Knowledge Abstraction Matching for Medical Question Answering

Jun Chen
Baidu Inc.
chenjun22@baidu.com

Jingbo Zhou
Baidu Inc.
zhoujingbo@baidu.com

Zhenhui Shi
Baidu Inc.
shizhenhui@baidu.com

Binjun Zhu
Baidu Inc.
zhubinjun@baidu.com

Bin Fan
Baidu Inc.
fanbin@baidu.com

Chengliang Luo
Baidu Inc.
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 the Artificial Intelligent (AI) in healthcare. Though many efforts have been made to improve the start-of-the-art of the QA system, the medical QA system still deserve to delicate algorithm design and special technique optimization due to the serious application scenario and strick requirement for the answer quality. In this paper, we introduce a novel Knowledge Abstraction Matching (KAM) method for the medical QA problem. Our 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 for the medical QA which consists of four steps: frequent segment N-gram mining, medical knowledge abstraction, medical segment matching and answer re-retrieval. The KAM method has been incorporated into Baidu's enterprise medical QA system MelodyQA deployed on the backbend of Muzhi Doctor. The evaluation demonstrates that the proposed method can generate more quality answers for MelodyQA with a significant improvement of question coverage under acceptable accuracy.

CCS CONCEPTS

• **Information systems** → **Question answering**; *Data mining*; *Similarity measures*; • **Computing methodologies** → **Information extraction**;

KEYWORDS

Medical Question Answering, Knowledge Abstraction Matching, Information Retrieval-based QA

1 INTRODUCTION

Question answering (QA) studies the problem of automatically finding or generating answers for users' questions. QA has broad real-life applications like personal digital assistant, intelligent customer service as well as the major topic of this study – medical QA where the technology is used to answer patients' medical questions online. Medical QA is one of the main applications of the artificial intelligent in health care. It attracts special research attention due to its own challenges like the high requirement of answer quality, the complex medical entities and the domain specific knowledge. The delicate algorithm design and special technique optimization for the medical QA system is still desirable. In the past decade, there have already been many efforts devoted to the study of the medical QA problem from different perspectives [7, 13, 15, 29, 29, 39, 44].

The research of this paper is based on Baidu's enterprise medical QA system (denoted by **MelodyQA**) deployed on the backbend of Muzhi Doctor [1], which is an online platform of Baidu that enables patients to consult with doctors through internet for professional medical advice, preventive nursery care, post-diagnosis services, disease management, medication alerts and more. One of the popular services provided by Muzhi Doctor is a free service of "the single round question&answer" where the user posts a question on the website and waits for the response from a doctor. MelodyQA is designed in a Business-to-Doctor-to-Customer style for such single round question answer service. It does not directly present the answer to the patient. Instead, after receiving a question from an online patient, MelodyQA 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. It is worthy noting that it is very difficult (and almost impossible) to perform automatic non-factoid question answering in the medical domain due to the following reasons:

- Most of patients' questions cannot be answered by simple and short facts. The medical answers are usually very long and complex that require professional knowledge.
- There is almost zero tolerance of mistake when dealing with patients' health issues. Thus, the quality of the answers must be high enough, which means the answers should get approved by the certificated doctors.

CIKM'18, October 2018, Turin, Italy

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

[1] http://muzhi.baidu.com/

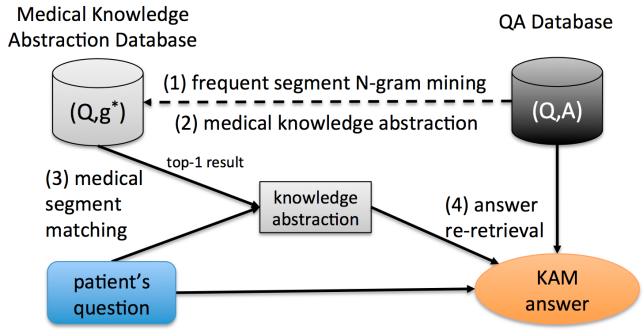

Figure 1: Overview of the Knowledge Abstraction Matching method. Step (1) and (2) are pre-processed offline. Step (3) and (4) are processed online.

