Interactive Mongolian Question Answer Matching Model Based on Attention Mechanism in the Law Domain

Yutao Peng, Weihua Wang✉, Feilong Bao
College of Computer Science, Inner Mongolia University, China
National & Local Joint Engineering Research Center of Intelligent Information Processing Technology for Mongolian, China
Inner Mongolia Key Laboratory of Mongolian Information Processing Technology, China
yutao.peng@mail.imu.edu.cn
{wangwh,csfeilong}@imu.edu.cn

Abstract

Mongolian question answer matching task is challenging, since Mongolian is a kind of low-resource language and its complex morphological structures lead to data sparsity. In this work, we propose an Interactive Mongolian Question Answer Matching Model (IMQAMM) based on attention mechanism for Mongolian question answering system. The key parts of the model are interactive information enhancement and max-mean pooling matching. Interactive information enhancement contains sequence enhancement and multi-cast attention. Sequence enhancement aims to provide a subsequent encoder with an enhanced sequence representation, and multi-cast attention is designed to generate scalar features through multiple attention mechanisms. Max-Mean pooling matching is to obtain the matching vectors for aggregation. Moreover, we introduce Mongolian morpheme representation to better learn the semantic feature. The model experimented on the Mongolian corpus, which contains question-answer pairs of various categories in the law domain. Experimental results demonstrate that our proposed Mongolian question answer matching model significantly outperforms baseline models.

1 Introduction

Question answer matching is used to identify the relationship between the question-answer pairs, and it is one of the application scenarios of text matching. Text matching is an important fundamental technology in Natural Language Processing (NLP) and can be applied to a large number of NLP tasks, such as Information Retrieval (IR), Natural Language Inference (NLI), question answering (QA) system, dialogue system, etc. For the tasks of Information Retrieval, text matching is utilized to compute the relevance between queries and documents to select the relevant documents (Huang et al., 2013). For the tasks of Natural Language Inference, text matching is employed to judge whether the premise can infer the hypothesis (Bowman et al., 2015). And for the question answering tasks, text matching is applied to pick the answers that are most relevant to a given question (Tan et al., 2016).

With the development of deep learning, text matching methods with neural network are increasingly emerging. These methods can be divided into two types—representation-based match and interaction-based match. The first type is representation-based match (Huang et al., 2013; Shen et al., 2014; Palangi et al., 2014), which is focused on modeling the representations of the two sentences, so that they are encoded into semantic vectors in the same embedding space. The second type is interaction-based match (Chen et al., 2017; Tay et al., 2017; Wang et al., 2017), which is targeted at interacting with each information between sentence pairs to improve the process of representation learning. Interaction-based match performs better than representation-based match, because representation-based match lacks a comparison of lexical and syntactic information between sentence pairs, while interaction-based match can take advantage of the interactive information across sentence pairs to enhance their own representations. Therefore, interactive matching methods are currently the mainstreaming methods of text matching.

However, the development of Mongolian question answering system is relatively slow, and there are few studies about it. The first reason for the slow development is that Mongolian is a kind of low-resource language. It lacks public labeled corpus. The second reason is the data-sparse problem caused

Proceedings of the 21st China National Conference on Computational Linguistics, pages 896-907, Nanchang, China, October 14 - 16, 2022.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

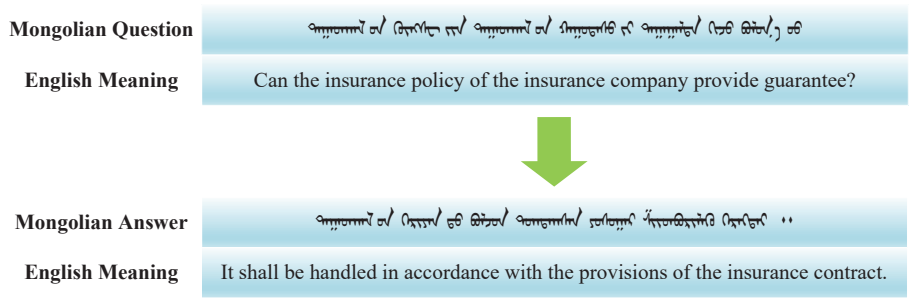

Figure 1: A Mongolian question-answer pair example.

by complex Mongolian morphological structures. Mongolian is an agglutinative language and its root can be followed by different suffixes, which is different from Chinese and English.

In this paper, we construct a Mongolian question answering data set in the law domain. An example is shown in Figure 1, there are two Mongolian sentences, including the question and its corresponding answer. Our task is to judge whether the answer corresponds to the question. In order to solve the problem of insufficient information enhancement of previous interaction-based matching methods, we propose an Interactive Mongolian Question Answer Matching Model (IMQAMM), which combines interactive information enhancement and max-mean pooling matching. Interactive information enhancement concatenates a series of feature vectors to get the enhanced sequence representation and adopts a compression function to reduce feature vectors to scalars based on multiple attention mechanisms for the issue of data-sparse. Max-Mean pooling matching is to compute the maximum and average cosine similarities corresponding to each morpheme representation.

