MIE: A Medical Information Extractor towards Medical Dialogues

Yuanzhe Zhang[1], **Zhongtao Jiang**[1,2], **Tao Zhang**[1,2], **Shiwan Liu**[1,2], **Jiarun Cao**[3]*, **Kang Liu**[1,2], **Shengping Liu**[4] and **Jun Zhao**[1,2]

[1] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China

[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 100049, China

[3] National Centre for Text Mining, University of Manchester, Manchester, M1 7DN, United Kingdom

[4] Beijing Unisound Information Technology Co., Ltd, Beijing, 100028, China

{yzzhang, zhongtao.jiang, tao.zhang, shiwan.liu, kliu, jzhao}@nlpr.ia.ac.cn

jiarun.cao@manchester.ac.uk, liushengping@unisound.com

Abstract

Electronic Medical Records (EMRs) have become key components of modern medical care systems. Despite the merits of EMRs, many doctors suffer from writing them, which is time-consuming and tedious. We believe that automatically converting medical dialogues to EMRs can greatly reduce the burdens of doctors, and extracting information from medical dialogues is an essential step. To this end, we annotate online medical consultation dialogues in a window-sliding style, which is much easier than the sequential labeling annotation. We then propose a Medical Information Extractor (MIE) towards medical dialogues. MIE is able to extract mentioned symptoms, surgeries, tests, other information and their corresponding status. To tackle the particular challenges of the task, MIE uses a deep matching architecture, taking dialogue turn-interaction into account. The experimental results demonstrate MIE is a promising solution to extract medical information from doctor-patient dialogues. [1]

1 Introduction

With the advancement of the informatization process of the medical system, Electronic Medical Records (EMRs) are required by an increasing number of hospitals all around the world. Compared with conventional medical records, EMRs are easy to save and retrieve, which bring considerable convenience for both patients and doctors. Furthermore, EMRs allow medical researchers to investigate the implicit contents included, such as epidemiologic study and patient cohorts finding. Despite the advantages, most doctors complain that writing EMRs makes them exhausted (Wachter and Goldsmith, 2018). According to the study of Sinsky et al. (2016), physicians spend nearly two hours doing administrative work for every hour of face-time with patients, and the most time-consuming aspect is inputting EMRs.

We believe that automatically converting doctor-patient dialogues into EMRs can effectively remove the heavy burdens of doctors, making them more deliberate to communicate with their patients. One straightforward approach is the end-to-end learning, where more supervised data, i.e., dialogue-EMR pairs are needed. Unfortunately, such data is hard to acquire in medical domain due to the privacy policy. In this paper, We focus on extracting medical information from dialogues, which we think is an essential step for EMR generation.

Extracting information from medical dialogues is an emerging research field, and there are only few previous attempts. Finley et al. (2018) proposed an approach that consists of five stages to convert a clinical conversation to EMRs, but they do not describe the detail method. Du et al. (2019) also focused on extracting information from medical dialogues, and successfully defined a new task of extracting 186 symptoms and their corresponding status. The symptoms were relatively comprehensive, but they did not concern other key information like surgeries or tests. Lin et al. (2019) collected online medical dialogues to perform symptom recognition and symptom inference, i.e., inference the status of the recognized symptoms. They also used the sequential labeling method, incorporated global attention and introduced a static symptom graph.

There are two main distinctive challenges for tackling doctor-patient dialogues: a) Oral expres-

*Contribution during internship at Institute of Automation, Chinese Academy of Sciences.

[1]Data and codes are available at https://github.com/nlpir2020/MIE-ACL-2020.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6460–6469

