Knowledge Abstraction Matching for Medical Question Answering

Jun Chen[#], Jingbo Zhou[†*], Zhenhui Shi[#], Bin Fan[#], Chengliang Luo[#]
[#]Baidu Inc, Beijing, China [†]Business Intelligence Lab, Baidu Research
[†]National Engineering Laboratory of Deep Learning Technology and Application, China
{chenjun22, zhoujingbo, shizhenhui, fanbin, luochengliang}@baidu.com

***Abstract*—Medical Question Answering (medical QA), which studies the problem of automatically answering patients' medical questions online, is one of the major applications of bioinformatics. Though many efforts have been made before, the medical QA system still deserves delicate algorithm optimization due to the serious application scenario and strict requirement for the answer quality. In this paper, we introduce a novel Knowledge Abstraction Matching (KAM) method for the medical QA problem. The intuition of KAM is that there are many frequent repeat text segments appearing in the answers across different questions. From this view, we propose a new method that consists of frequent segment N-gram mining, medical knowledge abstraction, medical segment matching and answer re-retrieval. KAM has been incorporated into Baidu's enterprise medical QA system MelodyQA deployed on the backend of Muzhi Doctor. The evaluation shows that the proposed method can generate more quality answers for MelodyQA with a significant improvement of question coverage under acceptable accuracy.**

***Index Terms*—Medical Question Answering, Knowledge Abstraction Matching, Information Retrieval-based QA**

I. Introduction

Question answering (QA) studies the problem of automatically finding or generating answers for users' questions. Medical QA, which investigates QA in the medical domain, is one of the main applications of healthcare informatics. It attracts special research attention due to the challenges like the high requirement of answer quality, the complex medical entities and the domain specific knowledge. In the past decade, there have already been many efforts devoted to the study of the medical QA problem from different perspectives [1]–[6].

This research is based on Baidu's enterprise medical QA system (denoted by **MelodyQA**) deployed on Baidu's online medical consultation platform, *Muzhi Doctor* [1], to enable patients to consult with doctors through internet for professional medical advice, post-diagnosis services, medication alerts and more. MelodyQA is designed in a Business-to-Doctor-to-Customer style where the QA model does not directly present the answer to the patient. Instead, after receiving a question from a patient, MelodyQA returns up to three candidate answers to a certificated doctor or physician who can further choose to *approve directly*, *approve with minor revision*, or *reject and manually compose the answer from scratch* before presenting to the patient. The objective of MelodyQA is to improve the doctors' work efficiency. It enables doctors to answer medical questions by simply clicking or revising only several words from the candidates generated by computers. MelodyQA has been providing stable service for a long time with iterative and incremental development. It has also been equipped with the advanced techniques for QA system like information retrieval-based QA (IR-QA), entity extraction, intent matching, learning to rank, deep learning based QA matching [7] as well as manually defined rules. After several rounds of the development and optimization, the improvement of the MelodyQA's performance has become very difficult.

In this paper, we propose the novel **K**nowledge **A**bstraction **M**atching (**KAM**) method for the medical QA problem. KAM is based on IR-QA methods, i.e. for a new question, we retrieve its candidate answers from the database of historical question-answer pairs. The novelty of the proposed method is that, instead of using the patient's question to match with the historical questions or answers (or their combination), we first use the patient's question to match with a set of knowledge abstractions derived from the frequent segment N-gram mining on the historical answers, and then we jointly use the patient's question and the matched knowledge abstraction to retrieve the final qualified answer.

KAM is inspired by the observation that, there are usually many repeated text segments across different historical answers in the medical QA corpus. In many cases, although the patients describe the same symptoms or disease in different ways, the answers are usually quite similar with sharing segments. The questions that share answer segments form a cluster which represents a specific field of *medical knowledge*, e.g. the treatment for influenza, the symptoms of gastritis and the medicine for hypertension. Figure 1 shows some examples in our medical corpus where the answers given by different doctors share some segments, for example, *no spicy food*, *Clarithromycin*, *gastroscopy* and *barium meal check*. The three questions are textually different from each other but all about the stomach issues.

