Matching Questions and Answers in Dialogues from Online Medical Forums

Anonymous EMNLP-IJCNLP submission

Abstract

Two-party question-motivated dialogues from online medical forums are rich resources for question-answer pairs. Matching QA relations between two utterances is not only the first step for analyzing dialogue structures, but also valuable for training dialogue systems. This paper presents an upgraded pairwise matching model with consideration of dialogue context and distance information. Two parallel attention mechanisms are used to capture the information flow between two parties, and an one-hot vector with fixed dimension is used to encode the distance before the output layer. Given the scores computed by the trained model between each non-question utterance with its candidate questions, a gated greedy decoding algorithm is adopted for final predictions. We create a dataset with 1,000 annotated dialogues demonstrate that our proposed model outperforms the state-of-the-art and controlled baselines, which is significantly better on matching long-distance QA pairs.

1 Introduction

Question motivated dialogues are very common in daily life and they are rich resources for question-answer (QA) pairs. Dialogues about online medical and health consultation is a typical example. In such online forums, both the doctor and the patient in a dialogue tend to ask and answer questions to narrow down the information gap and reach the final diagnosis or recommendations. Matching QA pairs from such resources is an important research task.

QA matching is an important part of analyzing discourse structures for dialogues and dialogue comprehension. Asher et.al (2016) shows that in online dialogues where participants are prompted to communicate with others to achieve their goals, 24.1% of the relations between elementary discourse units are QA pairs. Questions and answers

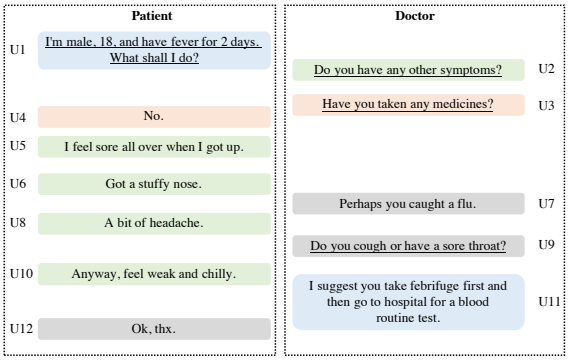

Figure 1: Questions and answers matching in dialogues from an online health forum. The identified pairs are painted in the same color and questions are underlined.

are also considerable components of dialogue acts (Stolcke et al., 2000), which are key features for doing dialogue summarization and decision detection (Fernández et al., 2008). Besides, figuring out the QA relations between these utterances can provide question answering models (Ji et al., 2014; Vinyals and Le, 2015; Cui et al., 2017) with more high-quality QA pairs and help with the exploration of proactive questioning (Yan et al., 2017).

However, many challenges exist. Due to the network delay, the differences in typing speed, or delays caused by distraction from either party, the dialogue sequences are always mix-matched. Also, sometimes a long and complete answer may be broken up into separated utterances such as U5, U6, U8 and U10 in Figure 1. Besides questions and answers, a dialogue also contains less informative chit-chats, distinguished from real answers and statements. Moreover,"personalized" orthography, ellipsis and abbreviations, and missing punctuations are all difficulties to parsing online dialogues.

In this work, we focus on the task of matching questions and answers in two-party multi-turn dialogues. We found that the distance between the

question and its corresponding answer isn't only influenced by the mixed and fragmented factors mentioned above, but also effected by its nature. Some questions can be answered directly based on personal knowledge, such as U3, while other questions can not. For instance, when a patient asks questions such as "what's wrong with me" or "what should I do" just like in U1, the doctor often has to ask follow-up questions {U2, U3} to seek for additional information in order to give the final diagnosis or recommendation (U11) after several turns of communications. We call this kind of QA pairs as incremental QA. Such QA pairs reflect the main idea of question motivated dialogues or sub-dialogues, significantly important for dialogue comprehension. Nevertheless, the answer is inherently far from the question for incremental QA pairs (distance ≥ 3 [1]), which aggravated the difficulty of matching such pairs.

