Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Hybrid embedding and joint training of stacked encoder for opinion question machine reading comprehension[*]

Xiang-zhou HUANG[†], Si-liang TANG[†], Yin ZHANG[†‡], Bao-gang WEI[^]

College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China

[†]E-mail: huangxiangzhou@zju.edu.cn; siliang@zju.edu.cn; yinzh@zju.edu.cn

Received Oct. 19, 2019; Revision accepted Mar. 16, 2020; Crosschecked Aug. 10, 2020

Abstract: Opinion question machine reading comprehension (MRC) requires a machine to answer questions by analyzing corresponding passages. Compared with traditional MRC tasks where the answer to every question is a segment of text in corresponding passages, opinion question MRC is more challenging because the answer to an opinion question may not appear in corresponding passages but needs to be deduced from multiple sentences. In this study, a novel framework based on neural networks is proposed to address such problems, in which a new hybrid embedding training method combining text features is used. Furthermore, extra attention and output layers which generate auxiliary losses are introduced to jointly train the stacked recurrent neural networks. To deal with imbalance of the dataset, irrelevancy of question and passage is used for data augmentation. Experimental results show that the proposed method achieves state-of-the-art performance. We are the biweekly champion in the opinion question MRC task in Artificial Intelligence Challenger 2018 (AIC2018).

Key words: Machine reading comprehension; Neural networks; Joint training; Data augmentation
https://doi.org/10.1631/FITEE.1900571 **CLC number:** TP391.1

1 Introduction

Artificial intelligence (AI) has experienced over 60 years of continuous development and changed the world (Pan, 2016). Teaching machines to read and comprehend is a vital part of AI. Machine reading comprehension (MRC) is a task of answering questions by understanding corresponding passages. It is a key goal in natural language processing (NLP). The MRC task has attracted a lot of attention in recent years and there are already several large-scale datasets released, including MCTest (Richardson et al., 2013), CNN/Daily Mail (Hermann et al., 2015), WikiQA (Yang et al., 2015), Stanford Question Answering dataset (SQuAD) (Rajpurkar et al., 2016), Microsoft MAchine Reading COmprehension dataset (MS-MARCO) (Bajaj et al., 2016), TriviaQA (Joshi et al., 2017), and DuReader (He W et al., 2018). SQuAD (Rajpurkar et al., 2016) is one of the most famous MRC datasets. It consists of 100 000+ questions posed by crowdworkers on Wikipedia articles, where the answer to each question is a segment of text from the corresponding passage. Another larger MRC dataset, MS-MARCO (Bajaj et al., 2016), consists of 1 000 000+ anonymized questions sampled from Bing's search query logs. In MS-MARCO, answers are human-generated and may not be the exact span from the corresponding passage like SQuAD. Most of the MRC datasets are in English, including SQuAD and MS-MARCO. Only a few MRC datasets are designed in Chinese, and DuReader (He W et al., 2018) is one of them. DuReader collects questions from

[‡] Corresponding author

[^] Deceased

[*] Project supported by the China Knowledge Centre for Engineering Sciences and Technology (No. CKCEST-2019-1-12) and the National Natural Science Foundation of China (No. 61572434)

ORCID: Xiang-zhou HUANG, https://orcid.org/0000-0001-8870-4341; Yin ZHANG, https://orcid.org/0000-0001-6986-4227

Baidu Search, while the answers are manually generated. However, all the passages in DuReader are from the same community question answering website Baidu Zhidao (http://zhidao.baidu.com/) and this limits it.

AI Challenger 2018 (AIC2018) is an international AI competition. Opinion question machine reading comprehension (OQMRC) is the main track in AIC2018. Organizers present the largest Chinese dataset so far for the OQMRC task, containing 300 000 samples. All the samples are from real-world community question answering websites including Sogou Ask (http://wenwen.sogou.com/), Baidu Zhidao (http://zhidao.baidu.com/), Zhihu (http://www.zhihu.com/), and Sina Ask (http://iask.sina.com.cn/), making the AIC2018 OQMRC dataset much more realistic.

Fig. 1 shows two examples of the OQMRC dataset. Every question contains multiple options, and the answer is not only restricted to spans of the corresponding passage like SQuAD, but also about an opinion which needs to be deduced from supporting evidence.

Example 1

Question: 维生素C可以长期吃吗
Can I take vitamin C tablets for a long period

Passage: 每天吃的维生素的量没有超过推荐量的话是没有太大问题的。
It's not a problem if you take vitamins everyday following the doctor's advice.

Options: 1. 是
Yes
2. 否
No
3. 无法确定
Unidentified

URL: https://wenwen.sogou.com/z/q143317203.htm

Example 2

Question: 深圳和广州哪个离北京远
Which is farther from Beijing, Shenzhen or Guangzhou

Passage: 深圳比广州更靠南，我每次回北京都要经过广州。
Shenzhen is farther south than Guangzhou. Every time I go back to Beijing, I have to go through Guangzhou.