The key idea of KAM is to generate knowledge abstraction, i.e. some normalized keywords indeed, by extracting feature representation from the question cluster, and then use the medical knowledge abstraction to augment the patient's question to retrieve better candidate answers from the historical QA database. Knowledge abstraction provides a new way to

*Jingbo Zhou is the corresponding author.
[1]https://muzhi.baidu.com

978-1-7281-1867-3/19/$31.00 © 2019 IEEE

Question	Answer	Shared Segment N-Grams	Medical Knowledge Abstraction
My stomach is uncomfortable. Sometimes it gets bloating and sometimes it feels like eating chilli in my stomach. What should I do?	Stomach bloating, pain, heartburn and acid reflux indicate there is gastritis. [1]Firstly, it should be light diet, No spicy food. Small meals and more times. Take medication regularly like Lansoprazole, [2] Clarithromycin. If not getting better, [3]recommend gastroscopy or barium meal check. Treat according to results.	1. Firstly, it should be light diet, No spicy food. Small meals and more times. 2. Clarithromycin. 3. Recommend gastroscopy or barium meal check. Treat according to results.	stomach, stomach bloating, stomach pain, stomach sour, acid water
My stomach bloating and pain get much worse when feeling hungry. What's wrong?	Stomach bloating, heartburn and pain when hungry are the typical symptoms of gastritis. [1]Firstly, it should be light diet, No spicy food. Small meals and more times. Eat more fresh vegetables and fruits. It is recommended to take medication like Lansoprazole, [2] Clarithromycin. If not getting any better, [3]recommend gastroscopy or barium meal check. Treat according to results.		
Feeling sour in stomach. Often spit out with acid water. What's wrong with me?	Hi, acid reflux, heartburn and the loss of appetite are all indications of gastritis. [1]Firstly, it should be light diet, No spicy food. Small meals and more times. Take medication regularly like Omeprazole, [2] Clarithromycin. If there is no relief of symptoms, [3]recommend gastroscopy or barium meal check. Treat according to results.		

Fig. 1. Examples of some shared segments across different answers and medical knowledge abstraction. The shared segments are highlighted in red color.

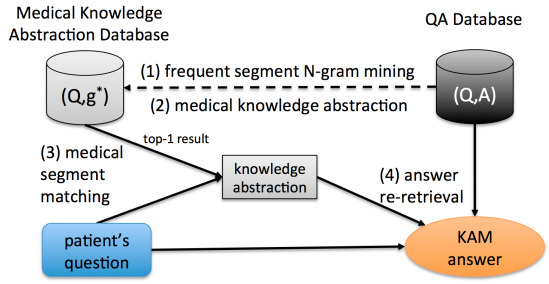

Fig. 2. Overview of the Knowledge Abstraction Matching method. Step (1) and (2) are pre-processed offline. Step (3) and (4) are processed online.

connect patient's question with the historical questions and answers in the database. Thus, it can be considered as a new method of question augmentation in solving medical QA.

Figure 2 shows the overview of KAM, which consists of four main steps: (1) frequent segment N-gram mining, (2) medical knowledge abstraction, (3) medical segment matching and (4) answer re-retrieval. Step (1) is an offline mining process to generate the frequent segment N-grams in the historical answers. These N-grams are important patterns, each of which can be used to answer a group of questions. Step (2) is offline pre-processing where we abstract the representation of the medical knowledge for each group of questions obtained from Step (1). When a patient posts a new medical question online, it triggers Step (3) to match the question with the medical knowledge abstraction. Finally, we use the patient's question and the matched knowledge abstraction to re-retrieve the quality answer from the historical QA database in Step (4).

In the collaboration with MelodyQA, if MelodyQA dose not return any candidate answer for a medical question due to low similarity or low confidence, KAM will be used to retrieve answers. Thus, KAM deals with more difficult questions than those handled by MelodyQA because the easy ones have already been covered by MelodyQA.

We summarise the major contribution as follows:

- We propose KAM, a novel solution to the medical QA problem. KAM aims to handle the questions which cannot be covered by the enterprise non-factoid medical question answering system, MelodyQA.
- We introduce the new discovery that in medical QA corpus, there are many frequent text segments appearing in the answers across different questions. This observation inspires us to cluster the questions which share text segments in the answers, which is the basis of KAM.
- We present a novel framework to extract knowledge abstraction from the question clusters based on the frequent segment N-gram mining. Then, we propose a medical segment matching method to match the patient's question with the knowledge abstraction. The matched knowledge abstraction and the patient's question are jointly used to re-retrieve the quality answer in the QA database.
- The real-world experiments show that KAM can significantly improve the coverage of MelodyQA while keeping the same answer quality.

II. Related Work

In this section, we present the brief review of question answering and medical question answering.

