When writing a document, people usually give an introduction of what to discuss at the beginning and make a conclusion of what have described in the end. Thus, we can use the first or last sentence as the summary of a document in some degree.

LDA and LSA are two easy and effective ways to get an overall topic of a document, although there are many ways used to summarize a document, including traditional machine learning methods and deep learning models.

We denote $t(d)$ as the general representation of a document, where $d$ is the plain text of the document. We explore several ways to get the overall meaning of a document, which are as follows:

$$t(d) = title \tag{1}$$

where $title$ is the original information of a document.

$$t(d) = document_1 \tag{2}$$

where $document_1$ denotes the first sentence of a document.

$$t(d) = document_{|d|} \tag{3}$$

where $document_{|d|}$ is the last sentence of a document.

$$t(d) = LDA(document) \tag{4}$$

where $LDA(document)$ means using the pre-trained LDA model to get the topic of a document.

$$t(d) = LSA(document) \tag{5}$$

where $LSA(document)$ means using the pre-trained LSA model to get the topic of a document.

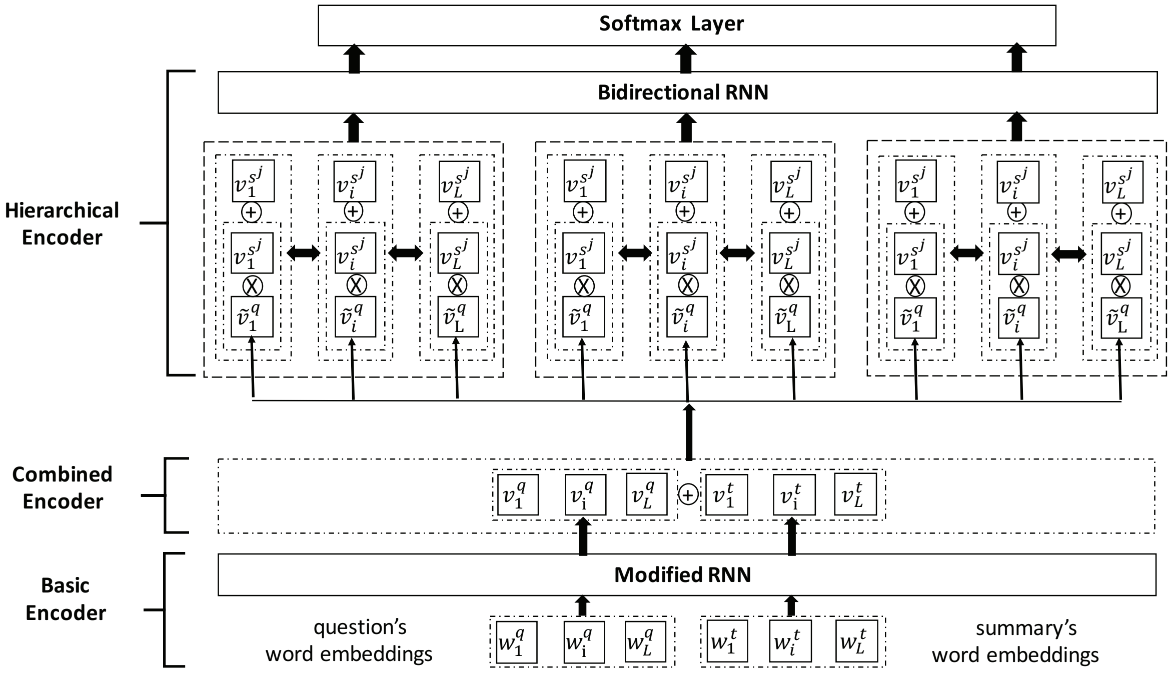

Figure 1: A Unified Model based on the Human-like Reading Strategy

## Basic Encoder

CNN and RNN have been used to encode the sentence in NLP tasks. Considering the time sequence of a sentence, RNN is widely used for the sentence's encoding. We adopt a modified version (Wang and Jiang 2016) of LSTM/GRU, in which only the input gates for remembering meaningful words are kept:

$$v^s = \sigma(W^i w^s + b^i \otimes e_{|s|}) \odot \tanh(W^u w^s + b^u \otimes e_{|s|}) \quad (6)$$

where $\odot$ is element-wise multiplication, and $W^i, W^u \in R^{l\times d}$ and $b^i, b^u \in R^l$ are parameters to be learned. The outer product $\left(\cdot \otimes e_{|s|}\right)$ produces a matrix or row vector by repeating the vector or scalar on the left for $|s|$ times. $|s|$ is the sentence's length. $w^s$ means a list of word embeddings of the sentence; $v^s$ is a list of hidden vectors of the sentence.