Options: 1. 深圳
Shenzhen
2. 广州
Guangzhou
3. 无法确定
Unidentified

URL: https://zhidao.baidu.com/question/427258407

Fig. 1 Examples from the AIC2018 OQMRC dataset

In this study, a novel method based on neural networks is presented to address the OQMRC task. The main contributions of this work can be summarized as follows:

1. During the competition, external data are not allowed and only a limited corpus can be used to pre-train embedding. To tackle this problem, part-of-speech (POS) tags are combined to enrich the semantic representation of questions and passages.

2. A joint training method is introduced to train stacked long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997). In the proposed model, the passage is encoded with four bi-directional LSTM layers. The 4th LSTM layer generates the main loss, and the other three layers generate the auxiliary losses. In the training process, the weights of the auxiliary losses are adjusted and finally decrease to zero. These auxiliary losses produce a better model performance.

3. To deal with data imbalance in the competition, irrelevancy of question and passage is used for data augmentation, and a total of 50 000+ new samples with label unidentified are generated. These new samples are added to the training set, and a higher accuracy is achieved in experiments.

2 Related works

Remarkable progress has been made since various MRC datasets were released. Neural networks achieve promising results in the MRC task and most related works are based on them. Hermann et al. (2015) first introduced an attention mechanism into the MRC task, which soon became a dominating model. Wang SH and Jiang (2016) used match-LSTM to build question-aware passage representation, and predicted answer spans in passage with pointer networks (Vinyals et al., 2015). Seo et al. (2016) introduced a bi-directional attention flow (BIDAF) network, a multi-stage hierarchical process which represents the context at different levels of granularity, and used BIDAF to obtain query-aware context representation. Wang W et al. (2017) introduced R-NET, which matches questions and passages with gated attention-based recurrent networks and uses a self-matching attention mechanism to refine passage representation. Yu et al. (2018) proposed QANet, whose encoder consists exclusively of convolution and self-attention but not recurrent networks. Devlin et al. (2018) introduced a bi-directional encoder representation from transformers (BERT), which was designed to pre-train deep bi-directional representations in all layers. The pre-trained BERT can be fine-tuned with a simple additional output layer for a wide range of tasks including MRC. Table 1 shows a comparative review of these works.

However, these related models focus mainly on English MRC datasets and none of them has been introduced to a formal competition with a limited corpus. On the other hand, the OQMRC task in AIC2018, as described in the introduction, is a formal competition channel within a Chinese MRC dataset. To accomplish this task, we propose an approach based on neural networks. Experimental results show that the proposed method outperforms other competing systems. We also won the biweekly championship in the competition.

3 Task definition

Most MRC datasets have emphasized span-selection methods with pointer networks (Vinyals et al., 2015). Such methods are appropriate when answers are facts or entities in passages. However, they cannot work for an opinion question whose answer needs to be deduced from multiple sentences in a passage.

To tackle this issue, we formulate the OQMRC task as a classification problem. To classify the answers into different opinion polarities, regular expressions are used to convert every question to a statement. For example, the question “Can I take vitamin C tablets for a long period” in Fig. 1 can be converted to a statement “I can take vitamin C tablets for a long period.” Then the original question answering task can be considered as a classification problem to determine if the converted statement can be inferred from its corresponding passage.

Table 2 lists the symbols used in this study. Formally, each sample in the dataset is represented as a triple (Q, P, c), where $Q = \{\boldsymbol{x}_i^{\mathrm{q}}\}_{i=1}^{N_{\mathrm{q}}}$ is the question with length N_{q}, $P = \{\boldsymbol{x}_j^{\mathrm{p}}\}_{j=1}^{N_{\mathrm{p}}}$ is the passage with length N_{p}, and $c \in \{1, 2, 3\}$ is the opinion label to represent whether the converted statement is true, false, or unidentified (in the case of word-segmentation, N_{q} and N_{p} represent the numbers of words in question and passage, respectively). Thus, the classifier is trained with the corresponding opinion labels but not the original answers.

4 Our approach

Fig. 2 gives an overview of the proposed approach based on neural networks. First, hybrid tagging embedding is pre-trained to represent the question and passage. Then in the encoding layer, the question and passage are processed by bi-directional

Table 2 Symbols used in this study*

Symbol	Definition
Q	A question in the sample
P	A passage in the sample
c	A candidate opinion in the sample
N_{q}	Length of the question
N_{p}	Length of the passage
W_{C}	Segmented word list of a corpus
P_{C}	POS tags of W_{C}
N_{c}	Total number of words in a corpus
T_{C}	Produced hybrid tags
F_{m}	Minimum word-frequency
$\oplus$	Operator to concatenate strings
$\hat{R}$	Label string denoting a certain rare word
$\boldsymbol{x}_i^{\mathrm{q}}$, $\boldsymbol{x}_j^{\mathrm{p}}$	The i^{th} word in a question and the j^{th} word in a passage, respectively
L^k	Objective function corresponding to the k^{th} stacked bi-directional LSTM

* Symbols used only in a single section will be defined where they appear and are not included in this table