Roughly, we can category the QA pairs into different classes according to the distance between them. When the distance is less than 3, we call it short distance QA pairs (SQA); and when the distance is larger than 3, we call it long distance QA pairs (LQA). It is obvious that matching SQAs is much difficult than matching LQAs. We assume that a two-party multi-turn dialogue contains two types of utterances, questions (Q) and non-questions (NQ), which are labeled in advance[2]. Our task is to identify all answers from the set of NQs to a given Q. Recent work by He et al. (2019) considers a slightly different QA alignment problem where one answer can match multiple questions. We argue that if a question is asked twice in two utterances, the answer should be matched to the second, closer question and the first one is considered missed. By our definition, a Q can match nothing (U9) or several NQs (U2). From the viewpoint of a NQ, it is either or not matched with a question (such as U7).

Previous methods on the task (Ding et al., 2008; Du et al., 2017; Jiang et al., 2018) suffer from a major weaknesses: while classifying a pair of sentences, they ignore the context of the sentences in the dialogue. Meanwhile, the pre-defined features such as question words and answer words, are already implied by the Q and NQ labels in our definition and hence are not suitable for our task. He et al. (2019) improves the above methods with a recurrent pointer network (RPN) model that takes the whole dialogue as an input sample. Their model was evaluated on a closed-source customer service dialogue dataset. Although their model makes use of the context, they treat every utterance in the context equally with RNN-based networks which fails to capture the influences between turns especially with long distance. It also encodes the distance information implicitly which downplays the effect of distance between the utterances. None of the above approaches perform well with LQA pairs.

In this paper, we bring the dialogue history information into the simple pairwise model. For a given pair of Q and NQ to be matched, the dialogue history refers to the utterances between the Q and NQ. The critical part of our model is two parallel attention mechanisms that combine dialogue history in an interleaving way. Compared with the state-of-the-arts and controlled baselines, our proposed models increase the F1-score from 75.20% to 77.43% for overall performance. The F1-score for LQAs is increased from 32.35% to 40.44% and 14.88% to 24.80% when distance is 4 and ≥ 5 respectively.

In brief, our main contributions are as follows:

- We focus on the task of matching a question and its answers in two-party multi-turn dialogues, and we are the first to consider QA pairs different distance categories to the best of our knowledge. (Section 2)

- We bring dialogue history and distance information into basic pairwise models. We show that distance is an important feature for matching QA pairs in dialogues which should be encoded explicitly. History between Q and A can be effectively captured by a parallel attention mechanism. (Section 3)

- Based on the fact there is no open source for such mixed dialogues with LQAs, we construct a dataset on online health counselling dialogues. The experimental results in Section 4 show that our proposed methods can effectively find the QA relations with the highest performance on this dataset, especially on LQAs. Based on proposed approach,

[1] There is at least one follow-up question and one corresponding anwer between the question and answer of an incremental QA pair. So the distance for such QA pairs is larger or equal to 3.

[2] Labeling utterances to Q and NQ can be done with a simple neural-based classifier, and the accuracy is more than 96%.

we further automatically label QA relations for around 160,000 raw dialogues crawled online. Both datasets will be published for research use.

2 Task Description

In this section, we give a formal problem definition of our task and introduce the dataset we created.

2.1 Problem Definition

Our work aims at identifying QA relations by matching Q and NQ in two-party dialogues, which can be regarded as a turn matching problem. Given a dialogue sequence with T turns:

$$[(R_1, L_1, U_1), (R_2, L_2, U_2), ..., (R_T, L_T, U_T)]$$

where $R \in \{P, D\}$ and $L \in \{Q, NQ\}$. P and D represent the role of two parties. Q and NQ categories all the turns into questions and non-questions.

Our job is to match each (Q, U_i) with corresponding (NQ, U_j), where:

$$\begin{aligned} j > i \quad 1 \leq i, j \leq T \\ R_i \neq R_j \end{aligned} \tag{1}$$

The *distance* of a Q-NQ pair is equal to $j - i$. And the *history* we considered in our approach are the turns $\{U_{i+1}, U_{i+2}..., U_{j-1}\}$ which located between the given Q and NQ.

2.2 Dataset Creation

Although many dialogue or QA datasets are published, QA pairs between two parties are always paired or in successive turns. Other dataset used in previous related works are closed due to different reasons, such as the customer service dialogue dataset used in He's paper(He et al., 2019). Wei et.al (2018) published their dialogue dataset collected from an online forum. However, their work focuses on the dialogue policy learning and doesn't preserve the original utterances.