We use Equation 6 as the basic encoder for a single sentence, like a question, a summary of the document or a sentence from the document.

## Combined Encoder

The combination methods are used in two locations of our work. The first is to add the document's summary to the encoding of the question; the second is to add the question's encoding vector to the encoding of the document.

Given two sentences $s^x$ and $s^y$, we explore four combining methods in our work.

$$\tilde{v}_i^{s^x} = v_i^{s^x} + v_i^{s^y} \quad (7)$$

$$\tilde{v}_i^{s^x} = v_i^{s^x} \times v_i^{s^y} \quad (8)$$

where $i$ is the $i$-th word in the sentence, $v_i^x$ and $v_i^y$ represent the $i$-th word vector of $x$ and $y$ respectively, and $\tilde{v}_i^x$ is the $i$-th word vector of $x$ after combing with that of $y$. And in which $+$ means element-wise addition, $\times$ is element-wise multiplication.

$$\tilde{v}^{s^x} = concat(v^{s^x}, v^{s^y}) \quad (9)$$

which means concatenate $y$'s vector to the $x$'s vector.

$$G = softmax((W^g \cdot v^{s^x} + b^g \otimes e_{|s^x|})^T v^{s^y}) \quad (10)$$

$$\tilde{v}^{s^x} = v^{s^y} \cdot G \quad (11)$$

where $W^g \in R^{d\times d}$ and $b^g \in R^l$ are parameters to be learned, $G \in R^{|s^y|\times|s^x|}$ is the attention weight matrix, and $\tilde{v}^{s^x} \in R^{|s^x|\times d}$ are the attention-weighted vectors. The attention way we use is the similar to the work of Wang and Jiang (2016).

## Hierarchical Encoder

Although the RNN model is suitable for the encoding of a single sentence, it is hard to capture the long-range dependencies of a document. A single-directional LSTM suffers from the weakness of not making full use of the contextual information from forward and backward tokens.

As for a document, which consists of dozens of sentences, we provide a hierarchical encoder which contains the basic encoder, the combined encoder and the bidirectional LSTM layer. The basic encoder and combined encoder are firstly applied for the sentences of the document separately, and then a LSTM layer is used to encode each sentence again based on the output of the basic encoder. Besides, the LSTM

| | Institution | Date | Source | Has Document Title | Language |
|---|---|---|---|---|---|
| **WikiQA** | Georgia Institute of Technology | 2015 | Wikipedia | yes | English |
| **NLPCC2016** | The NLPCC Conference | 2016 | website's texts | no | Chinese |

Table 3: Dataset Description

| | Qs/Ds | Q-A Pairs |
|---|---|---|
| **Train Set** | 2118 | 20360 |
| **Development Set** | 296 | 2733 |
| **Test Set** | 633 | 6165 |

Table 4: Details of WikiQA

| | Qs/Ds | Q-A Pairs |
|---|---|---|
| **Train Set** | 8772 | 181882 |
| **Development Set** | – | – |
| **Test Set** | 5997 | 122531 |