Table 1 Comparative review of related works

Methodology	Main contribution
Attentive reader (AR) (Hermann et al., 2015)	First introduced an attention mechanism into the MRC task
Match-LSTM (Wang SH and Jiang, 2016)	Built question-aware passage representation and first used pointer networks
BIDAF (Seo et al., 2016)	Used BIDAF to obtain query-aware context representation
R-NET (Wang W et al., 2017)	Matched the question and passage with gated attention-based recurrent networks and used the self-matching attention mechanism
QANet (Yu et al., 2018)	Adopted convolution networks and self-attention
BERT (Devlin et al., 2018)	Pre-trained deep bi-directional representations from unlabeled text by jointly conditioning on both left and right contexts in all layers

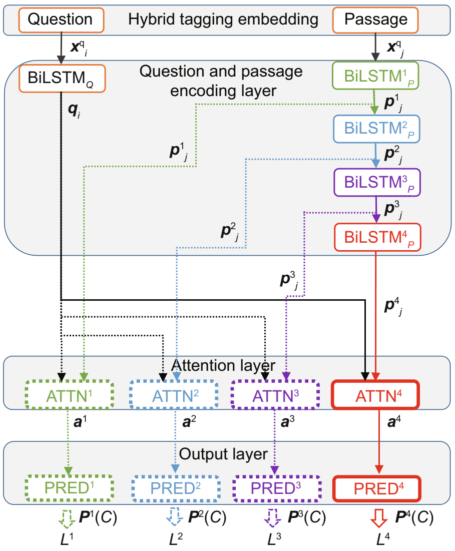

Fig. 2 Overview of the proposed approach based on neural networks

LSTMs separately. Outputs of every stacked LSTM layer are obtained and incorporated into question representation by the attention mechanism. As shown in the dashed parts in Fig. 2, extra LSTM, attention, and output layers are introduced to generate auxiliary losses, making the parameters of the model obtain more updates during the training process.

4.1 Hybrid embedding

Embedding has been widely used in the MRC task. However, according to the competition rules, external data are not allowed. Thus, the official OQMRC dataset provided in the competition is the only corpus for pre-training embedding. The common methods which pre-train word- or character-level embedding independently do not work well with such a small-scale corpus. To tackle this problem, we combine POS tags to pre-train hybrid embedding, which contains more semantic information and can represent questions and passages better.

The detailed process is illustrated in Algorithm 1. First, all the passages and questions are joined in the training set one by one as the corpus for later pre-training. Second, the official NLP tools provided in the competition are used to pre-process the corpus. After that, the segmented word list of the corpus $W_{\mathrm{C}}=\{w_n\}_{n=1}^{N_{\mathrm{c}}}$ and the corresponding POS tags $P_{\mathrm{C}}=\{p_n\}_{n=1}^{N_{\mathrm{c}}}$ are obtained, where N_{c} denotes the total number of words. Then W_{C} and P_{C} are combined to produce hybrid tags $T_{\mathrm{C}}=\{t_n\}_{n=1}^{N_{\mathrm{c}}}$ for every segmented word. Let F_{m} denote the minimum word-frequency and "$\oplus$" the operator to concatenate strings. For a word w_n, its hybrid tag t_n is set to $\{w_n\oplus p_n\}$ if its frequency is higher than F_{m}; otherwise, the corresponding POS tag p_n is set to a special label string $\hat{R}$ which denotes certain rare words, and its hybrid tag t_n is set to $\{p_{n-1}\oplus\hat{R}\oplus p_{n+1}\}$.

The open-source tool word2vec (https://pypi.org/project/word2vec/) (Mikolov et al., 2013a, 2013b) is used to pre-train the hybrid embedding, and the produced hybrid tags $T_{\mathrm{C}}=\{t_n\}_{n=1}^{N_{\mathrm{c}}}$ are the inputs. After pre-training, every unique hybrid tag is mapped to a d-dimensional embedding. Every word in the questions and passages is represented by its corresponding hybrid embedding.

Algorithm 1 Corpus tagging

```
Input: segmented word list of a corpus ($W_{\mathrm{C}}=\{w_n\}_{n=1}^{N_{\mathrm{c}}}$), POS tags of a segmented corpus ($P_{\mathrm{C}}=\{p_n\}_{n=1}^{N_{\mathrm{c}}}$), and the minimum word-frequency ($F_{\mathrm{m}}$)
Output: hybrid tag for every segmented word in the corpus ($T_{\mathrm{C}}=\{t_n\}_{n=1}^{N_{\mathrm{c}}}$)
$D=\{\ \}$ // Initialize an empty dictionary
for $w_n$ in $W_{\mathrm{C}}$ do
    if $w_n$ not in $D$ then
        $D[w_n]=1$ // Save a new word-frequency
    else
        $D[w_n]\mathrel{+}=1$ // Count the word-frequency
    end if
end for
$T_{\mathrm{C}}=[\ ]$
$P_{\mathrm{C}}$.insert($\hat{R}, 0$)
$P_{\mathrm{C}}$.insert($\hat{R}, -1$)
for $w_n$ in $W_{\mathrm{C}}$ do
    if $w_n$ in $D$ and $D[w_n]>=F_{\mathrm{m}}$ then
        $T_{\mathrm{C}}$.append($w_n\oplus p_n$)
    else
        $T_{\mathrm{C}}$.append($p_{n-1}\oplus\hat{R}\oplus p_{n+1}$)
        $p_n=\hat{R}$
    end if
end for
Return $T_{\mathrm{C}}$
```