The paper is organized as follows: Section 2 introduces the related work. Section 3 presents the details of our Mongolian question answer matching model. Section 4 shows our experimental setup and results. Section 5 gives the conclusion and the future work.

2 Related Work

Text semantic matching is the core of retrieval question answering system. In recent years, there has been a lot of research on text semantic matching, which has driven the development of question answering systems. Early work on text semantic matching was based on relatively simple models to compare the sentence pairs. Most of these work was based on sentence encoding methods. Huang et al. (2013) proposed Deep Structured Semantic Models (DSSM), which projected queries and documents into a common low-dimensional space to get similarities between them. Since the CNN has the characteristics of local perception and parameter sharing, Shen et al. (2014) improved the DSSM model and proposed the Convolutional Latent Semantic Model (CLSM). To tackle the limitations of RNN in capturing contextual dependencies, Palangi et al. (2014) presented the Long Short Term Memory DSSM (LSTM-DSSM). Mueller and Thyagarajan (2016) proposed the Manhattan LSTM (MaLSTM) model, which utilized the siamese LSTM to compute the Manhattan distance between sentence pairs. Shen et al. (2018) proposed Directional Self-Attention Network (DiSAN) without any RNN or CNN for sentence encoding. Yang and Kao (2020) proposed a Siamese sentence encoder that would not propagate any interactive information between sentence pairs.The matching methods based on attention networks have gradually developed (Tan et al., 2016; Yin et al., 2016; Kim et al., 2019). Tan et al. (2016) used the attentive LSTM to match questions and passage answers with question information. Yin et al. (2016) explored computing the attention matrix before and after convolution to select correct answers for a question. Kim et al. (2019) used densely-connected recurrent neural network and concatenated the co-attentive features in multiple layers for semantic sentence matching.

Compare-Aggregate networks are very popular in different tasks (Wang and Jiang, 2017; Wang et al., 2017; Chen et al., 2017; Tay et al., 2017; Tay et al., 2018). Wang and Jiang (2017) presented a compare-aggregate model that performed word-level matching by element-wise multiplication or subtraction and

Proceedings of the 21st China National Conference on Computational Linguistics, pages 896-907, Nanchang, China, October 14 - 16, 2022.