Table 5: Details of NLPCC2016

layer is also used to capture contextual features among sentences, which could make the understanding of a document more coherent. The LSTM network we adopt is similar to that in Graves, Mohamed, and Hinton (2013). The formulas are as follows:

$$i_t = \sigma(W_i v(t) + U_i h(t-1) + b_i) \tag{12}$$

$$f_t = \sigma(W_f v(t) + U_f h(t-1) + b_f) \tag{13}$$

$$o_t = \sigma(W_o v(t) + U_o h(t-1) + b_o) \tag{14}$$

$$\tilde{C}_t = \tanh(W_c v(t) + U_c h(t-1) + b_c) \tag{15}$$

$$C_t = i_t * \tilde{C}_t + f_t * C_{t-1} \tag{16}$$

$$h_t = o_t * \tanh(C_t) \tag{17}$$

where $v(t)$ is a list of $d$-dimension vectors produced by the basic encoder and $h(t)$ means the hidden vector in the time stamp $t$. There are three gates (input $i$, forget $f$ and output $o$), and a cell memory vector $C_t$. $\sigma$ is the $sigmoid$ function. $W \in R^{l \times d}, U \in R^{l \times l}$ and $b \in R^{l \times 1}$ are the network parameters. $l$ is the length of a sequence.When used for a sentence, $v(t)$ is a list of word vectors; when used for a document, it means a list of sentence vectors.

### Unified Model

Based on the reading strategy of humans, we propose a unified model composed of the basic encoder, combined encoder and hierarchical encoder for the DBQA task, as shown in Figure 1.

Let $q$ denotes the question, $t$ denotes the summary of the document and $s$ denotes a sentence in the document. Firstly, we use the basic encoder to get vectors $(v_1^q, v_i^q, v_L^q$ and $v_1^t, v_i^t, v_L^t)$ respectively based on their word embeddings $(w_1^q, w_i^q, w_L^q$ and $w_1^t, w_i^t, w_L^t)$. Then, we use the combined encoder to get the question's vectors $(\tilde{v}_1^q, \tilde{v}_i^q, \tilde{v}_L^q)$, in which $t$'s information has been added. In Figure 1, $\oplus$ is the combining computations in our work.

As for the document, we also use the basic encoder to get each sentence's vector $(v_1^{s^j}, v_i^{s^j}, v_L^{s^j})$, in which $j$ is the sentence index in the document. In Figure 1, $\otimes$ means an attentive computation (Formula 10 and Formula 11), which combines a sentence's vectors $(\tilde{v}_i^{s^j})$ with the representation of question $(\tilde{v}_i^q)$ via an attention mechanism. After that, we also make an addition between $\tilde{v}_i^{s^j}$ and $v_i^{s^j}$ via the combined encoder. Then we use a bidirectional RNN layer to further encode in the sentence and document separately. Especially, the input of the document's RNN layer is the concatenation of two vectors from both directions of each sentence. Finally, we use a softmax layer to choose the answer sentence among every step's output of the document's RNN layer. The model is trained to minimize the cross-entropy loss function:

$$L(a, \tilde{a}) = -\frac{1}{N} \sum_{i \in N} a_i \log \tilde{a}_i \tag{18}$$

## Experiments

To demonstrate the effectiveness of our proposed model, we conduct experiments on the open DBQA datasets. We will describe the details of datasets, settings, results and analysis in our experiments.

### Datasets

There are lots of datasets for reading comprehension and DBQA task. Our work focuses on choosing a sentence as the answer of a question. In order to validate our model's generality, we use two open datasets to evaluate the performance. The first is English WikiQA (Yang, Yih, and Meek 2015), which is collected and annotated for research on open-domain question answering; the other is the Chinese DBQA task from NLPCC-ICCPOL 2016 Shared Task (Duan 2016), which is annotated by human annotators. WikiQA has a natural title for each document, but the Chinese DBQA doesn't have. Table 3 gives the description of the two datasets; Table 4 and 5 list the statistics distribution. In WikiQA, there are some questions which have no answer, we removed these questions and only kept questions that have answers. As for NLPCC2016, there is no development set, and so we divided 20% of the training set as our development data and the rest as our training data.

### Baselines

As for WikiQA dataset, we re-implemented $CA\text{-}Network$ proposed by Wang and Jiang (2016) as our baseline. We do not implement other baseline models but simply take the reported performance in the original paper.