Hence, we create a new dataset suitable for this task. Nearly 160,000 distinct dialogues are collected from an online health forum[3]. All the personal information in dialogues are removed in advance by this website. After some basic data cleaning methods such as deleting the unknown characters and irrelavant sentences like "Please pay ** coins to continue consultation", we randomly labelled 1000 two-party multi-turn dialogues with Q (question), A (answer) and O (others). Since the occurrence of a turn being both an answer and a question is only 0.24% by sampling, the annotators were asked to regard such turns as questions. The Fleiss' Kappa of our annotation between three annotators was 0.75.

For each dialogue, it owns an average of 19.68 doctors' turns and 17.32 patients' turns. Most turns are made up of less than one complete sentence, so the number of words for each turn is less than 10 words for questions, answers and others [4]. There are totally 21.9% questions which has no answers, 22.7% questions have more than one answer and the rest questions have the only answer. For each question that has answers, it is matched to 1.41 answers on average.

The annotated dialogues are divided into training set, development set and test set by 7:1:2. The number of QA pairs fall in each distance bin are shown in Table 1.

Dataset \ Distance	1	2	3	4	≥ 5
Training	3439	2068	1029	450	554
Development	454	331	167	76	99
Test	947	592	274	136	168

Table 1: The distribution on QA pairs of variable distances.

To meet the need for pairwise models which score the probability of each Q-NQ pair being a QA pair, we reconstructed the labeled dialogues into Q-NQ pairs with distance, history and binary golden label. A NQ from a person is paired with every earlier Q from the other person. If the pair exits QA relation, this pair is labeled as True(T). Otherwise, it is labeled as False(F). The distribution of positive and negative data on three datasets are shown in Table 2.

Label \ Dataset	Training	Development	Test
True	7540	1226	2116
False	80631	14889	23893

Table 2: The distribution of positive and negative Q-NQ pairs on three datasets.

[3] https://www.120ask.com/

[4] There are on average 9.80 questions with 8.89 words, 10.78 answers with 6.62 words, 16.41 casual chit chats with 6.99 words in each session according to annotated dialogues.

3 Approach

Inspired by match-LSTM with word-by-word attention (Wang and Jiang, 2015), we propose a parallel attention-based neural network model for QA matching in multi-turn dialogues between two parties. The model consists of four components: Given a Q-NQ pair with its distance and history turns, **Sentence Encoder** transforms the natural language turns into sentence-level embeddings. **History Attention** combines history turns based on two parallel attention mechanisms in an interleaving way. **Match-LSTM** is used to compare the processed sentence pair word by word. Finally, **Prediction Layer** brings the distance information into consideration and calculate the final alignment probability. After calculating the probability for all Q-NQ pairs in a dialogue, a **gated greedy decoding algorithm** is implemented for final matching decisions.

3.1 Sentence Encoder

Given an input turn as a sequence of words in natural language, we generate a neural representation using a LSTM (Gers et al., 1999). The input layer consists of pretrained word embeddings of the words which is fed into a single hidden layer. The output of all the hidden states or the last hidden state is regarded as the sentence-level embedding for this turn.

With the sentence encoder, we can get the neural representations for a Q-NQ pair and its corresponding input as follows:

$$\begin{aligned} Q &= \{h_i^q\}_{t=1}^N \\ NQ &= \{h_i^p\}_{t=1}^M \\ H_{RQ} &= \{d_t^q\}_{t=1}^A \\ H_{RNQ} &= \{d_t^p\}_{t=1}^B \end{aligned} \tag{2}$$

where H_{RQ} and H_{RNQ} represents the history turns with the same role label as Q and NQ respectively. Here, we divide the history turns into two parts to support for the idea of interleaving attentions in the following subsection.

The intuition for using different granularity of sentence embeddings is that we hope to keep for information for more important turns. So, in order to calculate the matching score between a Q-NQ pair, we preserve all the hidden states for Q and NQ. And the last hidden state of each turn in history is used to provide auxiliary information.

3.2 History Attention

To improve the prediction for each Q-NQ pair, naturally we will take advantage of the dialogue context, especially the turns between Q and NQ. If Q has been obviously answered by a incomplete turn before NQ, we will further figure out whether NQ is a supplementary answer. If there exists another Q which is closer to the NQ with distance and semantics, the probability of matching the given Q-NQ pair will definitely reduces. If the model can capture these features, it's more likely to matching the LQAs such as {U2,U10}.