The objective of MelodyQA is to improve the doctors' work efficiency with AI technology. It enables the doctors to answer medical questions by simply clicking or revising only several words among the candidates. MelodyQA has been providing stable service online 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 QA similarity 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 a novel **K**nowledge **A**bstraction **M**atching (**KAM**) method for the medical QA problem. KAM belongs to the category of the IR-QA method, 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 re-retrieve the final qualified answer.

The invent of KAM is inspired by our observation that there are usually many repeated text segments across different historical answers. The patients describe differently on the same diseases and symptoms, especially considering that the patients do not have enough medical knowledge. However, for the diverse description of the same symptom or disease, the doctor may response with the same answer. We also observe that for different questions there may be some common sentences or segments appearing in the answers in our database. Therefore, the questions sharing the same text segments in the answers may 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. We can further generate a medical knowledge abstraction by extracting feature representation from the question cluster. Fig. 2 shows some samples in our medical corpus where the answers from different doctors share several same segments. The three questions are textually different from each other but all about the stomach issues. All the answers share the segments that mention, for example, *no spicy food*, *Clarithromycin*, *gastroscopy* and *barium meal check*.

Knowledge abstraction figures out a new way to connect patient's question with the historical questions and answers in the database. On the one hand, the knowledge abstraction formed by the question clustering can represent the features of certain medical domain. On the other hand, a patient usually cannot provide a full description for certain diagnosis (e.g. missing some symptoms, forgetting some bad habits and hiding some historical diseases). Knowledge abstraction can be considered as a kind of knowledge completion of a question. Therefore, we can use knowledge abstraction to augment the patient's question to retrieve better candidate answers from historical QA database.

Fig. 1 shows an overview of KAM, which has 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 discover the frequent segment N-grams in the historical answers. These N-grams are important patterns that each can be used to answer a group of related questions. Step (2) is an offline pre-processing where we abstract the representation of the medical knowledge for each group of related questions obtained from Step (1). When a patient raises a new medical question in an online QA task, it trigger Step (3) of KAM to match the patient's question with the medical knowledge abstraction. Finally, we use the patient's question and the matched knowledge abstraction from Step (3) to re-retrieve the final quality answer from the historical QA database.

Our evaluation in the real application scenario also demonstrates the effectiveness of KAM. For an input question, if MelodyQA dose not return any candidate answer, we employ KAM method to process this question. Thus, KAM handles more difficult questions than the ones handled by MelodyQA because the easy ones have already been covered by MelodyQA. Our experiment shows that KAM can significantly improve the coverage of MelodyQA while keeping the same answer quality.

We summarise our contribution as follows:

- We propose KAM (Knowledge Abstraction Matching), a new method to solve the medical QA problem. KAM aims to handle the input questions which cannot be covered by Baidu's enterprise non-factoid medical question answering system, MelodyQA.
- We introduce a new discovery that in medical QA corpus there are usually some frequent text segments appearing in the answers across different questions. This observation inspires us to cluster the questions which share some text segments in the answers. This new question clustering method forms the intuition behind the KAM method.

Question	Answer	Shared Segment N-Grams	Medical Knowledge Abstraction
胃不舒服，有时涨,有时感觉肚子里像吃了辣椒一样,怎么办? My stomach is uncomfortable. Sometimes it gets bloating and sometimes it feels like eating chilli in my stomach. What should I do?	胃酸胃胀,胃痛,烧心反酸是胃炎的表现,[1]首先应该清淡饮食,不吃辛辣刺激食物,少食多餐!另外建议规律服药,如兰索拉唑,[2]三九胃泰,克拉霉素,如果效果不好,[3]建议做胃镜或者钡餐,根据结果治疗. 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]Sanjiu Weitai, Clarithromycin. If not getting better, [3]recommend gastroscopy or barium meal check. Treat according to results.	1. 首先应该清淡饮食，不吃辛辣刺激食物，少食多餐 Firstly, it should be light diet, No spicy food. Small meals and more times. 2. 三九胃泰，克拉霉素， Sanjiu Weitai. Clarithromycin. 3. 建议做胃镜或者钡餐,根据结果治疗. Recommend gastroscopy or barium meal check. Treat according to results.	肚子 stomach, 胃胀 stomach bloating, 胃痛 stomach pain, 胃酸 stomach sour, 酸水 acid water
胃胀胃痛特别是有点饿的时候更严重,怎么回事? My stomach bloating and pain get much worse when feeling hungry. What's wrong?	胃胀,胃痛,烧心，再就是有饥饿痛,这是胃炎的典型的症状.[1]首先应该清淡饮食,不吃辛辣刺激食物,少食多餐,多吃蔬菜水果,另外建议服用相应的药物,如兰索拉唑,[2]三九胃泰,克拉霉素,如果效果不好的话,[3]建议做胃镜或者钡餐,根据结果治疗. 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]Sanjiu Weitai, Clarithromycin. If not getting any better, [3]recommend gastroscopy or barium meal check. Treat according to results.		
胃酸,经常有酸水吐出.请问怎么回事? Feeling sour in stomach. Often spit out with acid water. What's wrong with me?	你好反酸烧心食欲不振,这都是胃炎的表现,[1]首先应该清淡饮食,不吃辛辣刺激食物,少食多餐,另外建议规律服药,如奥美拉唑,[2]三九胃泰,克拉霉素,如果症状不减轻,[3]建议做胃镜或者钡餐,根据结果治疗. 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]Sanjiu Weitai, Clarithromycin. If there is no relief of symptoms, [3]recommend gastroscopy or barium meal check. Treat according to results.		