A. Question Answering

Question Answering (QA) systems can be generally classified into two categories: knowledge base-based QA (KB-QA) and information retrieval-based QA (IR-QA). The KB-QA systems generate answers after searching the knowledge base. One of the main challenges of KB-QA is how to translate the questions into structured queries like SPARQL and SQL [8]. The IR-QA systems retrieve documents that are the most relevant to the question [9]. The documents can be historical question-answer pairs that answer the question directly or the relevant documents from which we can extract answers.

KAM belongs to the IR-QA category since its objective is to retrieve the candidate answers from historical QA data. Different machine learning models have also been adopted for IR-QA, such as the Tree Edit Distance (TED) model [10], Support Vector Machines (SVMs) [11] and Conditional Random Fields (CRFs) [12]. Although these methods show effectiveness upon the effort of feature engineering, in recent years, these feature engineering based approaches have been outperformed by deep learning based approaches [7], [13], [14].

The deep learning approache learns the low-dimensional representations of question and answer which can be used as the input features into the other layers [15]–[17]. The network architecture for question answering matching can be divided into three categories: siamense network [18]–[20], attentive network [13], [21], [22], and compare-aggregate network [23].

B. Medical Question Answering

A closely related domain of the medical QA in this study is clinical QA, which is usually a part of the Clinical Decision Support (CDS) system to rank the scientific articles after obtaining the comprehensive information of patient (e.g. the electronic medical records, a summary of the medical case, and generic questions of diagnosis and the tests) [2]–[4], [6]. In contrast, KAM works on the historical question-answer pairs generated by the patients and doctors which are quite different from the scientific articles and patients' information documents. Therefore, the techniques of the clinical QA cannot be applied directly to our problem.

There are also some previous work about question answering in medical domain. One of the pioneering systems of the medical QA system is presented in [1], which tries to automatically define the generic logic form of a medical question towards a set of matched questions, and then retrieve the relevant answers from medical website documents. Moreover, transfer learning and biomedical word embedding are used to improve the performance on medical QA [5]. How to translate medical questions into SPARQL query for KB-QA with medical entity extraction and semantic recognition is investigated in [6].

To sum up, to the best of our knowledge, there are no previous study using the knowledge abstraction matching method to improve the performance of medical QA system.

III. The MelodyQA System

Before presenting the proposed method, we briefly introduce the MelodyQA system first. MelodyQA provides professional online medical QA service powered by Baidu. It does not directly present the answers to the patients. Instead, it receives a question from an online patient and returns up to three candidate answers to a certificated doctor who can further choose to approve directly, approve with minor revision or reject and manually compose the answer from scratch before presenting to the patient. For each question, MelodyQA may return up to 3 answers based on the confidence threshold. The workflow of MelodyQA contains the following modules:

1) Preprocessing the QA texts with parsing, entity extraction and resolution and automatic typo correction.
2) IR based answer recall using ElasticSearch [2].
3) QA intent matching by filtering the retrieved questions with different intents of the query.
4) Initial ranking based on hand-crafted text features, e.g. word or char level TF-IDF.
5) Re-ranking based on deep learning models [7], [13].
6) Rule based answer quality control by removing unqualified answers.

If user's question cannot be answered by MelodyQA, a further attempt will be made by KAM which is possible to find appropriate answers. From this illustration, we can see that KAM mostly aims at increasing the coverage of users' medical questions that can be *answered* by the QA system, and the later experiments also validate this point.

MelodyQA has been well-developed and equipped with the state-of-the-art machine learning and natural language processing techniques in the QA research. It has put the most focus on matching the patient's question Q_u with the historical questions. In that case, not enough effort has been made to investigate the relation between Q_u and the historical answers, which motivates us to propose the KAM method.

IV. The KAM Method

The KAM method consists of four steps: 1) frequent segment N-gram mining, 2) medical knowledge abstraction, 3) medical segment matching and 4) answer re-retrieval.