Besides, the question and corresponding answer turns are definitely contributed by different parties. In other words, the QA relations in a dialogue is focusing on the process of narrowing down the information gap between two parties, where the information interaction between parties is critical. So, we further divide the history into two parts: H_{RQ} and H_{RNQ} by different role labels as mentioned above. Q is expected to interact with H_{RNQ} while NQ is expect to interact with H_{RQ}, which is called as "interleaving".

Borrowing the idea from Wang et al. (2017), we use two attention mechanisms to incorporate the history information into Q and NQ individually in a unified manner. For example, when dealing with the Q-NQ pair $\{U2, U10\}$, $H_{RQ} = U3, U7, U9$ and $H_{RNQ} = U4, U5, U6, U8$. The neural representation of $U2$ is attended to H_{RNQ} and $U10$ is attended to H_{RQ} in parallel. Take the Q and H_{RNQ} as an example. The question containing historical information via soft alignment of words in the question $Q = [h_1^q, h_2^q, ..., h_N^q]$ and history sentences $H_{RP} = \{d_1^p, d_2^p, ..., d_B^p\}$ can be obtained as follows (see Part I in Figure 2):

$$u_t^q = [h_t^q, c_t^q] \tag{3}$$

where $c_t^q = att(h_t^q, H_{RP})$ is an attention pooling vector of the whole history(H_{RP}):

$$\begin{aligned} s_j^t &= v^T tanh(W_Q h_t^q + W_H d_j^p) \\ a_i^t &= exp(s_i^t) / \sum_{j=1}^{B} exp(s_j^t) \\ c_t^q &= \sum_{i=1}^{B} a_i^t d_i^p \end{aligned} \tag{4}$$

Each word representation in Q dynamically incorporates aggregated matching information from the history H_{RNQ}.

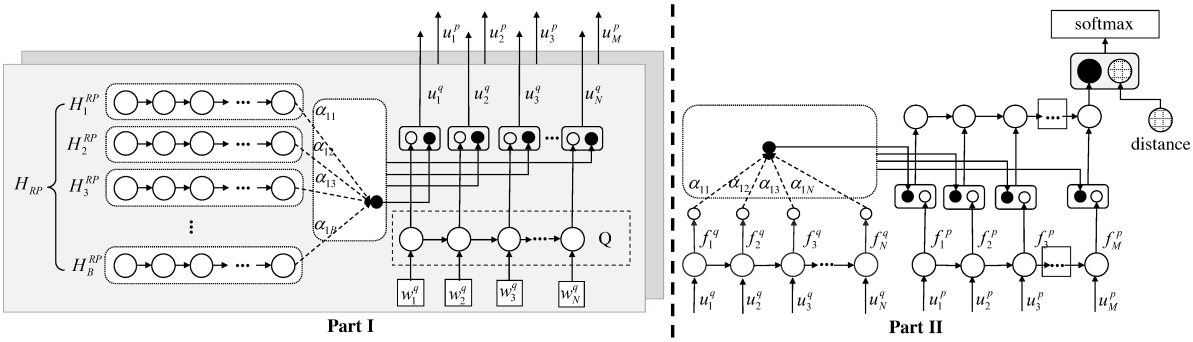

Figure 2: The architecture of the proposed match-LSTM based model with parallel attention mechanisms. **Part I:** History Attention. **Part II:** Match-LSTM Component

Finally, we get the question and non-question as $Q' = [u_1^q, u_2^q, ..., u_N^q]$ and $NQ' = [u_1^p, u_2^p, ..., u_M^p]$. Each word representation for both utterance not only represent the sentence meaning but also contain dialogue context. The comparison of the impacts on different choices of turns in history and the part of the history to attend to will be discussed in 5.

3.3 Match-LSTM

We follow the work of Wang and Jiang (2015) and adopt match-LSTM to capture the features between the processed Q' and NQ' word by word.