Figure 2: Examples of some shared segments across different answers and medical knowledge abstraction. The shared segments are highlighted in red color. *Sanjiu Weitai* is a kind of traditional Chinese medicine for the stomach issues.

- We present a 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. Finally, the matched knowledge abstraction and the patient's question are jointly used to re-retrieve the quality answer in the QA database.
- We conduct the evaluation on top of the MelodyQA system to demonstrate the effectiveness of KAM method in dealing with the medical QA problem.

The rest of the paper is organized as follows. We will discuss the related work in Section 2, followed by the detailed introduction of KAM in Section 3. Then we will evaluate our method is Section 4, and conclude the paper in Section 5.

2 RELATED WORK

In this section, we first present a brief review of of related work of question answer. Next, we investigate literatures about the question answer in medical domain.

2.1 Question Answering

Question Answer (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 [53]. One of the main challenges of KB-QA is how to translate the questions into structure queries like SPARQL and SQL [5, 6, 24, 46, 52, 53]. The end-to-end neural network approach is also investigated for such query translation [14]. The KB-QA systems are usually more suitable for answering factoid question which has only simple relations among the question's entities [5]. The IR-QA systems retrieve text documents that are the most relevant to the question [1, 20, 26, 28]. The text documents can be historical question-answers pairs that can answer the question directly, or can be the relevant documents from which we can extract answers.

Our KAM method belongs to the IR-QA category since its objective is to retrieve the candidate answers from historical QA data. The IR-QA has a quite long history with the advancement in IR research. Previous work have been primarily focused on lexical and syntactic feature engineering based approach, such as semantic features constructed based on WordNet [50], syntactical feature matching on the question/answer parse trees [41], and automatic discriminative tree-edit features extraction over parsing trees [30]. Different machine learning models have also been adopted for this task, such as the Tree Edit Distance (TED) model [17], Support Vector Machines (SVMs) [32] and Conditional Random Fields (CRFs) [49]. While these methods show effectiveness after the effort of feature engineering, in recent years there are more and more evidences that such feature engineering based

approaches have been outperformed by deep learning based approaches [10, 37, 38].

The deep learning architecture approaches to IR-QA generally is to learn low-dimensional representations of question and answer which can be used as input features [31, 51]. The question and answer representations can be separate vectors for matching by a similarity metrics [8], or joint feature vectors for inputting to classification or learning-to-rank model [25, 40]. The network structure for such question answer feature matching can be divided into three categories: siamense network, attentive network, and compare-aggregate network [42]. The siamense network uses the same structure, such as RNN or CNN, to build the representations for the question and answer separately. Then similarity matching metrics such as cosine similarity [12, 48], element-wise operation [22, 34] and tensor layer combination [3] are used for question answer matching. With utilizing the soft-attention mechanism, the attentive network uses weighted sum of all the states of RNN for question answer matching (instead of simply use the final state) [10, 35, 47]. The compare-aggregate network performs the word level matching [2, 16, 23, 42].