A. Frequent Segment N-gram Mining

Figure 3 illustrates the process of frequent segment N-gram mining. The raw QA database contains all *(question, answer)* pairs in the original text form. The text of each answer A_i (subscript denotes index) is firstly segmented by using pause punctuation as separation like *comma*, *semicolon*, *period* and *question mark*. Thus, each answer corresponds to a list of text segments $\mathcal{S}_i = \{s_1, s_2, ..., s_n\}$. Please note that each segment is a string of text instead of a single letter, digit or symbol. Then, the **segment N-grams** of each answer are generated by outputting the N consecutive segments over the segmented texts using a sliding window. To avoid that each segment N-gram is too short to be useful, N is required to be no less than a threshold, e.g. $3 \leq N \leq |\mathcal{S}_i|$. We generate all valid segment N-grams for each answer by increasing N. In Figure 3, answer A_1 is first divided into 4 segments $\{s_1, s_2, s_3, s_4\}$ which further generate two 3-grams ($s_1s_2s_3$ and $s_2s_3s_4$) and one 4-gram ($s_1s_2s_3s_4$). Segment-level N-gram which is one of the major contributions of this work, carries more information than word-level or letter-level N-gram in expressing ideas. Without specific statement, N-gram in this paper means segment N-gram.

Next, the same N-grams are merged and counted. A two-step filtering with the following two criteria is performed to select out the **Frequent Segment N-grams**: (1) The N-grams whose frequency counts are less than a threshold η (e.g. $\eta = 3$) are removed. The rest N-grams are therefore considered frequent. (2) If an N-gram is fully covered by another N-gram, the shorter one is removed. This ensures that no single N-gram can be fully expressed by another. In Figure 3, $s_3s_4s_5$ and $s_1s_2s_3s_4$ are filtered by (1) since their frequency counts are less than 3, while $s_2s_3s_4$ is filtered by (2) because it is fully covered by $s_2s_3s_4s_5$ which is preserved.

After filtering, there may still be large overlaps between the rest N-grams. For example, $s_2s_3s_4s_5$ is heavily overlapped with $s_3s_4s_5s_6$. Overlapping N-grams bring much redundancy. Therefore, the hierarchical agglomerative clustering (the *bottom-up* approach) is performed to group the similar N-grams based on the TF-IDF feature representations. Let $\mathcal{C}_i = \{g_{i1}, ..., g_{ij}, ...\}$ denote the i-th cluster while frequent N-gram g_{ij} be the j-th member of $\mathcal{C}_i$. The longest member

[2]https://www.elastic.co/

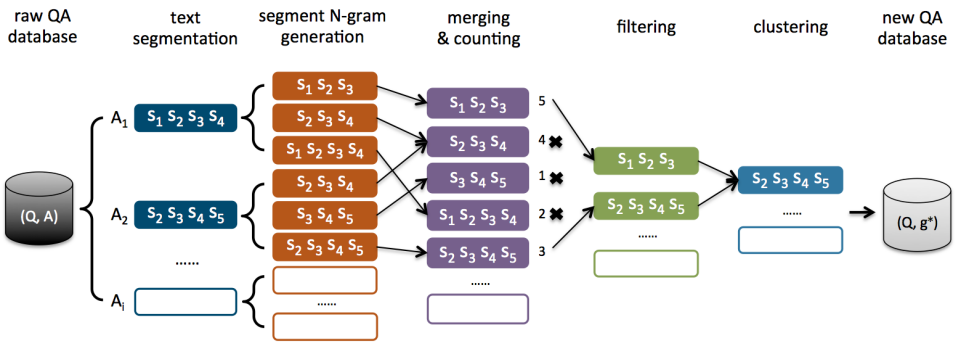

Fig. 3. Frequent segment N-gram mining. $\eta = 3$.

$g_i^* = \arg\max_{g \in \mathcal{C}_i} \text{length}(g)$ is selected as the center of cluster $\mathcal{C}_i$ because g_i^* usually covers most of the N-grams in this cluster. For example, $s_1 s_2 s_3$ and $s_2 s_3 s_4 s_5$ in Figure 3 may be grouped together after clustering. Since each frequent N-gram originates from a specific *(question, answer)* pair in the database, center g_i^* is therefore associated with a set of questions, denoted by $\mathcal{Q}_i$, w.r.t. all the members of cluster $\mathcal{C}_i$. Thus, the proposed method outputs a list of pairs $(\mathcal{Q}_i, g_i^*)$ as the results of frequent N-gram mining.

In our implementation, frequent N-gram mining is performed under a MapReduce framework where text segmentation and N-gram generation are processed in Map job while merging and counting are processed in Reduce job. The results are post-processed by the two-step filtering and the clustering. The output pairs $(\mathcal{Q}_i, g_i^*)$ consist of the new QA database.

B. Medical Knowledge Abstraction

After frequent segment N-gram mining, each cluster discusses a specific field of medical knowledge, e.g. the symptoms of gastritis or the medicine for hypertension. Next, we extract a compound representation of the questions for each pair $(\mathcal{Q}_i, g_i^*)$ so that if a patient's question Q_u matches with the compound representation, Q_u can be answered by g_i^* as well. We name this extraction process the **Medical Knowledge Abstraction** (MKA).