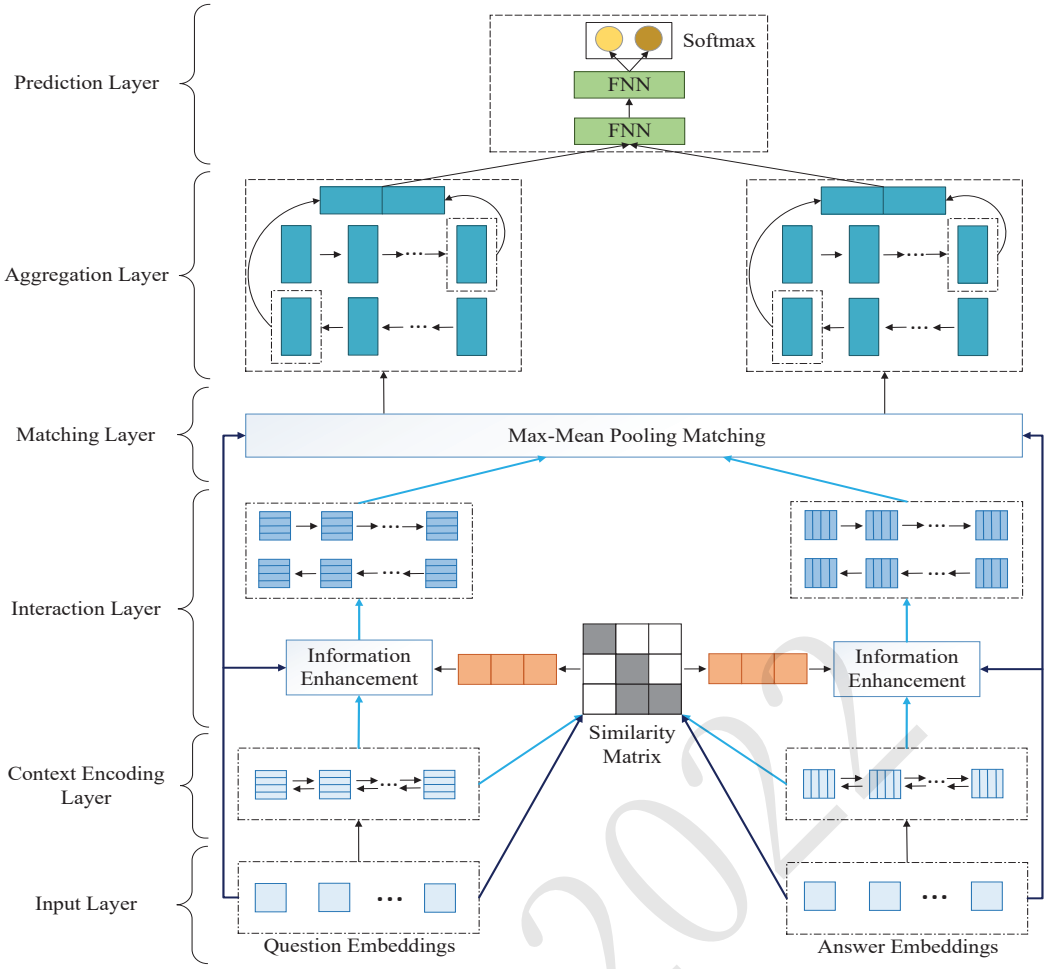

Figure 2: Architecture for Interactive Mongolian Question Answer Matching Model (IMQAMM), where the initial morpheme representations and the contextual representations are respectively applied to compute the similarity matrix for interactive information enhancement.

aggregated by CNN. Wang et al. (2017) proposed a bilateral multi-perspective matching (BiMPM) model, which used multi-perspective cosine matching strategy between encoded sentence pairs. Chen et al. (2017) improved the approach proposed by Parikh et al. (2016) and achieved sequential inference model using chain LSTMs. Tay et al. (2017) presented ComProp Alignment-Factorized Encoders (CAFE) that used factorization machines to compress the alignment vectors into scalar features, which can effectively augment the word representations. Tay et al. (2018) explored using Multi-Cast Attention Networks (MCAN) to improve learning process by adopting several attention variants and performing multiple comparison operators.

These text semantic matching models laid the foundation for later IR models and QA systems. Although these models have achieved state-of-the-art performance on various datasets, they may not be suitable for low-resource agglutinative languages. In this paper, we introduce Mongolian morpheme representation, then use interactive information enhancement to take full advantage of the information across Mongolian question-answer pairs and apply max-mean pooling matching to capture the maximum influence and the overall influence between Mongolian question-answer pairs.

3 Model Architecture

In this section, we will describe our model architecture layer by layer. Figure 2 shows a high-level view of the architecture, and then the details of our model are given as follows.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 896–907, Nanchang, China, October 14 – 16, 2022.

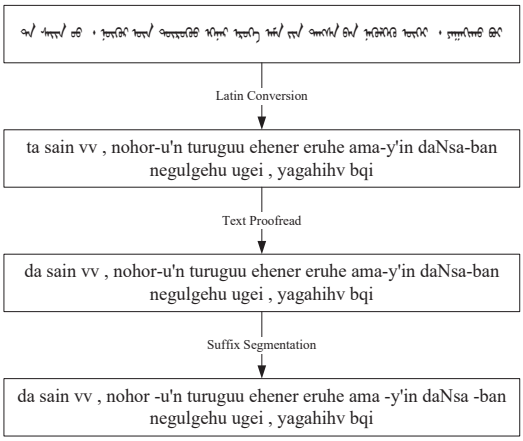

Figure 3: An example of traditional Mongolian transformation steps.

3.1 Input Layer

Mongolian is a kind of agglutinative language with complex morphological structures (Wang et al., 2015). Although there are natural spaces between Mongolian words, morphological segmentation is still needed for us. Mongolian word-formation is achieved by appending different suffixes to the stem, and they can also be concatenated layer by layer, which can lead to data sparsity. In this paper, we use Latin to deal with Mongolian and segment the suffixes to get the morpheme representations (Wang et al., 2019).

Before getting the morpheme representations of Mongolian question-answer pairs, we need to make some transformations to the traditional Mongolian language. As shown in Figure 3, the steps of transformation are divided into three steps. First of all, we convert the traditional Mongolian alphabet to the corresponding Latin alphabet. Next, because a Mongolian glyph can map to different letters, it is necessary to proofread the text (Lu et al., 2019). Finally, the suffixes connect to the stem through a Narrow No-Break Space (NNBS) (U+202F, Latin:"-"), so we can segment the suffixes to get the independent training units.

To obtain the morpheme embeddings of Mongolian question-answer pairs, we adopt Word2Vec (Mikolov et al., 2013), which contains CBOW (Continuous Bag of Word) and Skip-gram. And we choose the Skip-gram model to train the morpheme vectors.

3.2 Context Encoding Layer

LSTM is a variant of RNN, which can capture contextual dependencies effectively. In order to better represent the semantic information, we utilize the bi-directional LSTM (BiLSTM) to extract contextual features from question embeddings q and answer embeddings a.

$$\bar{q_i} = BiLSTM(q,\ i)\ ,\ \forall i\ \in\ [1,\ \ldots,\ m] \tag{1}$$

$$\bar{a_j} = BiLSTM(a,\ j)\ ,\ \forall j\ \in\ [1,\ \ldots,\ n] \tag{2}$$

where m is the length of question sentence, and n is the length of answer sentence.

3.3 Interaction Layer

In this layer, we introduce the interactive information enhancement, which contains sequence enhancement based on LSTMs and multi-cast attention using four variants of attention mechanism.