A sub-area of IR-QA is community question answering (CQA) which ranks answer from CQA websites for a particular question to boost the best answer to the top position [4, 18]. The feature types [33] and ranking models [18, 45] for CQA have also been studied in previous work. Many techniques and models of IR-QA can also be applied to CQA problem. The application scenario of KAM is slight different from the CQA problem: the matched answer is not showed to users directly but to certified doctors for verification and editing.

2.2 Medical Question Answering

A close related domain of the medical QA in this study is clinical QA (sometime it is also referred as medical QA), which usually is 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) [7, 13, 15, 29]. Our medical QA system works on the historical question-answer pairs generated by the patients and doctors which is quite different from the scientific articles and patient's information document. Therefore, the techniques of the clinical QA cannot be applied directly to our medical QA system.

There are also some previous work about question answer in medical domain. One of the pioneering systems of the medical QA system is presented in [39], 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. The authors of [44] use transfer learning and biomedical word embedding to improve the performance on medical QA. How to translate medical questions into SPARQL query for KB-QA with medical entity extraction and sematic recognition is investigated in [29].

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

3 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. The first two steps are offline mining processes. In the step 1) we find the frequent text segment from answers of different questions, and then try to cluster the questions whose answers contain the same frequent text segment. In the step 2) we extract knowledge abstraction from question clustering mined in step 1). The last two steps are online processes, when a new input question arises in the online QA task, in step 3) we match the question with the medical knowledge abstraction, and in step 4) we use the question and the knowledge abstraction to re-retrieve the answers from the historical QA database.

3.1 Frequent Segment N-gram Mining

Fig. 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 Fig 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 ($s_1s_2s_3s_4$). Without specific statement, N-gram in this paper means segment N-gram.

Next, the same N-grams are merged and counted. Then, 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 Fig. 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 clustering method DBSCAN [11] 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 $\mathcal{C}_i$.

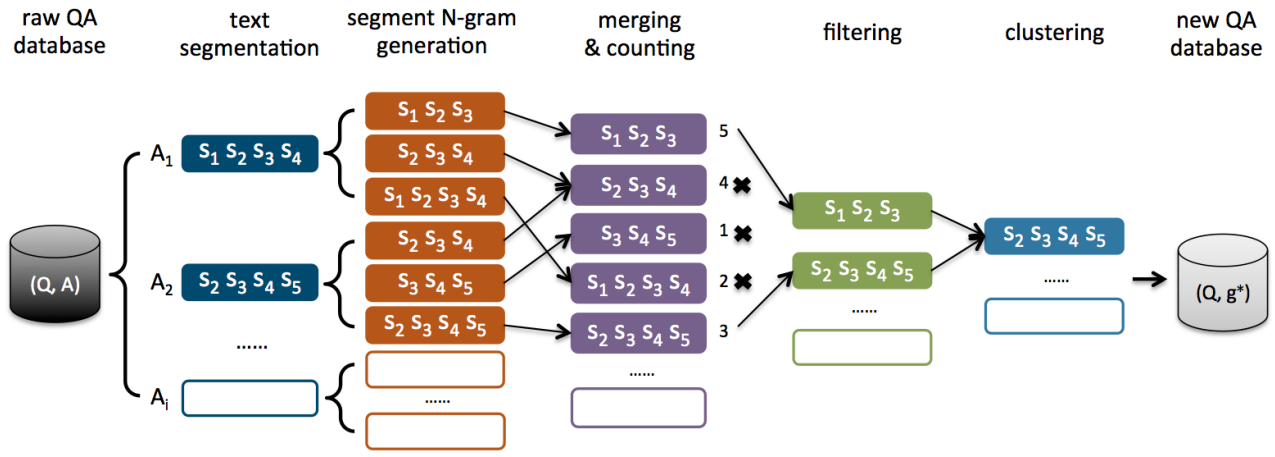

Figure 3: Frequent segment N-gram mining. $\eta = 3$.

The longest member $g_i^* = \arg\max_{g \in \mathcal{C}_i} \text{length}(g)$ is selected as the center of cluster $\mathcal{C}_i$. For example, $s_1 s_2 s_3$ and $s_2 s_3 s_4 s_5$ in Fig. 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 the Map job while merging and counting are processed in the 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.