The major task of MKA is to extract feature representation for each pair $(\mathcal{Q}_i, g_i^*)$. Given $\mathcal{Q}_i = \{Q_1, ..., Q_n\}$ of the i-th cluster, we extract and count the keywords of all questions after stemming, removing the stop words and only preserving the nouns and the verbs which are the most critical parts in sentences [3]. Then, the TF-IDF feature $\mathbf{f}_i^{tfidf}$ is extracted as its textual representation. Specifically, we consider $\mathcal{Q}_i$ as a document that is composed of the many paragraphs $\{Q_1, ..., Q_n\}$. The extracted critical keywords form the vocabulary and they are used to compute the document-level term frequency and inverse document frequency, based on which the TF-IDF feature is obtained.

Besides, the structured data of each question in $\mathcal{Q}_i$ are also utilized to construct the structural feature (named by the structured data). The structured data of a medical question can be: the medical department of the question (e.g. gastroenterology, gynecology and orthopedics), the gender and age of the patient. We use the probability distribution of the medical departments of the questions in $\mathcal{Q}_i$ as its structural feature denoted by $\mathbf{f}_i^{struct}$. In this study, $\mathbf{f}_i^{tfidf}$ and $\mathbf{f}_i^{struct}$ are called the *medical knowledge abstraction* of $(\mathcal{Q}_i, g_i^*)$.

Figure 1 shows some real-world examples of shared segment N-grams and medical knowledge abstraction from our corpus. The three questions in Figure 1 are all about the stomach issues but are textually different in the patient's description. The doctors' answers to the three questions share many answer segments which are about the medicine suggestion, the check recommendation and the cautions. The keywords *stomach*, *stomach bloating*, *stomach pain*, *stomach sour* and *acid water* are extracted from the three questions, which are further used to construct the textual representation $\mathbf{f}_i^{tfidf}$ of MKA. That is, when a patient's question highly matches with the keywords in the last column in Figure 1, it is very likely that (s)he is asking about the stomach issues, and thus the segment N-grams in the third column in Figure 1 are potentially parts of the final answer to the patient. This facilitates the finding or generation of more accurate answers due to the incorporation of answer segments, which distinguishes itself from the state-of-the-art answer selection methods [7], [13], [23], [25].

C. Medical Segment Matching

When a patient raises a new medical question Q_u which does not match with any answer using the traditional IR-QA or KB-QA methods [7], [13], [23], [25], it triggers our system for the further matching based on MKA. The proposed medical segment matching consists of two steps: basic text ranking and neural re-ranking.

Basic Text Ranking. Similar to MKA, the textual feature and the structural feature of the new medical question can be extracted, denoted by $\mathbf{h}^{tfidf}$ and $\mathbf{h}^{struct}$, respectively. The proposed basic text ranking function is defined below:

$$m((\mathcal{Q}_i, g_i^*), Q_u) = \alpha \frac{\mathbf{f}_i^{tfidf} \cdot \mathbf{h}^{tfidf}}{\|\mathbf{f}_i^{tfidf}\| \|\mathbf{h}^{tfidf}\|} + (1-\alpha) \frac{\mathbf{f}_i^{struct} \cdot \mathbf{h}^{struct}}{\|\mathbf{f}_i^{struct}\| \|\mathbf{h}^{struct}\|}, \quad (1)$$

where $0 \leq \alpha \leq 1$ is a parameter to balance the weight of textual similarity and that of structural similarity. The empirical setting is $\alpha = 0.8$ in our evaluation. $\|\cdot\|$ denotes the L2-norm. The frequent segment N-gram answers g_i^* are

[3]For Chinese corpus, we extract keywords using the TextRank method [24] in the Jieba Package https://github.com/fxsjy/jieba.

first ranked based on Eq. (1). The top-k (e.g. $k = 100$) most similar answers are selected for neural re-ranking to avoid the massive online computation cost of neural networks over large pool of candidates.

Neural Re-ranking. The basic text ranking is able to match the patient's question with MKA on the level of keyword co-occurrences and meta data similarity. We further employ neural networks to uncover the latent relevance between patient's question and the MKA results.