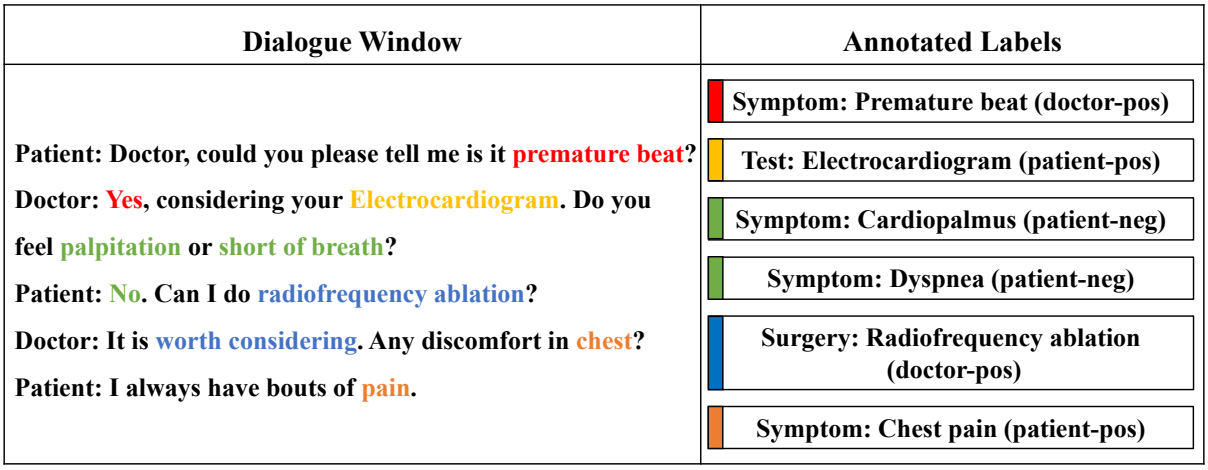

Figure 1: A typical medical dialogue window and the corresponding annotated labels. "Pos" is short for "positive" and "neg" is short for "negative". Text color and label color are aligned for clarity. All the examples in the paper are translated from Chinese.

sions are much more diverse than general texts. There are many medical terms in the dialogue, but many of them are not uttered formally, which will lead to performance degradation of conventional Natural Language Processing (NLP) tools. b) Available information is scattered in various dialogue turns, thus the interaction between turns should be also considered. In order to meet these challenges, we first annotate the dialogues in a window-sliding style, as illustrated in Figure 1. Then, we propose MIE, a **M**edical **I**nformation **E**xtractor constructed on a deep matching model. We believe our annotation method could put up with informal expressions, and the proposed neural matching model is able to harness the turn-interactions.

We collect doctor-patient dialogues from a popular Chinese online medical consultation website, Chunyu-Doctor [2], where medical dialogues are in text format. We focus on the cardiology domain, because there are more inquiries and less tests than other departments. The annotation method considers both effectiveness and feasibility. We define four main categories, including symptoms, tests, surgeries and other information, and we further define frequent items in the categories and their corresponding status at the same time. There are two merits of our annotation method: a) the annotation is much easier than the sequential labeling manner and does not need the labelers to be medical experts; b) we can annotate the circumstances that a single label is expressed by multiple turns. We totally annotate 1,120 dialogues with 18,212 segmented windows and obtain more than 40k labels.

We then develop MIE constructed on a novel neural matching model. MIE model consists of four main components, namely encoder module, matching module, aggregate module and scorer module. We conduct extensive experiments, and MIE achieves a overall F-score of 69.28, which indicates our proposed approach is a promising solution for the task.

To sum up, the contributions of this paper are as follows:

- We propose a new dataset, annotating 1,120 doctor-patient dialogues from online consultation medical dialogues with more than 40k labels. The dataset will help the following researchers.
- We propose MIE, a medical information extractor based on a novel deep matching model that can make use of the interaction between dialogue turns.
- MIE achieves a promising overall F-score of 69.28, significantly surpassing several competitive baselines.

2 Related Work

Extracting information from medical texts is a long-term objective for both biomedical and NLP community. For example, The 2010 i2b2 challenge provides a popular dataset still used in many recent researches (Uzuner et al., 2011). Three tasks were presented: a concept extraction task focused on the

[2]https://www.chunyuyisheng.com

extraction of medical concepts from patient reports; an assertion classification task focused on assigning assertion types for medical problem concepts; a relation classification task focused on assigning relation types that hold between medical problems, tests, and treatments.

Extracting medical information from dialogues just gets started. Finley et al. (2018) proposed a pipeline method to generate EMRs. The approach contains five steps: dialogue role labeling, Automatic Speech Recognition (ASR), knowledge extraction, structured data processing and Natural Language Generation (NLG) (Murty and Kabadi, 1987). The most important part is knowledge extraction, which uses dictionary, regular expression and other supervised machine learning methods. However, the detailed explanations are left out, which make us hard to compare with them.

Du et al. (2019) aimed at generating EMRs by extracting symptoms and their status. They defined 186 symptoms and three status, i.e., experienced, not experienced and other. They proposed two models to tackle the problem. Span-Attribute Tagging Model first predicted the span of a symptom, and then used the context features to further predict the symptom name and status. The seq2seq model took k dialogue turns as input, and then directly generated the symptom name and status. They collected incredible 90k dialogues and annotated 3k of them, but the dataset is not public.

The most similar work to ours is (Lin et al., 2019), which also annotated Chinese online medical dialogues. Concretely, they annotated 2,067 dialogues with the BIO (begin-in-out) schema. There are two main components, namely symptom recognition and symptom inference in their approach. The former utilized both document-level and corpus-level attention enhanced Conditional Random Field (CRF) to acquire symptoms. The letter serves determining the symptom status.