3.3.1 Sequence Enhancement

Inspired by the ESIM proposed by Chen et al. (2017), we also adopt the non-parameterized comparison strategy for sequence enhancement. Firstly, we calculate the similarity matrix between a question-answer pair encoded by BiLSTM.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 896-907, Nanchang, China, October 14 - 16, 2022.

$$e_{ij} = \bar{q_i}^T \bar{a_j} \tag{3}$$

Then the key of the strategy is soft alignment attention, which can get an attentive vector of a weighted summation of the other hidden states ($\bar{a_j}$ or $\bar{q_i}$). This process is shown in the following formulas:

$$\tilde{q_i} = \sum_{j=1}^{n} \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \bar{a_j} \ , \ \forall i \in [1, \ldots, m] \tag{4}$$

$$\tilde{a_j} = \sum_{i=1}^{m} \frac{\exp(e_{ij})}{\sum_{k=1}^{m} \exp(e_{kj})} \bar{q_i} \ , \ \forall j \in [1, \ldots, n] \tag{5}$$

where $\tilde{q_i}$ is a weighted summation of $\{\bar{a_j}\}_{j=1}^{n}$, $\tilde{a_j}$ is a weighted summation of $\{\bar{q_i}\}_{i=1}^{m}$.

Finally, we use the original hidden states and the attentive vectors to compute the difference and the element-wise product, which are then concatenated with the original hidden states and the attentive vectors.

$$T_i^q = [\bar{q_i}; \tilde{q_i}; \bar{q_i} - \tilde{q_i}; \bar{q_i} \odot \tilde{q_i}] \ , \ \forall i \in [1, \ldots, m] \tag{6}$$

$$T_j^a = [\bar{a_j}; \tilde{a_j}; \bar{a_j} - \tilde{a_j}; \bar{a_j} \odot \tilde{a_j}] \ , \ \forall j \in [1, \ldots, n] \tag{7}$$

3.3.2 Co-Attention

Co-attention is a pair-wise attention mechanism, which has a natural symmetry between sentence pairs or other pairs (Lu et al., 2017). Co-attention is a kind of variant of attention mechanism, and in this work, we decide to adopt four variants of attention mechanism: (1) **max-pooling co-attention**, (2) **mean-pooling co-attention**, (3) **alignment-pooling co-attention**, and (4) **self attention**.

The first step is to connect question and answer by calculating the similarity matrix between the initial morpheme embeddings of question-answer pairs.

$$s_{ij} = q_i^T M a_j \tag{8}$$

where M is a trainable parameter matrix.

Extractive pooling includes max-pooling and mean-pooling. **Max-pooling co-attention** aims to attend each morpheme of the sequence based on the maximum effect on each morpheme of the other sequence, while **mean-pooling co-attention** is focused on the average effect. The formulas are as following:

$$q_1' = Softmax(\max_{col}(s))^\top q \qquad a_1' = Softmax(\max_{row}(s))^\top a \tag{9}$$

$$q_2' = Softmax(\underset{col}{mean}(s))^\top q \qquad a_2' = Softmax(\underset{col}{mean}(s))^\top a \tag{10}$$

where q_1', q_2', a_1' and a_2' are the co-attentive representations of q or a.

Similar to the sequence enhancement mentioned above, **alignment-pooling co-attention** is computed individually to softly align each morpheme to the other sequence. The process is shown in the following formulas:

$$\tilde{q_i}' = \sum_{j=1}^{n} \frac{\exp(s_{ij})}{\sum_{k=1}^{n} \exp(s_{ik})} a_j \ , \ \forall i \in [1, \ldots, m] \tag{11}$$

$$\tilde{a_j}' = \sum_{i=1}^{m} \frac{\exp(s_{ij})}{\sum_{k=1}^{m} \exp(s_{kj})} q_i \ , \ \forall j \in [1, \ldots, n] \tag{12}$$

where $\widetilde{q_i}'$ is a weighted summation of $\{a_j\}_{j=1}^{n}$, $\widetilde{a_j}'$ is a weighted summation of $\{q_i\}_{i=1}^{m}$.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 896-907, Nanchang, China, October 14 - 16, 2022.

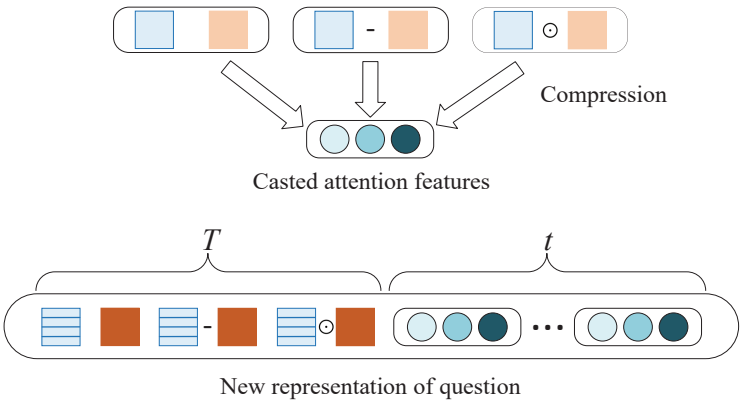

Figure 4: Information enhancement of question.