Again, an one-layer LSTM mentioned in 3.1 is implemented encode the processed representations for question and non-question. Then, we get $Q'' = \{f_t^q\}_{t=1}^N$ and $NQ'' = \{f_t^p\}_{t=1}^M$. When looking through the non-question, we introduce a series of attention-weighted combinations of the hidden states of the question, where each combination is for a particular word in the non-question. The sentence-pair representation $P = \{p_t\}_{t=1}^M$ is calculated with attention mechanism as follows (see Part II in Figure 2):

$$p_t = LSTM(p_{t-1}, [f_t^p, c_t]) \tag{5}$$

where $c_t = att(Q, f_t^p, p_{t-1})$ is an attention pooling vector of the whole question(Q):

$$\begin{aligned} s_j^t &= v^T tanh(W_{NQ} f_t^p + W_Q f_j^q + W_p p_{t-1}) \\ a_i^t &= exp(s_i^t) / \sum_{j=1}^{N} exp(s_j^t) \\ c_t &= \sum_{i=1}^{N} a_i^t f_i^q \end{aligned} \tag{6}$$

Finally, we use p_M to represent the whole pair which is used for predicting the final result.

3.4 Prediction Layer

At the last step, we use a fully-connected (FC) layer with softmax to do binary classification, which indicates whether this pair of utterance has QA relation.

Based on the fact that the distance is an extremely important feature when matching QA pairs (This will be shown in Section 5), we explicitly add the distance at the end of our model to preserve this information to a great extent. The distance d is defined as a 10-dimension one-hot vector, where the index of the position with 1 refers to the absolution distance. And the list dimension is 1 when $d \geq 10$.

Finally, the probability is calculated as follows:

$$\begin{aligned} FC &= W[p_M, d] + b \\ P(Q, NQ) &= Softmax(FC) \end{aligned} \tag{7}$$

3.5 Gated Greedy Decoding Algorithm

For a NQ, it has a matching probability with each Q ($Q \in \Omega$) before it. The final prediction is chosen as follows:

$$Q^\star = \begin{cases} \arg\max_{Q \in \Omega} P(Q, NQ) & P(Q^\star, NQ) \geq 0.5 \\ \emptyset & otherwise \end{cases} \tag{8}$$

Here, we select the Q-NQ with the maximum probability which exceeds 0.5 as QA pairs. The threshold 0.5 is chosen naturally since the model is actually a two-class classifier.

4 Experiment Setup

In this section, we explain the baselines and the ablations of our full model. Next, we define the evaluation metrics. Finally, we show the details of hyperparameters in our model.

4.1 Baseline

4.1.1 Greedy strategy (GD)

A simple baseline *Greedy* is that, when a question is posed by RQ, we can directly match the following several NQs said by $\overline{RQ}$ as the answers. It stops when meeting another Q or a turn said by RQ. Specificly, GD1 select only one satisfied answer, and GDn select multiple satisfied answers. The rules with *Jump* (**J**) is that, we can skip the non-question sentence said by RQ when matching the NQs.

4.1.2 Distance

A simple model has a 10-dimension one-hot distance vector as input into a fully-connected layer and outputs the score for each Q-NQ pair.

4.1.3 Word-by-word match LSTM (mLSTM)

This model is proposed by Wang et al. (2015), used for natural language inference. They perform word-by-word matching based on an attention mechanism, with the aims of predicting the relation between two sentences.

4.1.4 Recurrent Pointer Network (RPN)

The model proposed by He et al. (2019) is implemented with some modifications to fit our task. We use two parallel RPN to distinguish questions from two parties. Beside, we compare their proposed classification and regression loss and choose the one that performs best on our test set.

4.2 Our Models

We adjust our complete model by wiping off some of its components or using different inputs and name the following models:

- **Distance Model (DID)** wipe off the history information. It directly put Q-NQ pairs with the distances into the Part II of the complete model in Figure 2.
- **History Model (HTY)** only wipe off the distance information at the last prediction layer.
- **History-Distance Model (HDM)** is the complete model we have explained in Section 3.

4.3 Evaluation Metrics

Once we have identified all of the QA pairs, we count the true positive, false positive and false negative for each question. Micro-averaging precision (**P**), recall (**R**) and F1-score (**F1**) are calculated for equally treating all the questions in test dataset to measure the quality of predicted QA pairs.

4.4 Experiment Set-up

We use Jieba[5] to do Chinese word segmentation on all the utterances and pre-train the 100 dimentional word embeddings with Skip-gram model (Mikolov et al., 2013). For our proposed models, we use LSTM with hidden state size equaling 128 and 256 for Part I and Part II of the model respectively. We adopt Adam optimizer with 0.001 learning rate and 0.3 dropout. Learning rate decay is 0.95 and the training process terminates if the loss stop reducing for 3 epochs. The experimental results are averaged over three runs.