Our work differs from (Du et al., 2019) and (Lin et al., 2019) mainly in the following two points: a) we only extract 45 symptom items, but the status are more detailed, furthermore, we extract surgeries, tests and other information; b) we use different extracting method. Since the annotation system is different, our approach does not need the sequential labeling, which relieves the labeling work.

3 Corpus Description

3.1 Annotation Method

We collect doctor-patient dialogues from a Chinese medical consultation website, Chunyu-Doctor. The dialogues are already in text format. We select cardiology topic consultations, since there are more inquiries, while dialogues of other topics often depend more on tests. A typical consultation dialogue is illustrated in Figure 1. The principle of the annotation is to label useful information as comprehensive as possible.

A commonly utilized annotation paradigm is sequential labeling, where the medical entities are labeled using BIO tags (Du et al., 2019; Lin et al., 2019; Collobert et al., 2011; Huang et al., 2015; Ma and Hovy, 2016). However, such annotation methods cannot label information that a) expressed by multiple turns and b) not explicitly or not consecutively expressed. Such situations are not rare in spoken dialogues, as can be seen in Figure 1.

To this end, we use a window-to-information annotation method instead of sequential labeling. As listed in Table 1, we define four main categories, and for each category, we further define frequent items. The item quantity of `symptom`, `surgery`, `test` and `other info` is 45, 4, 16 and 6, respectively. In medical dialogues, status is quite

Category	Item	Status
Symptom	Backache Perspiration Hiccups Nausea Cyanosis Fever Fatigue Abdominal discomfort ...	patient-positive (appear) patient-negative (absent) doctor-positive (diagnosed) doctor-negative (exclude) unknown
Surgery	Interventional treatment Radiofrequency ablation Heart bypass surgery Stent implantation	patient-positive (done) patient-negative (not done) doctor-positive(suggest) doctor-negative (deprecated) unknown
Test	B-mode ultrasonography CT examination CT angiography CDFI Blood pressure measurement Ultrasonography MRI Thyroid function test Treadmill test ...	patient-positive(done) patient-negative (not done) doctor-positive(suggest) doctor-negative (deprecated) unknown
Other info	Sleep Diet Mental condition Defecation Smoking Drinking	patient-positive (normal) patient-negative (abnormal) unknown

Table 1: The detailed annotation labels of the dataset.

crucial that cannot be ignored. For example, for a symptom, the status of appearance or absence is opposite for a particular diagnose. So it is necessary to carefully define status for each category. The status options vary with different categories, but we use unified labels for clarity. The exact meanings of the labels are also explained in Table 1.

The goal of annotation is to label all the predefined information mentioned in the current dialogue. As the dialogues turn to be too long, it is difficult for giving accurate labels when finishing reading them. Thus, we divide the dialogues into pieces using a sliding window. A window consists of multiple consecutive turns of the dialogue.

It is worth noting that the window-sliding annotations can be converted into dialogue-based ones like dialogue state tracking task (Mrkšić et al., 2017), the later annotation state will overwrite the old one. Here, the sliding window size is set to 5 as Du et al. (2019) did, because this size allows the included dialogue turns contain proper amount of information. For windows with less than 5 utterances, we pad them at the beginning with empty strings. The sliding step is set to 1.

We invite three graduate students to label the dialogue windows. The annotators are guided by two physicians to ensure correctness. The segmented windows are randomly assigned to the annotators.

In all, we annotate 1,120 dialogues, leading to 18,212 windows. We divide the data into train/develop/test sets of size 800/160/160 for dialogues and 12,931/2,587/2,694 for windows, respectively. In total, 46,151 labels are annotated, averaging 2.53 labels in each window, 41.21 labels in each dialogue. Note that about 12.83% of windows have no gold labels, i.e., there is no pre-defined information in those windows. The distribution of the labels is shown in Table 2. The status distribution is shown in Table 3. The annotation consistency, i.e., the cohen's kappa coefficient (Fleiss and Cohen, 1973) of the labeled data is 0.91, which means our annotation approach is feasible and easy to follow.

	Dialogue	Window	Symptom	Surgery	Test	Other info
Train	800	12931	21420	839	8879	1363
Dev	160	2587	4254	119	1680	259
Test	160	2694	4878	264	1869	327
Total	1120	18212	30552	1222	12428	1949

Table 2: The detailed annotation statistics of the dataset.

	Patient-pos	Patient-neg	Doctor-pos	Doctor-neg	Unknown
Symptom	15119	1782	1655	910	11086
Surgery	169	48	698	10	297
Test	5589	303	4443	44	2049
Other info	550	1399	-	-	1505

Table 3: The distribution of status over all labels.