Self attention is applied to both question and answer independently. The sentence representation is denoted by x instead of q or a. The self attention function is computed as:

$$x_i' = \sum_{j=1}^{l} \frac{\exp(s_{ij})}{\sum_{k=1}^{l} \exp(s_{ik})} x_j \tag{13}$$

where $x_i{}'$ is the self-attentional representation of x_j, l is the length of the sentence.

3.3.3 Multi-Cast Attention

Multi-cast attention can get a multi-casted feature vector from multiple attention mechanisms. Each attention mechanism performs concatenation, subtractive and multiplicative operations respectively, and uses a compression function to get three scalars. The initial morpheme embeddings of a question-answer pair q and a are replaced by x, and $\tilde{x}$ is the attentive vector. The casted attention features for each attention mechanism are shown in the following formulas:

$$f_{con} = F_c([\tilde{x}; x]) \tag{14}$$

$$f_{sub} = F_c(\tilde{x} - x) \tag{15}$$

$$f_{mul} = F_c(\tilde{x} \odot x) \tag{16}$$

where F_c is a compression function, $[.;.]$ is the concatenation operator and $\odot$ is the element-wise product.

Factorization Machines (FM) can make predictions on any real-valued feature vector (Rendle, 2010). Therefore, we adopt FM as a compression function to get casted scalars. The function is as follows:

$$F(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < v_i, v_j > x_i x_j \tag{17}$$

where $w_0 \in \mathbb{R}$, $w_i \in \mathbb{R}^n$, $v_1, \ldots, v_n \in \mathbb{R}^{n \times k}$, and k is the number of latent factors of the FM model.

For each Mongolian question-answer pair, we apply four variants of attention mechanism mentioned above: (1) Max-pooling co-attention (2) Mean-pooling co-attention (3) Alignment-pooling co-attention and (4) Self-attention. Take the question sentence as an example, as shown in the Figure 4, three scalars are generated from each attention mechanism, so the final multi-casted feature vector is $t \in \mathbb{R}^{12}$. As such, for each morpheme, we concatenate the enhanced sequence representation T and the multi-casted feature vector t to get the new representation O^q. And O^a can be obtained in the same way.

$$O_i^q = [T_i^q; t_i^q] \ , \ \forall i \in [1, \ldots, m] \tag{18}$$

$$O_j^a = [T_j^a; t_j^a] \ , \ \forall j \in [1, \ldots, n] \tag{19}$$

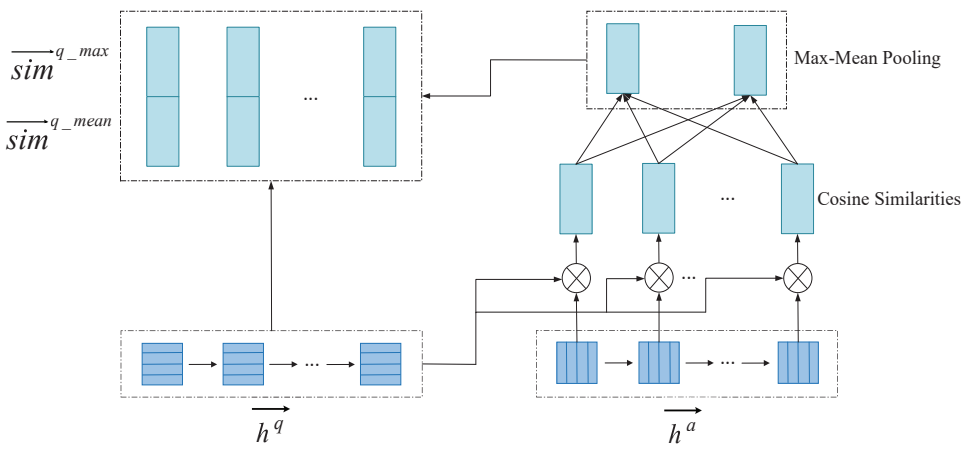

Figure 5: The max-mean pooling matching in forward direction of matching direction $q \rightarrow a$.

We use BiLSTM to encode interaction information at each time-step of O^q and O^a.

$$\overrightarrow{h_i^q} = \overrightarrow{LSTM}(h_{i-1}^q, O_i^q) \quad i = 1, \ldots, m \qquad \overleftarrow{h_i^q} = \overleftarrow{LSTM}(h_{i+1}^q, O_i^q) \quad i = m, \ldots, 1 \tag{20}$$

$$\overrightarrow{h_j^a} = \overrightarrow{LSTM}(h_{j-1}^a, O_j^a) \quad j = 1, \ldots, n \qquad \overleftarrow{h_j^a} = \overleftarrow{LSTM}(h_{j+1}^a, O_j^a) \quad j = n, \ldots, 1 \tag{21}$$

3.4 Matching Layer

To match question-answer pairs, we adopt the max-mean pooling matching strategy. Firstly, the cosine function is defined as follows:

$$sim = f_s(v_1, v_2; W) \tag{22}$$

where v_1 and v_2 are the d-dimensional vectors to be matched, $W \in \mathbb{R}^{l \times d}$ is the trainable parameter matrix, and l is the number of perspectives. For each dimension of the dimension space, it can be assigned different weights. Thus, the matching value from the k-th perspective is calculated by the formula as follows:

$$sim_k = cosine(W_k \circ v_1, W_k \circ v_2) \tag{23}$$

where $\circ$ represents the element-wise product, W_k is the k-th low of W.