4.2 Question and passage encoding layer

Let $\boldsymbol{x}_i^{\mathrm{q}}\in\mathbb{R}^d$ and $\boldsymbol{x}_j^{\mathrm{p}}\in\mathbb{R}^d$ denote the i^{th} $(1\le i\le N_{\mathrm{q}})$ word in the question and the j^{th} $(1\le j\le N_{\mathrm{p}})$ word in the passage, respectively. A bi-directional LSTM is used in the question encoding layer to produce a new representation $\boldsymbol{q}_i\in\mathbb{R}^d$ of $\boldsymbol{x}_i^{\mathrm{q}}$. Consider that the average length of passages is larger than that of questions in the dataset, not one but K $(K>1)$ stacked bi-directional LSTMs are used in the passage encoding layer. Let $\boldsymbol{p}_j^k \in \mathbb{R}^d$ denote the output of the k^{th} $(1 \leq k \leq K)$ stacked bi-directional LSTM BiLSTM_P^k. Then the new representations of question and passage are given as follows:

$$\boldsymbol{q}_i = \text{BiLSTM}_Q(\boldsymbol{q}_{i-1}, \boldsymbol{x}_i^{\text{q}}), \tag{1}$$

$$\boldsymbol{p}_j^1 = \text{BiLSTM}_P^1(\boldsymbol{p}_{j-1}^1, \boldsymbol{x}_j^{\text{p}}), \tag{2}$$

$$\boldsymbol{p}_j^k = \text{BiLSTM}_P^k(\boldsymbol{p}_{j-1}^k, \boldsymbol{p}_j^{k-1}). \tag{3}$$

Note that the first stacked bi-directional LSTM BiLSTM_P^1 takes the representation of the original passage $\boldsymbol{x}_j^{\text{p}}$ as input. Shown as the dashed parts in Fig. 2, all the k^{th} $(k < K)$ stacked bi-directional LSTMs and their following connected layers are used only during training but not in the prediction process.

4.3 Attention layer

The rise of attention mechanisms has greatly boosted the performance of many NLP tasks (Zhuang et al., 2017). Following previous works, question-to-passage attention is used to incorporate passage information into question representation. Specifically, the attention mechanism is applied to $\boldsymbol{q}_i$ with every output $\boldsymbol{p}_j^k$ of stacked bi-directional LSTMs in the passage encoding layer. Note that both $\boldsymbol{q}_i$ and $\boldsymbol{p}_j^k$ are d-dimensional, and simple dot attention is chosen; thus, no extra parameters are introduced in this layer. We also try out other attention ways or their combination (Tan et al., 2018), but find no significant difference or even worse performance. Let "$\odot$" denote the operation of dot product and ATTN^k the attention function corresponding to the k^{th} stacked bi-directional LSTM. Then the attention result $\boldsymbol{a}^k \in \mathbb{R}^d$ is computed as

$$\begin{aligned}\boldsymbol{a}^k &= \text{ATTN}^k(\boldsymbol{q}_i, \boldsymbol{p}_j^k) \\ &= \sum_{j=1}^{N_{\text{p}}} \frac{\exp\left(\max\limits_{1 \leq i \leq N_{\text{q}}} e_{ij}^k\right)}{\sum_{j=1}^{N_{\text{p}}} \exp(e_{ij}^k)} \boldsymbol{p}_j^k,\end{aligned} \tag{4}$$

where

$$e_{ij}^k = \boldsymbol{q}_i \odot \boldsymbol{p}_j^k. \tag{5}$$

4.4 Output layer

As mentioned in Section 3, the problem is now converted to a classification task to determine if the statement converted from the original question is true, false, or unidentified. After computing the attention result $\boldsymbol{a}^k$ corresponding to every stacked bi-directional LSTM, we feed it into a classifier PRED^k to predict one from three candidate labels of each converted statement. Specifically, a fully connected layer with parameter $\boldsymbol{W}^k \in \mathbb{R}^{3 \times d}$ and a standard softmax function are used to output the multi-class label probabilities $\boldsymbol{P}^k(C) \in \mathbb{R}^3$:

$$\boldsymbol{P}^k(C) = \text{PRED}^k(\boldsymbol{a}^k) = \text{softmax}(\boldsymbol{W}^k \boldsymbol{a}^k). \tag{6}$$

Note that the multi-class label probabilities $\boldsymbol{P}^k(C)$ form a three-dimensional vector which contains three real values corresponding to the probability of each class. The objective function L^k corresponding to the k^{th} stacked bi-directional LSTM is to minimize the following multi-class cross-entropy loss:

$$L^k = -\frac{1}{N_{\text{b}}} \sum_{s=1}^{N_{\text{b}}} \sum_{c=1}^{3} y_{sc} \log_2 p_{sc}^k. \tag{7}$$

We use mini-batch gradient descent to train the model, where N_{b} denotes the number of samples in a batch, y_{sc} and p_{sc}^k denote the one-hot label corresponding to class c of the s^{th} sample in the batch and the probability predicted by the output layer, respectively. Specifically, as mentioned in Section 3, $c \in \{1, 2, 3\}$ represents the converted statement to be true, false, or unidentified.