3.2 Evaluation Metrics

We evaluate the extracted medical information results as ordinary information extraction task does, i.e., Precision, Recall and F-measure. To further discover the model behavior, we set up three evaluation metrics from easy to hard. **Category** performance is the most tolerant metric. It merely considers the correctness of the category. **Item** performance examines the correctness of both category and item, regardless of status. **Full** performance is the most strict metric, meaning that category, item and the corresponding status must be completely correct.

We will report both window-level and dialogue-level results.

Window-level: We evaluate the results of each segmented window, and report the micro-average of all the test windows. Some windows have no gold labels, if the prediction on a window with no gold labels is also empty, it means the model performs well, so we set the Precision, Recall and F-measure to 1, otherwise 0.

Dialogue-level: First we merge the results of the windows that belong to the same dialogue. For labels that are mutually exclusive, we update the old labels with the latest ones. Then we evaluate the results of each dialogue, and finally report the micro-average of all the test dialogues.

4 Our Approach

In this section, we will elaborate the proposed MIE model, a novel deep matching neural network model. Deep matching models are widely used in multiple natural language processing tasks such as machine reading comprehension (Seo et al., 2017; Yu et al.), question answering (Yang et al., 2016) and dialogue generation (Zhou et al., 2018; Wu et al., 2017). Compared with classification models, matching models are able to introduce more information of the candidate side and promote interaction between both ends.

The architecture of MIE is shown in Figure 2. There are four main components, namely encoder module, matching module, aggregate module and scorer module. The input of MIE is a doctor-patient

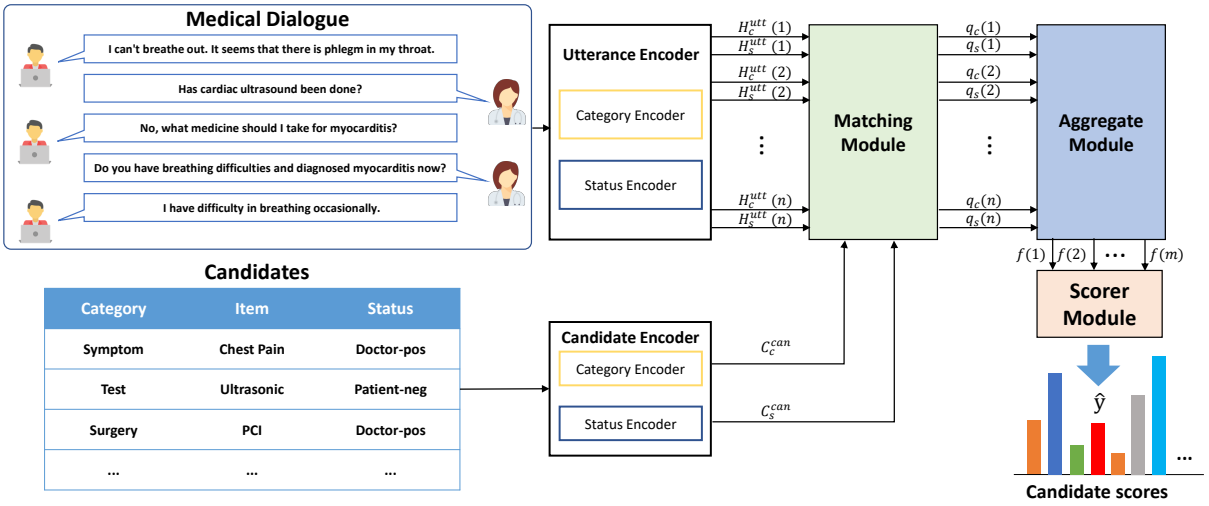

Figure 2: The architecture of MIE model.

dialogue window, and the output is the predicted medical information.

Encoder Module

The encoder is implemented by Bi-LSTM (Hochreiter and Schmidhuber, 1997) with self-attention (Vaswani et al., 2017). Let the input utterance be $X = (x_1, x_2, ..., x_l)$, the encoder works as follows:

$$\begin{aligned} H &= \text{BiLSTM}(X) \\ a[j] &= WH[j] + b \\ p &= \text{softmax}(a) \\ c &= \sum_j p[j]H[j] \end{aligned} \tag{1}$$

We denote $H, c = \text{Encoder}(X)$ for brevity. H consists contextual representations of every token in input sequence X, and c is a single vector that compresses the information of the entire sequence in a weighted way.