5 Results and Analysis

In this section, we first show the end-to-end results on QA matching and accuracy on variable distances. Then we do the ablation tests to show the specific architectural decisions.

5.1 Overall Performance

The main results of different model are shown in Table 3. The last row shows the human performance, which can be regarded as the upper bound of this task due to some ambiguities of answers.

Models	P	R	F1
GD1	69.84	44.73	54.53
GDN	70.03	69.11	69.57
GD1+J	70.38	50.40	58.74
GDN+J	51.47	82.90	63.51
Distance	71.57	69.34	70.44
mLSTM	58.17	4.20	7.84
RPN	73.88	76.57	75.20
DID	78.46	70.34	74.70
HTY	75.40	76.42	75.90
HDM	76.44	78.44	**77.43**
Human	85.11	84.21	84.66

Table 3: The end-to-end performance of all methods on test dataset.

From the Table 3, we get following conclusions:

- The results of the rule-based methods are not bad, which indicate that questions are followed by their answers in most cases. The GDN increases the F1-score and accuracy to 69.57% and 79.39% compared with Greedy-1 because it can solve the case of simple FQA. For GDN+J, the recall obtains the best

[5] https://github.com/fxsjy/jieba

score among all the methods while accuracy and F1-score reduce. The reason is that GDN tends to match NQ with Q as much as possible, so many chit chats will be regarded as answers and F-accuracy reduces.

- Model mLSTM fails while the Distance obtains outstanding performance. It shows that it is difficult to solve the QA matching problem with only two short texts. The word distribution between the questions and answers are quite different without back knowledge. Besides, the distance information is significantly important when identifying QA relations in dialogues. People tend to answer the question at the moment they see it except in IQA condition.

- RPN obtains great improvements. It mainly benefits from taking the dialogue session as a whole which contains all the information in a session. However, it can't effectively identify the useful features and may bring more noise into the model.

- Our proposed models achieve the best results compared with above models while the HDM increases the F1-score to 77.79% and the accuracy to 85.28%. Although the recall and precision of both models are not better than GDN+J and DM respectively, the overall quantity and quality of QA pairs we identified are the best based on the highest F1-score. In addition, the results demonstrate that QA matching not only depends on the distance but also relies on the history information. Our model successully combine the dialogue context into the basic pairwise model.

5.2 Variable Distance Matching

Since distance is a really important feature for QA matching, we also compute the accuracy on QA pairs with different distances. The results are shown in Table 4.

As for rule-based methods, it is no doubt that they will achieve the accuracy with 100% when distance=1. As distance getting longer, the performance of GDN and GDN+J surpass GD1 and GD1+J. Although GDN+J obtains the highest accuracy among these models, according to the Table 3, the quality of QA pairs identified by this method is bad because of redundant answers to the questions. Distance model fails when the distance is longer than 2.

Models	1	2	3	4	≥ 5
RPN	95.78	80.57	61.31	32.35	14.88
DID	96.23	**89.13**	17.03	2.45	0.0
HTY	94.37	78.89	57.42	38.48	**28.17**
HDM	95.99	83.16	59.37	**40.44**	24.80

Table 4: The matching accuracy(%) of Q-NQ pairs on variable distances.

Here, we mainly compare the models which guarantee the overall accuracy on the test set. Their results are listed in the last four rows in Table 4. Comparing these four neural-based models, although they can't guarantee 100% accuracy on distance=1, their accuracy is still comparable. It shows that the neural-based models with dialogue context information including RPN, H-M and DHM perform well with distance longer than 3. Although RPN takes the whole session into consideration, the accuracy with distance=4 and ≥ 5 obviously lower than HM. It indicates that RPN can not work well in long-distance situations while our model achieves. When bring distance information into our model, it harms the accuracy on QA pairs with longer distance while increases with shorter distance.

5.3 Ablation Tests

We analyze our model decisions in following two aspects:

Analysis on the choice of turns in history.

To evaluate the effectiveness of choosing the turns between Q and NQ as the history, we devised the following two variants of the HDM model for comparison:

- **Qhistory Model (QH)** has the same structure of HDM where the history is all the turns before Q.