3.2 Medical Knowledge Abstraction

After frequent segment N-gram mining, each cluster discusses a specific field of medical knowledge, e.g. the treatment for influenza, the symptoms of gastritis or the medicine for hypertension. The questions w.r.t. a given field of medical knowledge can be very textually different, but may have similar answers. The frequent segment N-gram mining tackles this issue by grouping the questions which share many textual segments. Thus, we propose to abstract the medical knowledge based on the mining results $(\mathcal{Q}_i, g_i^*)$.

The major task of medical knowledge abstraction is to extract feature representation for each pair $(\mathcal{Q}_i, g_i^*)$. Given the mining result $\mathcal{Q}_i = \{Q_1, ..., Q_n\}$ 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 [2]. Then, the TF-IDF feature $\mathbf{f}_i^{tfidf}$ is extracted based on the keywords of each $\mathcal{Q}_i$ as its textual representation.

[2]For Chinese corpus, we extract keywords using the TextRank method [21] in the Jieba Package https://github.com/fxsjy/jieba.

Besides, the meta data of each question in $\mathcal{Q}_i$ are also utilized to construct the structural feature. The meta 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 $\mathbf{f}_i^{struct}$.

$\mathbf{f}_i^{tfidf}$ and $\mathbf{f}_i^{struct}$ are the medical knowledge abstraction of $(\mathcal{Q}_i, g_i^*)$. As the base representation before matching, the extraction of $\mathbf{f}_i^{tfidf}$ and $\mathbf{f}_i^{struct}$ in this study is efficient yet effective, but they are also open to other sophisticated implementations.

Fig. 2 shows some examples of shared segment N-grams and medical knowledge abstraction from our medical corpus. The three questions in Fig. 2 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 the medical knowledge abstraction. That is, when a patient's question highly matches with the keywords in the last column in Fig. 2, it is very likely that (s)he is asking about the stomach issues, and thus the segment N-grams in the third column in Fig. 2 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 [9, 27, 36, 43]. In Sec. 3.3, the method of matching patient's question with the medical knowledge abstraction will be discussed.

3.3 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

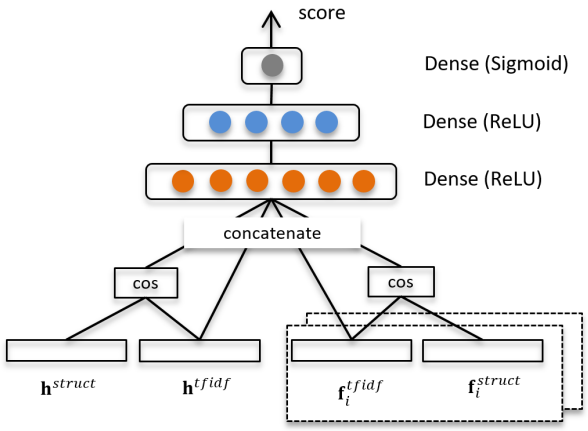

Figure 4: Re-ranking with neural networks.

KB-QA methods [9, 27, 36, 43], it triggers our system for the further matching based on the medical knowledge abstraction (Sec. 3.2). The proposed medical segment matching consists of two steps: basic text ranking and neural re-ranking.

Basic Text Ranking. Similar to Sec. 3.2, 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. We set $\alpha = 0.8$ in our evaluation. $\|\cdot\|$ denotes the L2-norm. The frequent segment N-gram answers g_i^* are 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 the medical knowledge abstraction 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 medical knowledge abstraction.

The top-k most similar answers obtained from basic text ranking will be re-ranked using the neural network shown in Fig. 4. The model takes as input the TF-IDF feature $\mathbf{h}^{tfidf}$ and meta feature $\mathbf{h}^{struct}$ of patient's question as well as the TF-IDF feature $\mathbf{f}_i^{tfidf}$ and meta feature $\mathbf{f}_i^{struct}$ of the medical knowledge abstraction. Please note that each $(\mathcal{Q}_i, g_i^*)$ of the top-k most similar medical knowledge abstraction from basic text ranking is a candidate.

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)–(6) 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 the parameters of this model.