Then we compare each time-step of question (or answer) representation against all time-steps of answer (or question) representation. For convenience, we only define the matching direction $q \rightarrow a$.

Morpheme Matching For the initial morpheme embeddings of question-answer pairs, we define the max-mean pooling matching strategy. The formulas are as following:

$$\overrightarrow{sim}_i^{q\_max} = \max_{j \in (1 \ldots n)} f_s(q, a; W^1) \tag{24}$$

$$\overrightarrow{sim}_i^{q\_mean} = \operatorname*{mean}_{j \in (1 \ldots n)} f_s(q, a; W^1) \tag{25}$$

Interaction Matching And for the representations of question-answer pairs after interaction, we also define the max-mean pooling matching strategy in forward direction and backward direction. Figure 5 shows the max-mean pooling matching in forward direction. The formulas are as following:

$$\overrightarrow{sim}_i^{q\_max} = \max_{j \in (1 \ldots n)} f_s\left(\overrightarrow{h_i^q}, \overrightarrow{h_j^a}; W^2\right) \qquad \overleftarrow{sim}_i^{q\_max} = \max_{j \in (1 \ldots n)} f_s\left(\overleftarrow{h_i^q}, \overleftarrow{h_j^a}; W^3\right) \tag{26}$$

$$\overrightarrow{sim}_i^{q\_mean} = \operatorname*{mean}_{j \in (1 \ldots n)} f_s\left(\overrightarrow{h_i^q}, \overrightarrow{h_j^a}; W^2\right) \qquad \overleftarrow{sim}_i^{q\_mean} = \operatorname*{mean}_{j \in (1 \ldots n)} f_s\left(\overleftarrow{h_i^q}, \overleftarrow{h_j^a}; W^3\right) \tag{27}$$

At last, we concatenate all the results of the max-mean pooling matching.

$$sim_i^q = [\overrightarrow{sim}_i^{q\_max}; \overrightarrow{sim}_i^{q\_mean}; \overrightarrow{sim}_i^{q\_max}; \overrightarrow{sim}_i^{q\_mean}; \overleftarrow{sim}_i^{q\_max}; \overleftarrow{sim}_i^{q\_mean}] \tag{28}$$

where $i \in [1, \ldots, m]$, max is element-wise maximum and $mean$ is element-wise mean. The calculation process of sim_j^a is similar to that of sim_i^q.

3.5 Aggregation Layer

We utilize BiLSTM to aggregate the matching vectors sim_i^q and sim_j^a, which are calculated from two matching directions $q \to a$ and $a \to q$.

$$\overrightarrow{v_i^q} = \overrightarrow{LSTM}(v_{i-1}^q, sim_i^q) \quad i = 1, \ldots, m \qquad \overleftarrow{v_i^q} = \overleftarrow{LSTM}(v_{i+1}^q, sim_i^q) \quad i = m, \ldots, 1 \tag{29}$$

$$\overrightarrow{v_j^a} = \overrightarrow{LSTM}(v_{j-1}^a, sim_j^a) \quad j = 1, \ldots, n \qquad \overleftarrow{v_j^a} = \overleftarrow{LSTM}(v_{j+1}^a, sim_j^a) \quad j = n, \ldots, 1 \tag{30}$$

Then we concatenate the last hidden states of BiLSTM models used in two matching directions.

$$y_{out} = [\overrightarrow{v_m^q}; \overleftarrow{v_1^q}; \overrightarrow{v_n^a}; \overleftarrow{v_1^a}] \tag{31}$$

3.6 Prediction Layer

Mongolian question answer matching in this paper is a binary classification problem. We then pass the output of aggregation y_{out} into a two-layer feed-forward neural network and a softmax layer.

$$y_{pred} = softmax(W_2^F \cdot tanh(W_1^F \cdot y_{out} + b_1^F) + b_2^F) \tag{32}$$

where $W_1^F \in \mathbb{R}^{h_1 \times h_2}$, $b_1^F \in \mathbb{R}^{h_2}$, $W_2^F \in \mathbb{R}^{h_2 \times 2}$, $b_2^F \in \mathbb{R}^2$.

3.7 Model training

To train our model, we minimize the binary cross-entropy loss.

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log P_i + (1 - y_i)\log(1 - P_i)] \tag{33}$$

where N is the number of labels, $y_i \in \{0, 1\}$ and P_i is the predicted probability.

4 Experiments

In this section, we describe our experimental setup and give our experimental results.