- **IARNN-Occam**: this model adds regularization on the attention weights (Wang, Liu, and Zhao 2016).

| Method | Evaluation Methods | |
|---|---|---|
| | MAP | MRR |
| IARNN-Occam | 0.7341 | 0.7418 |
| IARNN-Gate | 0.7258 | 0.7394 |
| CNN-Cnt | 0.6520 | 0.6652 |
| ABCNN | 0.6921 | 0.7108 |
| CubeCNN | 0.7090 | 0.7234 |
| CA-Network | **0.7433** | **0.7545** |
| q-t | **0.7541** | **0.7659** |
| q-mul-t | 0.7056 | 0.7166 |
| q-add-t | **0.7429** | **0.7541** |
| qat | 0.7102 | 0.7246 |
| qat-add-q | **0.7424** | **0.7573** |
| q-fs | 0.7193 | 0.7308 |
| q-ls | 0.7304 | 0.7403 |
| q-LSA | 0.7247 | 0.7343 |
| q-LDA | **0.7441** | **0.7551** |

Table 6: Experiment Results on WikiQA

- **IARNN-Gate**: this model uses the representation of the question to build the GRU gates for each candidate answer (Wang, Liu, and Zhao 2016).
- **CNN-Cnt**: this model combines sentence representations produced by a convolutional neural network with the logistic regression (Yang, Yih, and Meek 2015).
- **ABCNN**: this model is an attention-based convolutional neural network (Yin et al. 2015).
- **CubeCNN**: this model builds a CNN on all pairs of word similarities (He and Lin 2016).

As for the Chinese DBQA dataset, we implemented a model $CA\text{-}Network$ based on the compare-aggregate network (Wang and Jiang 2016), in which we made a combination of Chinese character embeddings and word embeddings. For other baselines, we simply take the performance reported in the original paper, and they are:

- **CNN-Overlap**: it is the work of Fu, Qiu, and Huang (2016), which builds a CNN network by incorporating word overlap features. It is the best system at the shared task of NLPCC-ICCPOL 2016.
- **Hybrid-Way**: it is the work of Wu et al. (2016), which is based on feature engineering. It is the second best system at the campaign.

## Settings

The proposed models were implemented with TensorFlow (Abadi et al. 2016), and all experiments were conducted in a GPU cluster. We used the metric of accuracy on the development set to select the best epoch and best hyper-parameters and then applied them to the test data. We used the English and Chinese Wikipedia corpus to train the LDA and LSA models separately for two languages via Gensim [1], where the number of topics was set to 100. The word embeddings were pre-trained using word2vec (Mikolov et al. 2013), based on the training data of WikiQA for English and the training data of Chinese DBQA task for Chinese, respectively. The dimension of word embeddings was set to 300 in both languages. We used the Adam (Kingma and Ba 2014) optimizer to train our models. The learning rate was 0.001. Our models were trained in mini-batches (with batch size of 20), and the trained epochs was 20. The maximum length of the question and each sentence in the document was fixed to 200, and any tokens out of this range were discarded. The hidden vector size was set to 150 for a single RNN. Evaluation metrics we used are Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR).

| Method | Evaluation Methods | | |
|---|---|---|---|
| | MAP | MRR | Precision |
| CNN-Overlap | **0.8592** | **0.8586** | **0.7906** |
| Hybrid Way | 0.8269 | 0.8263 | 0.7385 |
| CA-Network | 0.8073 | 0.8082 | 0.7244 |
| Ours | **0.8443** | **0.8455** | **0.7708** |

Table 7: Experiment Results on Chinese Data

## Results and Analysis

Table 6 reports the experimental results on the WikiQA dataset. As the first choice, we use the natural title of a document as our summary information. The representation of the questions is obtained by:

- $q\text{-}t$ means the concatenation between the question and the title (Equation (10)).
- $q\text{-}mul\text{-}t$ means the element-wise multiplication between the question and the title (Equation (8)).
- $q\text{-}add\text{-}t$ means the element-wise addition between the question and the title (Equation (7)).
- $qat$ means the weighted-sum vector via an attention mechanism (Equation (10)(11)).
- $qat\text{-}add\text{-}q$ means the element-wise addition between a new question with the attention to the title and the original question.