We denote a window with n utterances as $\{U[1], ...U[n]\}$. For a candidate consists of category, item and status like `Symptom:Heart failure (patient-positive)`, we split it to category-item pair `Symptom:Heart failure` denoted by V and status `patient-positive` denoted by S. To introduce more oral information, we also add item-related colloquial expressions collected during the annotation to the end of V. Having defined the basic structure of the encoder, we now build representations for utterances U in the dialogue window, and the candidate category-item pair V and its status S:

$$\begin{aligned} H_c^{utt}[i], c_c^{utt}[i] &= \text{Encoder}_c^{utt}(U[i]) \\ H_s^{utt}[i], c_s^{utt}[i] &= \text{Encoder}_s^{utt}(U[i]) \\ H_c^{can}, c_c^{can} &= \text{Encoder}_c^{can}(V) \\ H_s^{can}, c_s^{can} &= \text{Encoder}_s^{can}(S) \end{aligned} \tag{2}$$

Where the superscript utt and can represents utterance encoder and candidate encoder respectively, the subscript c and s represents category encoder and status encoder respectively, and $i \in [1, n]$ is the index of utterance in the dialogue window. All the candidates will be encoded in this step, but we only illustrate one in the figure and equations for brevity. Note that U, V, S is encoded with encoders differ from utterance to candidate and from category to status in order to make each encoder concentrate on one specific type (category-specific and status-specific) of information.

Matching Module

In this step, the category-item representation is treated as a query in attention mechanism to calculate the attention values towards original utterances. Then we can obtain the category-specific representation of utterance $U[i]$ as $q_c[i]$.

$$\begin{aligned} a_c[i, j] &= c_c^{can} \cdot H_c^{utt}[i, j] \\ p_c[i] &= \text{softmax}(a_c[i]) \\ q_c[i] &= \sum_j p_c[i, j] H_c^{utt}[i, j] \end{aligned} \tag{3}$$

Meanwhile, the status representation is treated as another query to calculate the attention values

towards original utterances. Then we can obtain the status-specific representation of utterance $U[i]$ as $q_s[i]$.

$$\begin{aligned} a_s[i,j] &= c_s^{can} \cdot H_s^{utt}[i,j] \\ p_s[i] &= \text{softmax}(a_s[i]) \\ q_s[i] &= \sum_j p_s[i,j] H_s^{utt}[i,j] \end{aligned} \tag{4}$$

Where $[i, j]$ denotes the jth word in the ith utterance. The goal of this step is to capture the most relevant information from each utterance given a candidate. For example, if the category-item pair of the candidate is `Symptom: Heart failure`, the model will assign high attention values to the mentions of heart failure in utterances. If the status of the candidate is `patient-positive`, the attention values of expressions like "I have", "I've been diagnosed" will be high. So the matching module is important to determine the existence of a category-item pair and status related expressions.

Aggregate Module

The matching module introduced above have captured the information of the existence of category-item pairs and status. To know whether a candidate is expressed in a dialogue window, we need to obtain the category-item pair information and its status information together. In particular, we need to match every category-item representation $q_c[i]$ with $q_s[i]$.

Sometimes the category-item pair information and its status information appear in the same utterance. But sometimes, they will appear in different utterances. For example, many question-answer pairs are adjacent utterances. So we need take the interactions between utterances into account. Based on this intuition, we define two kinds of strategies to get two different models.

MIE-single: The first strategy assumes that the category-item pair information and its status information appear in the same utterance. The representation of the candidate in the ith utterance is a simple concatenation of $q_c[i]$ and $q_s[i]$:

$$f[i] = \text{concat}(q_c[i], q_s[i]) \tag{5}$$

Where $f[i]$ consists information of category-item pair and its status which can be used to predict the score of the related candidate. The model only considers the interaction within a single utterance. The acquired representations are independent from each other. This model is called MIE-single.

MIE-multi: The second strategy considers the interaction between the utterances. To obtain the related status information of other utterances, we treat $q_c[i]$ as a query to get the attention values towards the representations of status, i.e., q_s. Then we can obtain the candidate representation of the utterance:

$$\begin{aligned} a[i,k] &= q_c[i]^T W q_s[k] \\ p[i] &= \text{softmax}(a[i]) \\ \widetilde{q_s}[i] &= \sum_k p[i,k] q_s[k] \\ f[i] &= \text{concat}(q_c[i], \widetilde{q_s}[i]) \end{aligned} \tag{6}$$

Where W is a learned parameter, and $\widetilde{q_s}$ is the new representation of the status, containing the relative information of other utterances. The utterance order is an important clue in a dialogue window. For example, the category-item pair information can hardly related to status information whose utterance is too far. In order to capture this kind of information, we also take utterance position into account. Concretely, we add positional encoding (Vaswani et al., 2017) to each q_c and q_s at the beginning. We denote this model as MIE-multi.

The output of the aggregate module contains the information of a entire candidate, including category-item and status information.