The top-k most similar answers obtained from basic text ranking will be re-ranked using the neural ne Firstly, the TF-IDF feature of the patient question and that of a candidate abstraction are concatenated together with their textual and their structural cosine similarity. The concatenated feature is then fed into a multi-layer perceptron whose last layer is activated by the Sigmoid function while the rest activated by the ReLU function. Eqs. (2)–(5) show the layer-wise computation towards the output score. $\mathbf{W}^1$, $\mathbf{W}^2$, $\mathbf{W}^3$, $\mathbf{b}^1$, $\mathbf{b}^2$ and $\mathbf{b}^3$ are model parameters.

$$\mathcal{F}^0(Q_u, (\mathcal{Q}_i, g_i^*)) = [\frac{\mathbf{h}^{tfidf} \cdot \mathbf{f}_i^{tfidf}}{\|\mathbf{h}^{tfidf}\| \|\mathbf{f}_i^{tfidf}\|}, \frac{\mathbf{h}^{struct} \cdot \mathbf{f}_i^{struct}}{\|\mathbf{h}^{struct}\| \|\mathbf{f}_i^{struct}\|}, \frac{\mathbf{h}^{tfidf}}{\|\mathbf{h}^{tfidf}\|}, \frac{\mathbf{f}_i^{tfidf}}{\|\mathbf{f}_i^{tfidf}\|}]^\top, \quad (2)$$

$$\mathcal{F}^1 = ReLU(\mathbf{W}^1 \mathcal{F}^0(Q_u, (\mathcal{Q}_i, g_i^*)) + \mathbf{b}^1), \quad (3)$$

$$\mathcal{F}^2 = ReLU(\mathbf{W}^2 \mathcal{F}^1 + \mathbf{b}^2), \quad (4)$$

$$score = \sigma(\mathbf{W}^3 \mathcal{F}^2 + \mathbf{b}^3) \quad (5)$$

The model is trained in a pairwise manner. For a mined cluster $(\mathcal{Q}_i, g_i^*)$, any question $Q \in \mathcal{Q}_i$ and center $(\mathcal{Q}_i, g_i^*)$ consists of a positive pair of instance. Then, we randomly sample another cluster $(\mathcal{Q}_j, g_j^*)$, and pair Q and $(\mathcal{Q}_j, g_j^*)$ as a negative instance. The marginal Hinge loss (Eq. 6) is used as the loss function in the training where D_Q^{neg} is the set of negative samples w.r.t. the i-th cluster. The Adam algorithm [26] is used as the optimizer. To increase the difficulty of discrimination between the positive and negative pairs, we sample the negative instances which have the same medical department with Q to construct D_i^{neg}. Based on our evaluation, the empirical setting of the margin M is 0.2.

$$\mathcal{L} = \sum_{\mathcal{Q}_i, g_i^*} \sum_{Q \in \mathcal{Q}_i} \sum_{j \in D_i^{neg}} max(0, M - (score(Q, (\mathcal{Q}_i, g_i^*)) - score(Q, (\mathcal{Q}_j, g_j^*)))). \quad (6)$$

For a question Q_u, if $\max_i score(Q_u, (\mathcal{Q}_i, g_i^*)) \geq \tau$ where τ is the score threshold, e.g. $\tau = 0.8$, the proposed method will return the frequent segment N-gram answer w.r.t. the maximum score. Otherwise, the proposed method does not generate an answer.

D. Answer Re-retrieval

In most cases, the obtained frequent segment N-gram g_i^* is not well packed as a strictly qualified answer because g_i^* is composed of some parts of a real answer and thus it may not be syntactically complete. g_i^* should be augmented as a real answer before presenting to the patient.

To deal with this issue, we propose *answer re-retrieval* which guarantees that the returned candidate answer is a fully doctor-editted answer that exists in our QA database, which reduces the risk of incorrect medical information. We name it *answer re-retrieval* because this step is the second full-text IR query in the workflow where both the patient's question and the matched knowledge abstraction form the query. In contrast, the first IR query is in the beginning of MelodyQA workflow and it only consists of the patient's question.

Specifically, Q_u is used to match with historical questions and g_i^* is used to match with historical answers. The relevance scores computed on Q_u and g_i^* are summed as the final metric. The pair (Q^*, A^*) w.r.t. the largest score is returned and A^* is presented to the doctor. In our implementation, ElasticSearch is used as the IR engine.

V. Experimental Results

KAM is currently collaborating with MelodyQA in automatically providing answers to the medical questions from the web users in Muzhi Doctor of Baidu. Both KAM and MelodyQA aim at reducing the time and effort that the doctors need to answer patients' questions. Thus, there are two main metrics to measure the performance of the system: **Coverage** and **Approve Rate**.