4.5 Joint training

As encoders to a long text, stacked LSTMs often give better results than the single LSTM (Sutskever et al., 2014). However, stacked LSTMs are difficult to train because of exploding and vanishing gradient problems (Pascanu et al., 2012). To deal with this problem, more updates on the parameters of stacked LSTMs should be performed. Motivated by this insight, we jointly optimize all K multi-class cross-entropy losses corresponding to every stacked bi-directional LSTM in the passage encoding layer during the training process:

$$L = \sum_{k=1}^{K} \lambda_k L^k, \tag{8}$$

where λ_k denotes the weight of L^k and it is a hyper-parameter to be tuned. Note that only the output layer and attention layer following the K^{th} stacked bi-directional LSTM are used for model prediction, which means that L^K is the main loss and that all other $\{L^k\}_{k<K}$ are auxiliary losses. During joint training, all $\{\lambda_k\}_{k<K}$ linearly decrease after every epoch until they reach zero.

4.6 Data augmentation

During AIC2018, to better understand the OQMRC task, we carry out statistical analysis to obtain the distribution of three labels in the training set. Table 3 shows the results. Samples with label true account for the largest fraction in the training set (57.97%). Samples with label false are common and account for nearly one-third (32.04%) in total. In contrast, samples with label unidentified are relatively rare and account for only 9.99%. Such a distribution unbalances the training set and exerts a bad influence on model training.

To tackle this problem, we propose a data augmentation strategy to enrich the training set. Note that in the competition no external data are allowed, so our idea is to use the samples with label true to generate more samples with label unidentified. The strategy is applied to every sample with label true.

Fig. 3 shows an example of data augmentation. First, the top-10 term frequency-inverse document frequency (TF-IDF) (Wu HC et al., 2008) tokens of question and passage are selected, denoted as $\{\text{token}_\text{q}\}$ and $\{\text{token}_\text{p}\}$, respectively. After that, the intersection set $\{\text{token}_\text{qp}\}$ of $\{\text{token}_\text{q}\}$ and $\{\text{token}_\text{p}\}$ is computed. It is natural to consider that the sentences $\{s_\text{qp}\}$ containing words in $\{\text{token}_\text{qp}\}$ are more relevant to the question. After removing all such relevant sentences $\{s_\text{qp}\}$ from the original passage, the remaining part $\{s_\text{nqp}\}$ can be treated as irrelevant to the question. In this case, it cannot be known if the converted statement is true or false by reading $\{s_\text{nqp}\}$. Therefore, a new sample with label unidentified is generated, while its question is copied from the original sample with label true and its passage is $\{s_\text{nqp}\}$.

Table 3 Distribution of three labels in the training set for the AIC2018 OQMRC task

Label of converted statement	Number of samples	Ratio in total (%)
True	144 925	57.97
False	80 100	32.04
Unidentified	24 975	9.99

Question:
做礼拜能不能玩手机
Can I play on my cellphone at church?

Converted statement:
做礼拜能玩手机
I can play on my cellphone at church

Passage:
19%的美国人在去教堂做礼拜时玩手机。75%的美国人在任何时候手机都不会超出距离自己1.5米的范围。54%的美国人在床上玩手机，无论是睡觉前还是半夜醒来时。
19% of Americans play on cellphones while going to church. 75% of Americans keep their cellphones within 1.5 m from themselves. 54% of Americans play on cellphones in bed, no matter before sleeping or waking up in the midnight.

Label: true

$\{\text{token}_\text{q}\}$: {手机(cellphone)}

$\{\text{token}_\text{p}\}$: {手机(cellphone), 美国人(American)}

$\{\text{token}_\text{qp}\}$: {手机(cellphone)}

$\{s_\text{qp}\}$:
19%的美国人在去教堂做礼拜时玩手机。75%的美国人在任何时候手机都不会超出距离自己1.5米的范围。54%的美国人在床上玩手机
19% of Americans play on cellphones while going to church. 75% of Americans keep their cellphones within 1.5 m from themselves. 54% of Americans play on cellphones in bed

$\{s_\text{nqp}\}$:
无论是睡觉前还是半夜醒来时
no matter before sleeping or waking up in the midnight

Fig. 3 An example of data augmentation

5 Experiments

In this section, experiments are conducted to evaluate the performance of our approach. Experimental results show that the proposed model outperforms the AIC2018 baseline and other competing approaches. Ablation experiments are also conducted to analyze the contribution of each component.