- **Ahistory Model (AH)** has the same structure of HDM where the history is all the turns before NQ.

The main results with different choice as history is shown in Table 5[6]. Our final model (HDM) outperforms QH and AH, indicating that the turns between Q and NQ are significant when figuring out

[6]Due to the space limitation, we only listed the significantly different results. The complete results is shown in Appendix

the relation between Q-NQ pair. The turns before Q is actually not quite important when matching Q and NQ. Although there is an overlap between the history we defined and the turns before NQ, it brings much more noises than improving the model performances. This demonstrate that the definition of history as the turns between Q and NQ is both reasonable and effective.

Models	F1	3	4	≥ 5
QH	73.76	12.04	10.29	0.0
AH	74.18	20.80	13.24	8.33
HDM	**77.43**	**59.37**	**40.44**	**24.80**

Table 5: The matching results on different choices of history.

Analysis on the ways of aggregating history by attention mechanisms.

To evaluate the effectiveness of the interleaving way we aggregating the history, we also devised two variants of the HDM model for comparison:

- **Non-interleaving Model (NON)** has the same structure of HDM where Q attends to H_{RQ} and NQ attends to H_{RNQ}.
- **Same Model (SAME)** has the same structure of HDM where Q and NQ attends to the same history $H_{RQ} \bigcup H_{RNQ}$.

The main result in Table 6 reveals that our choice of separate the history by role label and interleaving attentions does works. The full model (HDM) outperforms both NON and Same. When make the use of both turns from both parties, it confused the model with too much turns to consider. The results of NON is slightly better than SAME especially on LQAs maybe due to the better representations of the flow of semantic information on individual parties. However, QA relations focus more on interactions between parties.

Models	F1	3	4	≥ 5
NON	75.49	56.93	38.24	19.05
SAME	75.38	54.01	27.94	8.33
HDM	**77.43**	**59.37**	**40.44**	**24.80**

Table 6: The matching results on different ways of aggregating history.

6 Related Work

Detection of QA pairs from online discussions has been widely researched these years. Shrestha and Mckeown (2004) learned rules using Ripper for detecting QA pairs in email conversations. Ding et al. (2008), Kim et al. (2010) and Catherine et al. (2012) applied the supervised learning method including conditional random field and support vector machine. Cong et al. (2008) proposed an unsupervised method combining graph knowledge to solve the task. Catherine et al. (2013) proposed semi-supervised approaches which require little training data. He et al. (2019) used the pointer network to find QA pairs in Chinese customer service. However, the tasks mentioned above are all different from ours. We identify QA pairs on the Chinese health forum, and focus on long-distance QA pairs. Besides, our dialogue is constrained between two roles and they can both ask questions and give answers.

There exists several methods in other tasks which can be adapted to our QA matching problem. Feature-based method is popular for solving many NLP problems. In the work of Ding et al. (2008), Wang et al. (2010) and Du et al. (2017), they examined lexical and semantic features in two sentences for QA matching. However, the features such as common question words and roles have already been obviously annotated in our data. Besides, other features such as special word occurrence or time stamp are inaccessible here. According to the data, we considered the distance as the most important feature and implemented this feature-based method as one baseline. Recent researches using deep neural networks have increased a lot. He and Lin (2016) and Liu et al. (2016) used the sentence pair interaction approach which takes word alignment and interactions between the sentence pair into account. Attention mechanism was also added for performance improvement (Rocktäschel et al., 2015; Wang and Jiang, 2015; Chen et al., 2016). We also use word alignment and interactions to calculate the QA similarity. Specially, we adopt attention mechanism to solve the LQA cases.

There are other kinds of alignment problems such as temporal sequences alignment. Video-text alignment is one of the temporal assignment or sequence alignment problems. Previous work automatically provides a time (frame) stamp for every sentence to align the two modalities such as (Bojanowski et al., 2015) and (Dogan et al., 2018). Bojanowski et al. (2015) extended prior work by including the alignment of actions with

verbs and aligned text with complex videos. Dynamic time warping (DTW) is anothor algorithm for measuring similarity between two temporal sequences. It's also widely used in Video-text alignment task (Dogan et al., 2018), speech recognition task (Vintsyuk, 1968).