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

$$\frac{\mathbf{h}^{tfidf}}{\|\mathbf{h}^{tfidf}\|},\frac{\mathbf{f}_i^{tfidf}}{\|\mathbf{f}_i^{tfidf}\|}]^\top, \quad (3)$$

$$\mathcal{F}^1 = ReLU(\mathbf{W}^1\mathcal{F}^0(Q_u,(\mathcal{Q}_i,g_i^*))+\mathbf{b}^1), \quad (4)$$

$$\mathcal{F}^2 = ReLU(\mathbf{W}^2\mathcal{F}^1+\mathbf{b}^2), \quad (5)$$

$$score = \sigma(\mathbf{W}^3\mathcal{F}^2+\mathbf{b}^3) \quad (6)$$

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. 7) 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, and the Adam algorithm [19] 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.

$$\mathcal{L} = \sum_{\mathcal{Q}_i,g_i^*}\sum_{Q\in\mathcal{Q}_i}\sum_{j\in D_i^{neg}} max(0, M-(score(Q,(\mathcal{Q}_i,g_i^*)) - score(Q,(\mathcal{Q}_j,g_j^*)))). \quad (7)$$

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.

3.4 Answer Re-retrieval

The last step, named *answer re-retrieval*, 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 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.

In most cases, the obtained frequent segment N-gram g_i^* from Sec. 3.3 is not well packed as a strictly qualified answer to a medical question 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.

In this work, we employ answer re-retrieval to deal with this issue. Specifically, another IR query (Q_u, g_i^*) will be performed to jointly index questions and answers with Q_u

and g_i^*, respectively, in the QA database. The relevance scores computed on Q_u and g_i^* are summed as the final metric. The real answer A^* w.r.t. the largest score is returned and presented to the patient. In our implementation, Baidu improved ElasticSearch [3] is used as the IR engine. More complicated question answer similarity ranking method after the IR search in ElasticSearch is possible, which is consistent with our deep learning-based answer ranking method after IR search in MelodyQA. However, the details of the MelodyQA's ranking method is beyond the scope of KAM and it is also not evaluated in the experiment of this paper.

4 EMPIRICAL EVALUATION

KAM is currently used as an important module of MelodyQA in Baidu. MelodyQA 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 its threshold of confidence score. MelodyQA aims 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.

4.1 Evaluation Results

We collected an evaluation dataset which consists of over 210,000 questions generated in Muzhi Doctor of Baidu within two consecutive weeks in December, 2017. We first run MelodyQA alone on this dataset and obtain the coverage as 17.5% (see Table 1). 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%.

Table 1: 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 of all participants.

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 is run 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 can only see the texts of Q and A without knowing where this pair comes from. The approve rate is computed based on the participants' judgement.

Table 1 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 for medical QA system like information retrieval, entity extraction, intent matching, ranking, deep QA similarity as well as 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 support Baidu's Muzhi Doctor. The online performance is close to

[3] https://github.com/baidu/Elasticsearch
[4] The rest $M - N$ questions will be completely and manually answered by doctors.

Questions	KAM Answers
我是慢性胃炎，会引起左小腹痛吗? I have chronic gastritis, and will it cause my left lower abdominal pain?	有可能引起. 建议口服兰索拉唑, 胶体果胶铋胶囊, 克拉霉素试试, 一定要禁烟, 酒, 咖啡, 茶, 生冷, 辛辣食物. 少吃含淀粉类的食物如: 土豆, 芋头, 粉丝, 粉条, 红薯, 凉粉, 苏打饼干, 碳酸饮料等, 少食多餐, 按时进餐, 不要吃过于坚硬和不消化的食物. 注意饮食, 否则治疗效果不好的. Possible. You may try lansoprazole, colloidal pectin capsules and clarithromycin. No smoking, alcohol, coffee, tea, cold or spicy food is allowed. Eat small meals regularly. Do not eat eat hard and indigestible food like potato, taro, vermicelli, jelly, soda crackers, carbonated drinks, etc. Otherwise, the treatment will not work well.
气不够用,头晕是怎么回事? Feeling short of breath and dizzy. What's wrong?	患者您好,出现这个情况的可能性很多,可以有高血压病,颈椎病,美年尔症等也会出现恶心等情况,建议患者检查下血压和血脂情况看下,如果都是正常的,还是消化道的情况引起的不适.建议消化科看下. Hi. There may be many causes of your symptoms such as hypertension, cervical spondylosis and Meniere's syndrome. It is recommended that the patient check the blood pressure and blood lipids. If the numbers are normal, the discomfort may be caused by digestive issues. And you may be sent for gastroenterology doctors.
感觉肚子大, 一天好几次想上厕所大便, 怎么回事? My belly is bulging. I want to use the bathroom many times a day. What's wrong?	你好, 青年人出现大便次数增多一般是由于饮食不当引起的肠炎, 腹部浮肿的感觉是由于腹部胀气引起的. 建议你, 注意饮食, 避免受凉, 吃清淡易消化食物, 避免吃辛辣刺激性食物, 口服丽珠肠乐治疗, 慢慢调理就会好的. Hi, the increase in the frequency of bowel movements among the young people is generally due to intestinal inflammation caused by improper diet. The feeling of abdominal swelling is caused by abdominal flatulence. You need to pay attention to your diet, eat light and digestible food, avoid cold and spicy food. Orally take P.O.Bifidobiogen, and you will get better.