5.1 Dataset and evaluation metrics

The AIC2018 OQMRC dataset consists of 300 000 samples in total, with 250 000 in the training set, 30 000 in the development set, and 20 000 in the hidden test set. To preserve the integrity of the competition, the organizers do not release the test set to the public. Everyone must submit the predicted file to obtain an official score on leaderboard.

Accuracy is used as the evaluation criterion in the AIC2018 OQMRC task. It is defined as the number of correctly answered questions divided by the total number of questions (in the dataset every sample contains only one question).

5.2 Baseline and competing systems

There is an official baseline system provided in the competition, which is implemented according to Tan et al. (2018). In addition, the proposed approach is compared with the following works introduced in Section 2: AR (Hermann et al., 2015), match-LSTM (Wang SH and Jiang, 2016), BIDAF (Seo et al., 2016), R-NET (Wang W et al., 2017), QANet (Yu et al., 2018), and BERT (Devlin et al., 2018). Note that some of these systems are designed for the span-selection MRC task. To make them suitable for this OQMRC competition, fully connected layers are used to replace the original pointer networks to deal with this task as described in Section 3.

5.3 Implementation details

In the AIC2018 competition, external data are forbidden. Thus, the word and character embeddings used by the competing systems (including BERT pre-training) can be trained only with the textual context in the AIC2018 OQMRC dataset. However, the official baseline system provided the basic Chinese NLP tool Jieba (https://pypi.org/project/jieba/) to obtain the POS tags for pre-training the hybrid embedding and TF-IDF values for data augmentation.

The dimensions of the hybrid embedding and the hidden vector for all layers are set to 256. Because of the GPU memory limitation of the virtual machine provided in AIC2018, at most four stacked LSTMs can be used in the passage encoding layers. Otherwise, there would be an out of memory (OOM) issue in the training process. Dropout (Srivastava et al., 2014) is applied between every two connected layers, with a rate of 0.2. To keep most passages (99.9%) complete, the maximum length of passage input is set to 300 words. Longer ones will be cut and zero-vectors will be appended to the tail if the passage is shorter. Specifically, only the generated samples whose passage is longer than five words are kept in data augmentation. We add 53 210 new samples with label unidentified in total to the training set. During the training process, the model is optimized via Adam (Kingma and Ba, 2014) with a fixed learning rate of 1.0×10^{-3} and a batch size of 256. All $\{\lambda_k\}$ are initialized as one, and all $\{\lambda_k\}_{k<K}$ linearly decrease by 0.05 after every epoch until zero.

5.4 Results

Table 4 shows the performances of the proposed approach and other competing systems on the AIC2018 OQMRC dataset. The aim is to compare the performances of different approaches; therefore, we report the single system but not multi-system ensemble results. The development (dev) set accuracy is computed offline and the test set accuracy is obtained by submitting the predicted file to the official website of AIC2018. As Table 4 shows, the proposed approach achieves the state-of-the-art results with 76.35% dev accuracy and 77.52% test accuracy, outperforming all other methods.

Table 4 Accuracies of the proposed approach and competing systems in the dataset for the AIC2018 OQMRC task

Method	Accuracy (%)	
	Dev	Test
Official baseline (Tan et al., 2018)	69.52	69.90
AR (Hermann et al., 2015)	65.32	66.04
Match-LSTM (Wang SH and Jiang, 2016)	70.25	70.99
BIDAF (Seo et al., 2016)	72.30	72.56
R-NET (Wang W et al., 2017)	73.66	74.14
QANet (Yu et al., 2018)	61.37	62.11
BERT (Devlin et al., 2018)	70.65	70.99
Our approach	**76.35**	**77.52**

Dev: development. The best results are in bold

5.5 Ablation studies and discussion

We conduct ablation experiments on each component of the proposed approach, and investigate the effect of data augmentation. Note that the test set is not published, so we must spend one submit chance to obtain an official test accuracy every time. To save submit chances during the competition, some of the ablation experiments are not conducted on the test set. However, the ablation studies are still convincing since all the results on the development (dev) set are reported.

For ablating hybrid embedding (HE) introduced in Section 4.1, we do not use the POS tagging tool and pre-train the embedding with a simple word-segmented corpus. Note that character-level embedding (CE) is also widely used in the MRC task (Seo et al., 2016; Wang W et al., 2017), and that there are other ways to incorporate POS information into the representation, like concatenating POS tag embedding (PE) with word embedding (Liu et al., 2018). We also test these methods in the ablation experiments.

For ablating joint training (JT) introduced in Section 4.5, all the extra layers in dashed parts of Fig. 2 are removed, and the networks are trained with a single main loss. We also test residual connections (RC) in stacked LSTMs for comparison. This is famous for training deeper and stacked layers (He KM et al., 2016; Wu YH et al., 2016). With residual connections between stacked LSTMs, Eq. (3) becomes

$$\boldsymbol{p}_j^k = \mathrm{BiLSTM}_P^k(\boldsymbol{p}_{j-1}^k, \boldsymbol{p}_j^{k-1}) + \boldsymbol{p}_j^{k-1}. \tag{9}$$

Data augmentation (DA) is ablating by excluding the new samples generated from the training set.