Coverage (abbr. **Cov**) is the percentage of questions that can be answered by MelodyQA. If no answer is returned for question Q, then Q is not *covered* by the system. Formally, if MelodyQA receives M questions among which N are returned with non-empty answers [4], the coverage of MelodyQA is $\frac{N}{M}$.

Approve Rate (abbr. **AR**) measures how often the doctors approve the answers returned by MelodyQA. In the evaluation, we consider both *approve directly* and *approve with minor revision* as successful approval. **AR** can be interpreted as a kind of *accuracy* measurement since the doctors approve the answers only when the answers are correct.

Before incorporating KAM, MelodyQA has been steadily developed for multiple times and has been providing stable service online for a long time. The goal of KAM aims at improving the coverage of MelodyQA while preserving its approve rate. Thus, we mainly evaluate the improvement of coverage after using KAM in later experiments while keeping the approve rate at its previous level.

A. Evaluation Results

We collected an evaluation dataset which consists of over 210,000 questions generated in an online medical consultation platform within two consecutive weeks in December, 2017. We first run MelodyQA alone on this dataset and obtain the coverage as 17.5% (see Table I). Then, KAM is incorporated into MelodyQA that when MelodyQA does not return any candidate answer for an input question Q, Q will be fed into KAM to search for answers again. Thus, we see a *net coverage increase* of 1.8% brought by KAM, which leads to a final coverage as 19.3%. Meanwhile, when running KAM alone, the coverage on the same dataset is 5.3%.

[4]The rest $M - N$ questions will be manually answered by doctors.

TABLE I

THE EVALUATION RESULTS. *Non-doc*: THE AVERAGE RESULT OF VOLUNTEERS WHO ARE NOT DOCTORS. *Doc*: THE RESULT GENERATED BY A REAL CERTIFICATED DOCTOR. *Avg*: THE AVERAGE RESULTS.

Group	MelodyQA	KAM	MelodyQA+KAM
Cov	17.5%	5.3%	19.3%
Non-doc AR	70.6%	71.8%	70.7%
Doc AR	72.9%	73.1%	72.9%
Avg AR	71.0%	72.0%	71.1%

Next, we invite some volunteers to conduct the evaluation on the approve rate. There are three non-doctor volunteers (with basic medical knowledge) and a real certificated doctor in the evaluation. Firstly, MelodyQA+KAM runs on the evaluation dataset and generates a pool of (Q, A) pairs as results. Since KAM returns at most one candidate answer each time, we only preserve the pairs with only one candidate answer generated by MelodyQA alone. Then, we separately and ramdonly sample 300 pairs generated by MelodyQA alone and another 300 pairs generated by KAM from the pool of pairs. The total 600 (Q, A) pairs are mixed together and randomly shuffled so that the participants does not know whether a given (Q, A) pair is generated by MelodyQA or KAM in the blind evaluation. Each participant is given about 200 (Q, A) pairs to judge if Q can be answered by the corresponding A or not. The participant do not knowing where the pair comes from. The approve rate is computed based on the participants' judgement.

Table I shows the evaluation results. In each group of participants, the approve rate of MelodyQA with KAM is slightly higher than that of MelodyQA without KAM, which validates the effectiveness of incorporating the proposed method. Besides, there is 1.8% net improvement of coverage after using KAM. It must be justified that it is very difficult to perform automatic non-factoid question answering in the medical domain because there is almost zero tolerance of mistake when dealing with people's health issues. Besides, MelodyQA has come to a bottleneck after evolving for multiple times and it has already been equipped with all the practical and advanced techniques like information retrieval, intent matching,deep QA similarity and manually defined rules. Therefore, it is already very difficult to improve the coverage of MelodyQA while preserving a high-level approve rate. Hence, the evaluation shows that KAM can improve the coverage of MelodyQA while keeping its approve rate high enough. KAM has been incorporated into MelodyQA to provide high-quality medical QA service. The online performance is close to Table I.

VI. CONCLUSION

We propose a novel knowledge abstraction matching method on top of a well-developed system to tackle the medical QA problem. It attempts to bridge the gap between medical questions and answers by utilizing the medical knowledge abstractions. The novelty of the method lies in the construction of segment N-grams and the medical knowledge abstraction as well as the matching. The evaluation conducted on the real-world dataset shows the effectiveness of the proposed method.