The experimental results show that our combination methods that incorporate the title's information into the question's encoding are effective, including $q\text{-}t$, $q\text{-}add\text{-}t$ and $qat\text{-}add\text{-}q$, which all achieve state-of-the-art performances. Specifically, $q\text{-}t$ yields nearly 1.1% improvement on both MAP and MRR over the previously best results on the challenging dataset.

Further, we explore some other methods that automatically produce a summary to the content of a document, and use the concatenation computation between the question and the summary to update the representation of question, because the concatenation gets the best performance in the combination of the question and title.

- $fs$ is the first sentence of the document.

[1] A python package, https://pypi.python.org/pypi/gensim.

| | |
|---|---|
| **Title** | Cardiovascular Disease |
| **Document** | Cardiovascular disease (also called heart disease) is a class of diseases that involve the heart or blood vessels (arteries, capillaries, and veins). ...... **The causes of cardiovascular disease are diverse but atherosclerosis and/or hypertension are the most common.** ...... |
| **Question** | what causes heart disease? |
| $Ours$ | The causes of cardiovascular disease are ...... |
| $CA$-$Network$ | Cardiovascular disease (also called heart ...... |

Table 8: Case Study

- $ls$ is the last sentence of the document.
- $LDA$ is the topic prediction of the document based on the pre-trained LDA model.
- $LSA$ is the topic prediction of the document based on the per-trained LSA model.

we find that $q$-$LDA$'s performance is a little higher than the published best results, demonstrating that the topic prediction pre-trained by the LDA model could be a valid substitution of the title of a document, which provides an effective solution to the documents without a natural title in the DBQA task. Actually, according to our observation, there is a close range between the natural title and the LDA summary in the hidden vector space, although they are different in words.

Different from the WikiQA data, the documents in the Chinese DBQA data of NLPCC2016 do not have titles. The experimental results are illustrated in Table 7, where our model in this paper is denoted as $ours$, which is the same as $q$-$LDA$ on the WikiQA dataset. Considering the meaning of Chinese characters, we make a combination of Chinese character embeddings and word embeddings in our model, which is the only difference between the English and Chinese models.

Our model obtains a MAP of 84.43%, which outperforms the re-implemented $CA$-$Network$ by 3.7 MAP points. Besides, our model has a very competitive performance compared with the best results of $CNN$-*overlap*, which is a CNN model combined with the word overlap features and so can be seen as the feature engineering work in some degree.

## Case Study

To make a further analysis of our model, we give a case study in Table 8, which lists the results of our best $q$-$t$ model comparing the re-implemented $CA$-$Network$ on the WikiQA dataset. When answering the question, our model takes into account the summary representation "Cardiovascular Disease" which brings important information to the prediction, while the $CA$-$Network$ neglects the title information and makes a wrong prediction.

## Conclusion

Based on human's strategy of doing reading comprehension tests, we propose a unified model to tackle the DBQA problem, which mainly consists of the basic encoder, combined encoder and hierarchical encoder. The experimental results verify the effectiveness of our proposed method in both English and Chinese datasets. In the WikiQA data, our model outperforms the previous state of the art results by 1.1 MAP points. However, we still haven't fully dug out the strategies or skills about how people do reading comprehension tests. In the future, we would like to explore more about people's reading strategy to improve our model and test its effectiveness on other tasks.

## Acknowledgments

This work is supported by National High Technology Research and Development Program of China (2015AA015403) and National Natural Science Foundation of China (61371129, 61773026). The corresponding author of this paper is Yunfang Wu.

## References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan):993–1022.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Choi, E.; Hewlett, D.; Uszkoreit, J.; Polosukhin, I.; Lacoste, A.; and Berant, J. 2017. Coarse-to-fine question answering for long documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 209–220.

Duan, N. 2016. Overview of the nlpcc-iccpol 2016 shared task: Open domain chinese question answering. In *International Conference on Computer Processing of Oriental Languages*, 942–948. Springer.

Dumais, S. T. 2004. Latent semantic analysis. *Annual review of information science and technology* 38(1):188–230.

Edmundson, H. P. 1969. New methods in automatic extracting. *Journal of the ACM (JACM)* 16(2):264–285.