As illustrated in Table 5, all components contribute towards the performance of the proposed approach. Removing any of the components, or replacing it with a comparative one, leads to lower accuracy. The use of JT allows the network parameters to obtain more and better updates, and thus it is crucial (both dev and test accuracies drop drastically by more than 3% if it is removed). Hybrid embedding is also a necessary component that contributes 1.7%/1.49% gain of dev/test accuracy, since it combines the POS information of context around rare words and can better handle the out-of-vocabulary (OOV) problem, especially in the situation where external data are forbidden. The strategy of data augmentation also makes a prominent contribution to performance. It causes slightly lower dev accuracy (almost 1%) if ablated. This demonstrates that the generated samples can relieve the problem of data imbalance.

As mentioned in Section 4.5, deeper stacked LSTMs are more difficult to train because of exploding and vanishing gradient problems. The JT method is proposed to solve this problem. Table 6 shows the additional ablation experiments to investigate the impact of depth in the model with the JT method. The number of layers in the stacked LSTMs is changed from two to four to evaluate the development set.

We have three major observations from Table 6. First, the three-layer stacked LSTMs have higher accuracy than the two-layer stacked LSTMs, whether there is JT or not. However, four-layer stacked LSTMs have lower accuracy without JT due to the difficulty in training deeper stacked LSTMs. Second, with JT the situation is reversed—the four-layer stacked LSTMs achieve higher accuracy than the three-layer ones. Moreover, all the two to four layers achieve higher accuracies. Last, deeper nets achieve more improvements in accuracy than shallower nets. With JT the four-layer stacked LSTMs achieve 3.23% improvement, while the three-layer ones achieve 1.16% and two-layer ones only 0.74%. The residual connections bring similar but smaller improvements. This demonstrates the effectiveness of the proposed JT method.

Table 5 Ablation performance of the proposed approach

Method	Accuracy (%)	
	Dev	Test
Our approach	76.35	77.52
-JT	73.12 (−3.23)	73.96 (−3.56)
-JT+RC	75.28 (−1.07)	N/A
+RC	76.26 (−0.09)	77.33 (−0.19)
-HE	74.65 (−1.70)	76.03 (−1.49)
-HE+CE	74.77 (−1.58)	N/A
-HE+PE	74.55 (−1.80)	N/A
-HE+CE+PE	74.71 (−1.64)	N/A
+CE+PE	75.84 (−0.51)	76.97 (−0.55)
-DA	75.44 (−0.91)	N/A

-: exclude; +: include. JT: joint training; HE: hybrid embedding; DA: data augmentation; RC: residual connections; CE: character-level embedding; PE: POS tag embedding; Dev: development. N/A: not applicable

Table 6 Ablation performance of joint training in the development set

Method	Accuracy (%)		
	Two-layer	Three-layer	Four-layer
Our approach	73.94	75.11	76.35
-JT	73.20 (−0.74)	73.95 (−1.16)	73.12 (−3.23)
-JT+RC	73.18 (−0.76)	74.27 (−0.84)	75.28 (−1.07)
+RC	73.77 (−0.17)	75.02 (−0.09)	76.26 (−0.09)

-: exclude; +: include. JT: joint training; RC: residual connections

6 Conclusions

In this paper, we have focused on real-world opinion question machine reading comprehension. A novel approach based on neural networks has been proposed to tackle the problem. POS tags have been combined into embedding pre-training to enrich the semantic representation of question and passage. Extra attention and output layers have been introduced in the training process, and multiple losses have been jointly optimized to better update the parameters of networks. To relieve the problem of data imbalance in the competition, a data augmentation strategy has been implemented to generate new samples. Experimental results indicated that the proposed approach achieved state-of-the-art performance in the challenging AIC2018 OQMRC dataset. The ablation analyses also demonstrated the importance of each component of the proposed approach. In the future, we plan to use hybrid embedding in other neural network models and NLP tasks. Furthermore, we will extend the JT framework to handle deeper stacked LSTMs.

Contributors

Xiang-zhou HUANG, Si-liang TANG, Yin ZHANG, and Bao-gang WEI designed the research. Xiang-zhou HUANG processed the data and drafted the manuscript. Si-liang TANG, Yin ZHANG, and Bao-gang WEI helped organize the manuscript. Xiang-zhou HUANG revised and finalized the paper.

Compliance with ethics guidelines

Xiang-zhou HUANG, Si-liang TANG, Yin ZHANG, and Bao-gang WEI declare that they have no conflict of interest.