REFERENCES

[1] R. M. Terol, P. Martínez-Barco, and M. Palomar, "A knowledge based method for the medical question answering problem," *Computers in biology and medicine*, vol. 37, no. 10, pp. 1511–1521, 2007.

[2] D. Demner-Fushman and J. Lin, "Answering clinical questions with knowledge-based and statistical techniques," *Computational Linguistics*, vol. 33, no. 1, pp. 63–103, 2007.

[3] T. R. Goodwin and S. M. Harabagiu, "Medical question answering for clinical decision support," in *CIKM*. ACM, 2016, pp. 297–306.

[4] S. A. Hasan, S. Zhao, V. V. Datla, J. Liu, K. Lee, A. Qadir, A. Prakash, and O. Farri, "Clinical question answering using key-value memory networks and knowledge graph," in *TREC*, 2016.

[5] G. Wiese, D. Weissenborn, and M. Neves, "Neural domain adaptation for biomedical question answering," in *CoNLL*. Association for Computational Linguistics, August 2017, pp. 281–289.

[6] H. Scells, G. Zuccon, B. Koopman, A. Deacon, L. Azzopardi, and S. Geva, "Integrating the framing of clinical questions via pico into the retrieval of medical literature for systematic reviews," in *CIKM*. ACM, 2017, pp. 2291–2294.

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

[8] W. Cui, Y. Xiao, H. Wang, Y. Song, S.-w. Hwang, and W. Wang, "Kbqa: learning question answering over qa corpora and knowledge bases," *PVLDB*, vol. 10, no. 5, pp. 565–576, 2017.

[9] D. R. Radev, H. Qi, Z. Zheng, S. Blair-Goldensohn, Z. Zhang, W. Fan, and J. Prager, "Mining the web for answers to natural language questions," in *CIKM*. ACM, 2001, pp. 143–150.

[10] M. Heilman and N. A. Smith, "Tree edit models for recognizing textual entailments, paraphrases, and answers to questions," in *NAACL-HLT*. Association for Computational Linguistics, 2010, pp. 1011–1019.

[11] A. Severyn, A. Moschitti, M. Tsagkias, R. Berendsen, and M. De Rijke, "A syntax-aware re-ranker for microblog retrieval," in *SIGIR*. ACM, 2014, pp. 1067–1070.

[12] X. Yao, B. Van Durme, C. Callison-Burch, and P. Clark, "Answer extraction as sequence tagging with tree edit distance," in *NAACL-HLT*, 2013, pp. 858–867.

[13] C. N. Dos Santos, M. Tan, B. Xiang, and B. Zhou, "Attentive pooling networks," *arXiv preprint arXiv:1602.03609*, 2016.

[14] Y. Tay, M. C. Phan, L. A. Tuan, and S. C. Hui, "Learning to rank question answer pairs with holographic dual lstm architecture," in *SIGIR*. ACM, 2017, pp. 695–704.

[15] A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," in *SIGIR*, 2015, pp. 373–382.

[16] C. Dos Santos, L. Barbosa, D. Bogdanova, and B. Zadrozny, "Learning hybrid representations to retrieve semantically equivalent questions," in *ACL-IJCNLP*, vol. 2, 2015, pp. 694–699.

[17] X. Qiu and X. Huang, "Convolutional neural tensor network architecture for community-based question answering." in *IJCAI*, 2015, pp. 1305–1311.

[18] Y. Yang, W.-t. Yih, and C. Meek, "Wikiqa: A challenge dataset for open-domain question answering," in *EMNLP*, 2015, pp. 2013–2018.

[19] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *ACL-IJCNLP*, vol. 1, 2015, pp. 1556–1566.

[20] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *EMNLP*, 2015, pp. 632–642.

[21] M. Tan, C. dos Santos, B. Xiang, and B. Zhou, "Improved representation learning for question answer matching," in *ACL*, 2016, pp. 464–473.

[22] L. Yang, Q. Ai, J. Guo, and W. B. Croft, "anmm: Ranking short answer texts with attention-based neural matching model," in *CIKM*. ACM, 2016, pp. 287–296.

[23] S. Wang and J. Jiang, "A compare-aggregate model for matching text sequences," in *ICLR*, 2017.

[24] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *EMNLP*, 2004.

[25] J. Rao, H. He, and J. Lin, "Noise-contrastive estimation for answer selection with deep neural networks," in *CIKM*, 2016, pp. 1913–1916.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *arXiv:1412.6980*, 2014.