7 Conclusion

In this paper, we focus on identifying QA pairs in two-party multi-turn online dialogues based on turns with Q or NQ labels. Our proposed models achieve the best overall and specifically performs well on LQAs. We also discuss the model decisions of using two parallel attention mechanism in an interleaving way and the definition of history. In the future, we are going to focus more on LQA matching, especially the incremental cases. Taking domain outside knowledge into consideration is another research direction for utterances matching task, especially for medical dialogues.

References

Nicholas Asher, Julie Hunter, Mathieu Morey, Farah Benamara, and Stergos Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: The stac corpus.

Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. 2015. Weakly-supervised alignment of video with text. In *Proceedings of ICCV*, pages 4462–4470.

Rose Catherine, Rashmi Gangadharaiah, Karthik Visweswariah, and Dinesh Raghu. 2013. Semi-supervised answer extraction from discussion forums. In *Proceedings of IJCNLP*, pages 1–9.

Rose Catherine, Amit Singh, Rashmi Gangadharaiah, Dinesh Raghu, and Karthik Visweswariah. 2012. Does similarity matter? the case of answer extraction from technical discussion forums. *Proceedings of COLING: Posters*, pages 175–184.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Gao Cong, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun. 2008. Finding question-answer pairs from online forums. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 467–474.

Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, and Ming Zhou. 2017. Superagent: A customer service chatbot for e-commerce websites. *Proceedings of ACL 2017, System Demonstrations*, pages 97–102.

Shilin Ding, Gao Cong, Chin-Yew Lin, and Xiaoyan Zhu. 2008. Using conditional random fields to extract contexts and answers of questions from online forums. *Proceedings of ACL-08: HLT*, pages 710–718.

Pelin Dogan, Boyang Li, Leonid Sigal, and Markus Gross. 2018. A neural multi-sequence alignment technique (neumatch). In *Proceedings of CVPR*, pages 8749–8758.

Wenchao Du, Pascal Poupart, and Wei Xu. 2017. Discovering conversational dependencies between messages in dialogs. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Raquel Fernández, Matthew Frampton, Patrick Ehlen, Matthew Purver, and Stanley Peters. 2008. Modelling and detecting decisions in multi-party dialogue. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, pages 156–163. Association for Computational Linguistics.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with LSTM.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of NAACL-HLT*, pages 937–948.

Shizhu He, Kang Liu, and Weiting An. 2019. Learning to align question and answer utterances in customer service conversation with recurrent pointer networks.

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988*.

Jyun-Yu Jiang, Francine Chen, Yan-Ying Chen, and Wei Wang. 2018. Learning to disentangle interleaved conversational threads with a siamese hierarchical network and similarity ranking. In *Proceedings of NAACL-HLT 2018, Volume 1 (Long Papers)*, volume 1, pages 1812–1822.

Su Nam Kim, Li Wang, and Timothy Baldwin. 2010. Tagging and linking web forum posts. In *Proceedings of CoNLL*, pages 192–202.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Modelling interaction of sentence pair with coupled-lstms. *arXiv preprint arXiv:1605.05573*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Lokesh Shrestha and Kathleen McKeown. 2004. Detection of question-answer pairs in email conversations. In *Proceedings of the 20th international conference on Computational Linguistics*, page 889.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Taras K Vintsyuk. 1968. Speech discrimination by dynamic programming. *Cybernetics and Systems Analysis*, 4(1):52–57.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Baoxun Wang, Xiaolong Wang, Chengjie Sun, Bingquan Liu, and Lin Sun. 2010. Modeling semantic relevance for question-answer pairs in web social communities. In *Proceedings of ACL*, pages 1230–1238.

Shuohang Wang and Jing Jiang. 2015. Learning natural language inference with LSTM. *arXiv preprint arXiv:1512.08849*.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of ACL*, volume 1, pages 189–198.

Zhongyu Wei, Qianlong Liu, Baolin Peng, Huaixiao Tou, Ting Chen, Xuanjing Huang, Kam-Fai Wong, and Xiangying Dai. 2018. Task-oriented dialogue system for automatic diagnosis. In *Proceedings of ACL*, pages 201–207.

Zhao Yan, Nan Duan, Peng Chen, Ming Zhou, Jianshe Zhou, and Zhoujun Li. 2017. Building task-oriented dialogue systems for online shopping. In *Thirty-First AAAI Conference on Artificial Intelligence*.