References

Bajaj P, Campos D, Craswell N, et al., 2016. MS MARCO: a human generated MAchine Reading COmprehension dataset.
https://arxiv.org/abs/1611.09268

Devlin J, Chang MW, Lee K, et al., 2018. BERT: pre-training of deep bidirectional transformers for language understanding.
https://arxiv.org/abs/1810.04805

He KM, Zhang XY, Ren SQ, et al., 2016. Deep residual learning for image recognition. IEEE Conf on Computer Vision and Pattern Recognition, p.770-778.
https://doi.org/10.1109/CVPR.2016.90

He W, Liu K, Liu J, et al., 2018. DuReader: a Chinese machine reading comprehension dataset from real-world applications. Proc Workshop on Machine Reading for Question Answering, p.37-46.
https://doi.org/10.18653/v1/W18-2605

Hermann KM, Kočiský T, Grefenstette E, et al., 2015. Teaching machines to read and comprehend. Proc 28th Int Conf on Neural Information Processing Systems, p.1693-1701.

Hochreiter S, Schmidhuber J, 1997. Long short-term memory. *Neur Comput*, 9(8):1735-1780.
https://doi.org/10.1162/neco.1997.9.8.1735

Joshi M, Choi E, Weld DS, et al., 2017. TriviaQA: a large scale distantly supervised challenge dataset for reading comprehension.
https://arxiv.org/abs/1705.03551

Kingma DP, Ba J, 2014. Adam: a method for stochastic optimization.
https://arxiv.org/abs/1412.6980

Liu JH, Wei W, Sun MS, et al., 2018. A multi-answer multi-task framework for real-world machine reading comprehension. Proc Conf on Empirical Methods in Natural Language Processing, p.2109-2118.
https://doi.org/10.18653/v1/D18-1235

Mikolov T, Sutskever I, Chen K, et al., 2013a. Distributed representations of words and phrases and their compositionality. Proc 26th Int Conf on Neural Information Processing Systems, p.3111-3119.

Mikolov T, Chen K, Corrado G, et al., 2013b. Efficient estimation of word representations in vector space.
https://arxiv.org/abs/1301.3781

Pan YH, 2016. Heading toward artificial intelligence 2.0. *Engineering*, 2(4):409-413.
https://doi.org/10.1016/J.ENG.2016.04.018

Pascanu R, Mikolov T, Bengio Y, 2012. Understanding the exploding gradient problem.
https://arxiv.org/abs/1211.5063v1

Rajpurkar P, Zhang J, Lopyrev K, et al., 2016. SQuAD: 100 000+ questions for machine comprehension of text. Proc Conf on Empirical Methods in Natural Language Processing, p.2383-2392.
https://doi.org/10.18653/v1/D16-1264

Richardson M, Burges CJC, Renshaw E, 2013. MCTest: a challenge dataset for the open-domain machine comprehension of text. Proc Conf on Empirical Methods in Natural Language Processing, p.193-203.

Seo M, Kembhavi A, Farhadi A, et al., 2016. Bidirectional attention flow for machine comprehension.
https://arxiv.org/abs/1611.01603

Srivastava N, Hinton G, Krizhevsky A, et al., 2014. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res*, 15(1):1929-1958.

Sutskever I, Vinyals O, Le QV, 2014. Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems, p.3104-3112.

Tan CQ, Wei FR, Wang WH, et al., 2018. Multiway attention networks for modeling sentence pairs. Proc 27th Int Joint Conf on Artificial Intelligence, p.4411-4417.
https://doi.org/10.24963/ijcai.2018/613

Vinyals O, Fortunato M, Jaitly N, 2015. Pointer networks. Advances in Neural Information Processing Systems, p.2692-2700.

Wang SH, Jiang J, 2016. Learning natural language inference with LSTM. Proc Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.1442-1451.
https://doi.org/10.18653/v1/N16-1170

Wang W, Yang N, Wei F, et al., 2017. R-NET: Machine Reading Comprehension with Self-matching Networks. Technical Report, Natural Language Computing Group, Microsoft Research Asia, Beijing, China. https://www.microsoft.com/en-us/research/wp-content/uploads/2017/05/r-net.pdf

Wu HC, Luk RWP, Wong KF, et al., 2008. Interpreting TF-IDF term weights as making relevance decisions. *ACM Trans Inform Syst*, 26(3):13. https://doi.org/10.1145/1361684.1361686

Wu YH, Schuster M, Chen ZF, et al., 2016. Google's neural machine translation system: bridging the gap between human and machine translation. https://arxiv.org/abs/1609.08144

Yang Y, Yih WT, Meek C, 2015. WikiQA: a challenge dataset for open-domain question answering. Proc Conf on Empirical Methods in Natural Language Processing, p.2013-2018. https://doi.org/10.18653/v1/D15-1237

Yu AW, Dohan D, Luong MT, et al., 2018. QANet: combining local convolution with global self-attention for reading comprehension. https://arxiv.org/abs/1804.09541

Zhuang YT, Wu F, Chen C, et al., 2017. Challenges and opportunities: from big data to knowledge in AI 2.0. *Front Inform Technol Electron Eng*, 18(1):3-14. https://doi.org/10.1631/FITEE.1601883