4.1 Data set and Evaluation Metrics

Our Mongolian question answering data set is translated from the Chinese question answering corpus and crawled from the Mongolian web sites. In order to improve the generalization ability of the model, we extend the original data set and construct negative samples. The ratio of positive and negative samples is 1 : 1. The data set contains 265194 question-answer pairs and each category is randomly divided into train, dev and test with the percent 80%, 10% and 10%, respectively.

We adopt Precision (P), Recall (R), F1-score (F1) and Accuracy (Acc) as the evaluation metrics of our experiments.

4.2 Model Configuration

We implement our model in TensorFlow. The batch size is set to 128, the epoch is set to 20, the max sentence length is set to 50 and the number of perspectives is set to 5. We use pre-trained 300-dimensional Mongolian Word2Vec embeddings. The size of hidden layers of all BiLSTM layers is set to 100. We use dropout with a rate of 0.1, which is applied to every layer. For training, we use the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.0005 to update parameters.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 896-907, Nanchang, China, October 14 - 16, 2022.

4.3 Baselines

In this subsection, we compare our model with several matching models on the Mongolian question answering data set. The first two models are based on sentence encoding methods, the next two models are based on attentive networks, while the others are based on compare-aggregate networks.

1) **SINN:** Yang and Kao (2020) proposed the model that applied self-attention based on RNN and CNN for sentence encoding.
2) **DiSAN:** Shen et al. (2018) proposed the model that used directional self-attention for encoding, and compressed features with multi-dimensional self-attention.
3) **ABCNN:** Yin et al. (2016) proposed the model that computed the attention matrix before and after convolution for modeling sentence pairs.
4) **DRCN:** Kim et al. (2019) proposed the model that used stacked RNN and co-attentive features to enhance representation.
5) **MULT:** Wang and Jiang (2017) presented the model that performed word-level matching by element-wise multiplication and aggregated by CNN.
6) **CAFE:** Tay et al. (2017) presented the model that adopted factorization machines to compress the alignment vectors into scalar features for augmenting the word representations.
7) **MCAN:** Tay et al. (2018) presented the model that adopted several attention variants and performed multiple comparison operators.
8) **ESIM:** Chen et al. (2017) presented the sequential inference model using chain LSTMs.

4.4 Results

Table 1 and Table 2 report the overall performance of the different models and the performance comparison of each category.

Model	Acc(%)
SINN	75.21
DiSAN	81.69
ABCNN	73.78
DRCN	75.31
MULT	81.19
CAFE	81.27
MCAN	81.63
ESIM	81.79
IMQAMM	**83.02**

Table 1: Test accuracy on Mongolian question answering data set.

Model	Matched			Mismatched		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
SINN	72.53	81.17	76.60	78.62	69.25	73.64
DiSAN	82.73	80.11	81.40	80.72	83.27	81.98
ABCNN	71.10	80.11	75.34	77.23	67.44	72.00
DRCN	77.53	71.29	74.28	73.43	79.33	76.27
MULT	82.80	78.74	80.72	79.73	83.64	81.64
CAFE	80.97	81.74	81.35	81.57	80.79	81.18
MCAN	80.78	83.01	81.88	82.53	80.25	81.37
ESIM	82.23	81.10	81.66	81.36	82.47	81.91
IMQAMM	83.68	82.04	**82.85**	82.39	84.00	**83.18**

Table 2: Performance comparison of different methods on test set.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 896-907, Nanchang, China, October 14 - 16, 2022.

Table 1 presents that our Interactive Mongolian Question Answer Matching Model (IMQAMM) achieves an accuracy of 83.02%, which has already outperformed all the baseline models. Notably, IMQAMM has an improvement of about 1.23% compared to the highest ESIM in the baseline models. It shows that the introduction of multi-cast attention is helpful. IMQAMM outperforms MCAN and CAFE by 1.39% and 1.75%, which proves the significance of sequence enhancement. Compared with DRCN and ABCNN, the five models at the bottom of Table 1 have significant improvements, thus compare-aggregate networks can provide more interactive information than attentive networks in this task. And the performance of our model is higher than SINN and DiSAN, which indicates that our interactive model is better than the sentence encoding based methods on Mongolian question answering data set.

Table 2 presents the performance comparison of different methods. The improvements of IMQAMM over the highest ESIM on the matched F1 score and mismatched F1 score are 1.19% and 1.27%. Compared with all the baseline methods, our IMQAMM is competitive in each category.

4.5 Ablation Study

As shown in Table 3, we conduct an ablation study to analyze the influence of each component. We remove three parts from IMQAMM to examine the influence: 1) Multi-Cast Attention. 2) Morpheme Matching. 3) Interaction Matching.

According to the results of ablation experiments in Table 3, we can see the key components of our model. Firstly, when removing Multi-Cast Attention, the accuracy decreases by 0.38%, which proves that Multi-Cast Attention is helpful for our model. Secondly, we find that Morpheme Matching is necessary for our model. When we remove it, the accuracy is reduced by 0.6%. Finally, when removing Interaction Matching, we can observe that the performance of our model drops dramatically. The accuracy drops from 83.02% to 80.52%. This result shows that Interaction Matching is crucial for our model.