Scorer Module

The output of the aggregate module is fed into a scorer module. We use each utterance's feature $f[i]$ to score the candidate, as it is already the candidate-specific representation. The highest score of all the utterances in the window is the candidate's final score:

$$\begin{aligned} s^{utt}[i] &= \text{feedforward}(f[i]) \\ y &= \text{sigmoid}(\max(s^{utt}[i])) \end{aligned} \tag{7}$$

Where feedforward is a 4 layer full-connection neural network.

Learning

The loss function is the cross entropy loss defined as follows:

$$\begin{aligned} L = \frac{1}{KL} \sum_k \sum_l & -y_l^k \log(\widehat{y}_l^k) + \\ & (1 - y_l^k) \log(1 - \widehat{y}_l^k) \end{aligned} \tag{8}$$

The superscript k denote the index of the training sample, and l is the index of the candidate. K and L are the number of samples and candidates respectively. $\widehat{y}_l^k$ is the true label of the training sample.

Inference

There could be more than one answer in a dialogue window. In the inference phase, we reserve all the candidates whose matching score is higher than the threshold of 0.5. Since the training process is performed in the window size, the inference phase should be the same situation. We also obtain the dialogue-level results by updating the results of windows as aforementioned.

5 Experiments

In this section, we will conduct experiments on the proposed dataset. It is worth to note that we are not going to compare MIE with (Du et al., 2019) and (Lin et al., 2019), because a) they all employed sequential labeling methods, leading to different evaluation dimensions from ours (theirs are more strict as they must give the exact symptom positions in the original utterance), and b) their approaches were customized for sequential labeling paradigm, thus cannot be re-implemented in our dataset.

5.1 Implementation

We use pretrained 300-dimensional Skip-Gram (Mikolov et al., 2013) embeddings to represent chinese characters. We use Adam (Kingma and Ba, 2015) optimizer. The size of the hidden states of both feed-forward network and Bi-LSTM is 400. We apply dropout (Srivastava et al., 2014) with 0.2 drop rate to the output of each module and the hidden states of feed-forward network for regularization. We adopt early stopping using the F1 score on the development set.

5.2 Baselines

We compare MIE with several baselines.

1) Plain-Classifier. We develop a basic classifier model that uses the simplest strategy to accomplish the task. The input of the model are the utterances in the window. We concatenate all the utterances to obtain a long sequence, and encode it using a Bi-LSTM encoder, then we use self-attention to represent it as a single vector. Next, the vector is fed into a feed-forward classifier network. The output labels of the classifier consist of all the possible candidates. The encoder adopts category-specific parameters.

2) MIE-Classifier. To develop a more competitive model, we reuse MIE model architecture to implement an advanced classifier model. The difference between the classifier model and MIE is the way of obtaining q_c and q_s. Instead of matching, the classifier model treats c_c^{utt} and c_s^{utt} directly as q_c and q_s respectively. Thanks to the attention mechanism in the encoder, the classifier model can also capture the category-item pair information and the status information to some extent. To further examine the effect of turn-interaction, we develop two classifiers as we do in MIE. MIE-Classifier-single treats each utterance independently, and the probability score of each utterance is calculated. The model uses a max-pooling operation to get the final score. MIE-Classifier-multi considers the turn-interaction as MIE-multi does.

5.3 Main Results

The experimental results are shown in Table 4. From the results, we can obtain the following observations.

1) MIE-multi achieves the best F-score on both window-level and dialogue-level full evaluation metric, as we expected. The F-score reaches 66.40 and 69.28, which are considerable results in such sophisticated medical dialogues.

2) Both of the models using multi-turn interactions perform better than models solely using single utterance information, which further indicates the relations between turns play an important role in dialogues. The proposed approach can capture the interaction. As a proof, MIE-multi achieves a 2.01% F-score improvement in dialogue-level full evaluation.

3) Matching-based methods surpass classifier models in full evaluation. We think the results are rational because matching-based methods can introduce candidate representation. This also motivates us to leverage more background knowledge in the future. Note that in category and item metrics, MIE-classifiers are better at times, but they fail to correctly predict the status information.

4) Both MIE models and MIE-classifier models overwhelm Plain-Classifier model, which indicates the MIE architecture is far more effective than the basic LSTM representation concatenating method.