Feng, M.; Xiang, B.; Glass, M. R.; Wang, L.; and Zhou, B. 2015. Applying deep learning to answer selection: A study and an open task. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, 813–820. IEEE.

Fu, J.; Qiu, X.; and Huang, X. 2016. Convolutional deep neural networks for document-based question answering. In

*International Conference on Computer Processing of Oriental Languages*, 790–797. Springer.

Ghosh, S.; Vinyals, O.; Strope, B.; Roy, S.; Dean, T.; and Heck, L. 2016. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*.

Gillick, D. J. 2011. *The elements of automatic summarization*. University of California, Berkeley.

Golub, D.; Huang, P.-S.; He, X.; and Deng, L. 2017. Two-stage synthesis networks for transfer learning in machine comprehension. *arXiv preprint arXiv:1706.09789*.

Graves, A.; Mohamed, A.-r.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, 6645–6649. IEEE.

He, H., and Lin, J. J. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *HLT-NAACL*, 937–948.

Hovy, E., and Lin, C.-Y. 1998. Automated text summarization and the summarist system. In *Proceedings of a workshop on held at Baltimore, Maryland: October 13-15, 1998*, 197–214. Association for Computational Linguistics.

Hu, B.; Lu, Z.; Li, H.; and Chen, Q. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, 2042–2050.

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

Lu, Z., and Li, H. 2013. A deep architecture for matching short texts. In *Advances in Neural Information Processing Systems*, 1367–1375.

Luhn, H. P. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development* 2(2):159–165.

Masson, M. E. 1983. Conceptual processing of text during skimming and rapid sequential reading. *Memory & Cognition* 11(3):262–274.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Miller, A.; Fisch, A.; Dodge, J.; Karimi, A.-H.; Bordes, A.; and Weston, J. 2016. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*.

Palangi, H.; Deng, L.; Shen, Y.; Gao, J.; He, X.; Chen, J.; Song, X.; and Ward, R. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 24(4):694–707.

Pang, L.; Lan, Y.; Guo, J.; Xu, J.; Wan, S.; and Cheng, X. 2016. Text matching as image recognition. In *AAAI*, 2793–2799.

Qiu, X., and Huang, X. 2015. Convolutional neural tensor network architecture for community-based question answering. In *IJCAI*, 1305–1311.

Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Tan, M.; Santos, C. d.; Xiang, B.; and Zhou, B. 2015. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

Tan, M.; dos Santos, C. N.; Xiang, B.; and Zhou, B. 2016. Improved representation learning for question answer matching. In *ACL (1)*.

Wan, S.; Lan, Y.; Guo, J.; Xu, J.; Pang, L.; and Cheng, X. 2016. A deep architecture for semantic matching with multiple positional sentence representations. In *AAAI*, 2835–2841.

Wang, S., and Jiang, J. 2016. A compare-aggregate model for matching text sequences. *arXiv preprint arXiv:1611.01747*.

Wang, B.; Liu, B.; Wang, X.; Sun, C.; and Zhang, D. 2011. Deep learning approaches to semantic relevance modeling for chinese question-answer pairs. *ACM Transactions on Asian Language Information Processing (TALIP)* 10(4):21.

Wang, J.; Yu, L.; Zhang, W.; Gong, Y.; Xu, Y.; Wang, B.; Zhang, P.; and Zhang, D. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. *arXiv preprint arXiv:1705.10513*.

Wang, B.; Liu, K.; and Zhao, J. 2016. Inner attention based recurrent neural networks for answer selection. In *ACL (1)*.

Wu, F.; Yang, M.; Zhao, T.; Han, Z.; Zheng, D.; and Zhao, S. 2016. A hybrid approach to dbqa. In *International Conference on Computer Processing of Oriental Languages*, 926–933. Springer.

Yang, Y.; Yih, W.-t.; and Meek, C. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *EMNLP*, 2013–2018.

Yin, W., and Schütze, H. 2015. Multigrancnn: An architecture for general matching of text chunks on multiple levels of granularity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, 63–73.

Yin, W.; Schütze, H.; Xiang, B.; and Zhou, B. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*.