Model	Acc(%)
IMQAMM	**83.02**
w/o Multi-Cast Attention	82.64
w/o Morpheme Matching	82.42
w/o Interaction Matching	80.52

Table 3: Ablation study on Mongolian question answering data set.

5 Conclusion

In this paper, we propose an Interactive Mongolian Question Answer Matching Model (IMQAMM), which mainly combines interactive information enhancement and max-mean pooling matching. First of all, we make some transformations to traditional Mongolian language and introduce the morpheme vectors. Second, we enhance the sequence representation by concatenating a series of feature vectors. Third, the multi-cast attention is introduced to alleviate the data-sparse problem caused by complex Mongolian morphological structures. Finally, the max-mean pooling matching strategy is applied to match question-answer pairs in two directions. Experimental results show that our model performed well on the Mongolian question answering data set.

However, there is still a lot of room for improvement. In the future work, we will consider using the pre-trained language model BERT to get a better initialization, which may help improve the performance of our model.

Acknowledgements

This work is supported by National Key R&D Program (Nos. 2018YFE0122900); National Natural Science Foundation of China (Nos. 62066033, 61773224); Inner Mongolia Applied Technology Research and Development Fund Project (Nos. 2019GG372, 2020GG0046, 2021GG0158, 2020PT0002); Inner Mongolia Achievement Transformation Project (Nos. 2019CG028); Inner Mongolia Natural Science Foundation (2020BS06001); Inner Mongolia Autonomous Region Higher Education Science and

Technology Research Project (NJZY20008); Inner Mongolia Autonomous Region Overseas Students Innovation and Entrepreneurship Startup Program; Inner Mongolia Discipline Inspection and Supervision Big Data Laboratory Open Project. We are grateful for the useful suggestions from the anonymous reviewers.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1657–1668.

Yichen Gong, Heng Luo and Jian Zhang. 2017. Natural Language Inference over Interaction Space. *arXiv preprint arXiv:1709.04348.*

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Seonhoon Kim, Inho Kang and Nojun Kwak 2019. Semantic Sentence Matching with Densely-Connected Recurrent and Co-Attentive Information. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6586–6593.

Diederik P. Kingma and Jimmy Lei Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Jiasen Lu, Jianwei Yang, Dhruv Batra and Devi Parikh. 2017. Hierarchical Question-Image Co-Attention for Visual Question Answering. *arXiv preprint arXiv:1606.00061.*

Min Lu, Feilong Bao, Guanglai Gao, Weihua Wang, and Hui Zhang. 2019. An automatic spelling correction method for classical mongolian. In *International Conference on Knowledge Science, Engineering and Management*, pages 201–214.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity. In *Thirtieth AAAI Conference on Artificial Intelligence.*

H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song and R. Ward. 2014. Semantic Modelling with Long-Short-Term Memory for Information Retrieval. *arXiv preprint arXiv:1412.6629.*

Ankur P. Parikh, Oscar Täckström, Dipanjan Das and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. *arXiv preprint arXiv:1606.01933.*

Steffen Rendle. 2010. Factorization machines. In *IEEE International conference on data mining*, pages 995–1000.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng and Grégoire Mesnil. 2014. A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 101–110.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan and Chengqi Zhang. 2018. DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence.*

Ming Tan, Cicero dos Santos, Bing Xiang and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 464–473.

Yi Tay, Luu Anh Tuan and Siu Cheung Hui. 2017. Compare, Compress and Propagate: Enhancing Neural Architectures with Alignment Factorization for Natural Language Inference. *arXiv preprint arXiv:1801.00102.*

Yi Tay, Luu Anh Tuan and Siu Cheung Hui. 2018. Multi-Cast Attention Networks for Retrieval-based Question Answering and Response Prediction. *arXiv preprint arXiv:1806.00778.*

Weihua Wang, Feilong Bao and Guanglai Gao. 2015. Mongolian Named Entity Recognition using Suffixes Segmentation. In *2015 International Conference on Asian Language Processing (IALP)*, pages 169–172.

Shuohang Wang and Jing Jiang. 2017. A compare-aggregate model for matching text sequences. In *proceedings of 5th International Conference on Learning Representations, ICLR 2017.*

Zhiguo Wang, Wael Hamza and Radu Florian. 2017. Bilateral Multi-Perspective Matching for Natural Language Sentences. *arXiv preprint arXiv:1702.03814.*

Weihua Wang, Feilong Bao, and Guanglai Gao. 2019. Learning morpheme representation for mongolian named entity recognition. In *Neural Processing Letters*, 50(3): 2647–2664.

Kai-Chou Yang and Hung-Yu Kao. 2020. Generalize Sentence Representation with Self-Inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, paegs: 9394–9401.

Wenpeng Yin, Hinrich Schütze, Bing Xiang and Bowen Zhou. 2016. ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. In *Transactions of the Association for Computational Linguistics*, 4: 259–272.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 896-907, Nanchang, China, October 14 - 16, 2022.