5) Dialogue-level performance is not always better than window-level performance in full evalua-

Model	Window-level									Dialogue-level								
	Category			Item			Full			Category			Item			Full		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Plain-Classifier	67.21	63.78	64.92	60.89	49.20	53.81	53.13	49.46	50.69	93.57	89.49	90.96	83.42	73.76	77.29	61.34	52.65	56.08
MIE-Classifier-single	80.51	76.39	77.53	76.58	64.63	68.30	68.20	61.60	62.87	97.14	91.82	93.23	91.77	75.36	80.96	71.87	56.67	61.78
MIE-Classifier-multi	**80.72**	**77.76**	**78.33**	76.84	**68.07**	**70.35**	67.87	64.71	64.57	96.61	**92.86**	**93.45**	90.68	82.41	84.65	68.86	62.50	63.99
MIE-single	78.62	73.55	74.92	76.67	65.51	68.88	69.40	64.47	65.18	96.93	90.16	92.01	94.27	79.81	84.72	75.37	63.17	67.27
MIE-multi	80.42	76.23	77.77	**77.21**	66.04	69.75	**70.24**	**64.96**	**66.40**	**98.86**	91.52	92.69	**95.31**	**82.53**	**86.83**	**76.83**	**64.07**	**69.28**

Table 4: The experimental results of MIE and other baseline models. Both window-level and dialogue-level metrics are evaluated.

tion. In our experiment, the classifier-based models perform better in window-level than dialogue-level in full evaluation. The possible reason is error accumulation. When the model predicts results the current window does not support, the errors will be accumulated with the processing of the next window, which will decrease the performance.

5.4 Error Analysis

To further analyze the behavior of MIE-multi, we print the confusion matrix of category-item predictions, as shown in Figure 3. We denote the matrix as A, $A[i][j]$ means the frequency of the circumstance that the true label is i while MIE-multi gives the answer j.

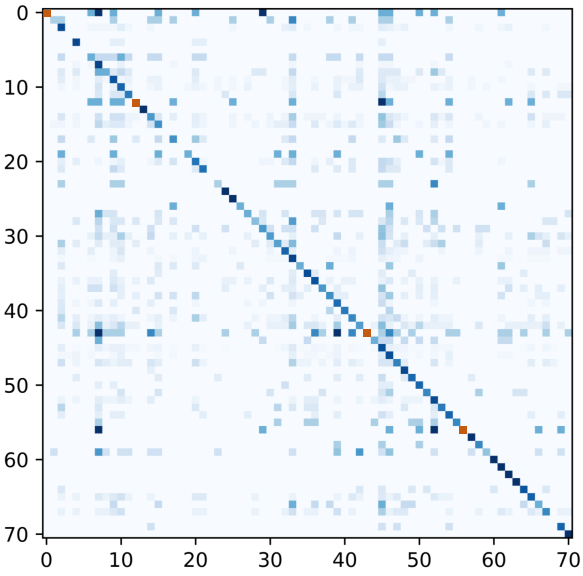

Figure 3: Illustration of the confusion matrix of MIE-multi. Darker color means higher value. The figure in the axis is the category-item pair index of a total number of 71. Values of orange blocks are 0.

We study the matrix and find that MIE-multi failed to predict `Symptom:Limited mobility`, `Symptom:Nausea`, `Symptom:Cardiomyopathy`, and `Test: Renal function test`, which are emphasized by orange blocks ($A[i][i] = 0$) in Figure 3. The possible reason is that they rarely appear in the training set, with frequency of 0.63%, 2.63%, 2.38% and 1.25%, respectively. The results reveal that the data sparse and uneven problems are the bottlenecks of our approach.

Patient: I have atrial fibrillation, heart failure, anemia and loss my appetite.
Doctor: Hello! How long did them last? Did you examine blood routine?
Patient: Yes.
Doctor: Is there coronary heart disease?
Patient: No.
(a)

Patient: I have atrial fibrillation, heart failure, anemia and loss my appetite.
Doctor: Hello! How long did them last? Did you examine blood routine?
Patient: Yes.
Doctor: Is there coronary heart disease?
Patient: No.
(b)

Patient: I have atrial fibrillation, heart failure, anemia and loss my appetite.
Doctor: Hello! How long did them last? Did you examine blood routine?
Patient: Yes.
Doctor: Is there coronary heart disease?
Patient: No.
(c)

Figure 4: Case illustration of attentions: a) attention heat map of category-item pair for each utterance; b) attention heat map of status for each utterance; c) attention heat map for the fourth utterance in the window.

5.5 Case Discussion

Attention Visualization

In this part, we will analyze some cases to verify the effectiveness of the model with best performance, e.g. MIE-multi. Particularly, we investigate an example shown in Figure 4. To determine whether the candidate `Symptom:Coronary heart disease (patient-negative)` is mentioned in the window, we should focus on the interaction between the adjacent pair located in the last of the window. This adjacent pair is a question-answer pair, the category-item pair information is in the question of the doctor while the status information is in the answer of the patient. In this case, MIE-

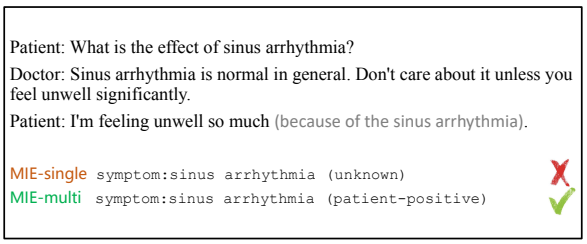

Figure 5: Predictions of MIE-single and MIE-multi. The gray string is the implicit reason.

single does not predict right result due to its independence between utterances, while MIE-multi manages to produce the correct result.

For better understanding, we utilize visualization for matching module and aggregate module. Figure 4(a) is the attention heat map when the category-item pair information vector c_c^{can} matches the utterances category representations H_c^{utt}. We can observe that the attention values of the mention of coronary heart disease are relatively high, which illustrates that the model can capture the correct category-item pair information in the window.

Figure 4(b) is the attention heat map when the status information c_s^{can} matches the utterances status representation H_s^{utt}. The attention values of the expressions related to status such as "Yes" and "No" are high, and the expression "No" is even higher. So MIE-multi can also capture the status information in the window.

We also visualize the interaction between the fourth utterance and the other utterances. In Figure 4(c), the score of the fifth utterance is the highest, which is in line with the fact that the fifth utterance is the most relevant utterance in the window. In this way the model successfully obtains the related status information for the category-item pair information in the window.

In a nutshell, MIE-multi can properly capture the category-item pair and status information.

The Effectiveness of Turn Interaction

We demostrate a case in Figure 5 that can explicitly show the need for turn interaction, where MIE-multi shows its advancement. In this case, the label `Symptom:Sinus arrhythmia (patient-positive)` requires turn interaction information. Specifically, in the third utterance, the patient omits the reason that makes him sick. However, under the complete context, we can infer the reason is the sinus arrhythmia, since the patient consulted the doctor at the beginning of the window. The model need to consider the interaction between different utterances to get the conclusion. Interaction-agnostic model like MIE-single makes prediction on single utterance, and then sums them up to get the final conclusion. Consequently, it fails to handle the case when the expressions of category-item and status are separated in different utterances. As a result, MIE-single only obtains the category-item information `Symptom:Sinus arrhythmia`, but the status prediction is incorrect. In contrast, MIE-multi is able to capture the interaction between different utterances and predicts the label successfully.

6 Conclusion and Future Work

In this paper, we first describe a new constructed corpus for the medical information extraction task, including the annotation methods and the evaluation metrics. Then we propose MIE, a deep neural matching model tailored for the task. MIE is able to capture the interaction information between the dialogue turns. To show the advantage of MIE, we develop several competitive baselines for comparison. The experimental results indicate that MIE is a promising solution for medical information extraction towards medical dialogues.

In the future, we should further leverage the internal relations in the candidate end, and try to introduce rich medical background knowledge into our work.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No.61533018, No.61922085, No.61906196) and the Key Research Program of the Chinese Academy of Sciences (Grant NO. ZDBS-SSW-JSC006). This work is also supported by Beijing Academy of Artificial Intelligence (BAAI2019QN0301), the Open Project of Beijing Key Laboratory of Mental Disroders (2019JSJB06) and the independent research project of National Laboratory of Pattern Recognition.

References

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (almost) from Scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

Nan Du, Kai Chen, Anjuli Kannan, Linh Tran, Yuhui Chen, and Izhak Shafran. 2019. Extracting symptoms and their status from clinical conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 915–925.

Gregory Finley, Erik Edwards, Amanda Robinson, Michael Brenndoerfer, Najmeh Sadoughi, James Fone, Nico Axtmann, Mark Miller, and David Suendermann-Oeft. 2018. An automated medical scribe for documenting clinical encounters. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–15.

Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619.

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

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.

Xinzhu Lin, Xiahui He, Qin Chen, Huaixiao Tou, Zhongyu Wei, and Ting Chen. 2019. Enhancing dialogue symptom diagnosis with global attention and symptom graph. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5032–5041.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.

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.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788.

Katta G Murty and Santosh N Kabadi. 1987. Some np-complete problems in quadratic and nonlinear programming. *Mathematical programming*, 39(2):117–129.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations, ICLR 2017*.

Christine Sinsky, Lacey Colligan, Ling Li, Mirela Prgomet, Sam Reynolds, Lindsey Goeders, Johanna Westbrook, Michael Tutty, and George Blike. 2016. Allocation of physician time in ambulatory practice: a time and motion study in 4 specialties. *Annals of internal medicine*, 165(11):753–760.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Robert Wachter and Jeff Goldsmith. 2018. To combat physician burnout and improve care, fix the electronic health record. *Harvard Business Review*.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 496–505.

Liu Yang, Qingyao Ai, Jiafeng Guo, and W Bruce Croft. 2016. anmm: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 287–296.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. In *6th International Conference on Learning Representations, ICLR 2018*.

Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1118–1127.