Figure 5: Examples of answers generated by KAM.

Table 1. Considering the daily fluctuation, the net increase of online coverage is around 2% while the online approve rate is around 72%.

4.2 Some KAM Examples

Fig. 5 shows some examples of the answers generated by KAM on the evaluation dataset. Firstly, the answers found by KAM are capable of answering the corresponding patient's questions. Secondly, the answers given by KAM are comprehensive, complicated and professional, which means the medical QA is a very difficult task and is quite different from the open-domain factoid question answering problem. Last but not the least, the answers given by KAM come from the historical QA data which were manually composed by the certificated doctors before. This guarantees that the answer texts are readable, patient-friendly and the risk of potential health issues is minimized.

Based on the above evaluation, the proposed method is effective in dealing with the medical question answering problem. It brings about 2% net increase on the coverage of the original well-developed MelodyQA system, while keeping the approve rate at a high level. Considering the fact that MelodyQA has already been incorporated with many advanced QA techniques, such improvement is still significant.

5 CONCLUSION

In this paper, we investigate the medical question answering (QA) problem which is more difficult than the common QA task because the answers to the medical questions are usually lengthy, complex, non-factoid and professional, and there is zero tolerance of mistake when dealing with health issues. We propose a novel knowledge abstraction matching method, named KAM, on top of a well-developed system *MelodyQA* to tackle the medical QA problem in Baidu's Muzhi Doctor service. The intuition behind the KAM is that we find many frequent repeat text segments appearing in the answers across different questions. Therefore, the questions can be clustered based on the same text segment shared by their corresponding answers. Then the knowledge abstractions can be extracted from the question clusters. The proposed method further attempts to bridge the gap between medical questions and answers by utilizing the medical knowledge abstractions. The basic text ranking and the neural re-ranking are used to match the question with the knowledge abstractions, after which the answer re-retrieval is performed to find the final answer. The novelty of this 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 medical dataset shows the effectiveness of the proposed method.

REFERENCES

[1] Kisuh Ahn, Beatrix Alex, Johan Bos, Tiphaine Dalmas, Jochen L Leidner, and Matthew Smillie. 2004. Cross-lingual Question Answering with QED.. In *CLEF (Working Notes)*.

[2] Weijie Bian, Si Li, Zhao Yang, Guang Chen, and Zhiqing Lin. 2017. A Compare-Aggregate Model with Dynamic-Clip Attention for Answer Selection. In *CIKM*. ACM, 1987–1990.

[3] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*. 632–642.

[4] Meng-Fen Chiang, Wen-Chih Peng, and S Yu Philip. 2012. Exploring latent browsing graph for question answering recommendation. *WWW* 15, 5-6 (2012), 603–630.

[5] Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seungwon Hwang, and Wei Wang. 2017. KBQA: learning question answering over QA corpora and knowledge bases. *PVLDB* 10, 5 (2017), 565–576.

[6] Wanyun Cui, Yanghua Xiao, and Wei Wang. 2016. KBQA: An Online Template Based Question Answering System over Freebase.. In *IJCAI*. 4240–4241.

[7] Dina Demner-Fushman and Jimmy Lin. 2007. Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics* 33, 1 (2007), 63–103.

[8] Cicero Dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. 2015. Learning hybrid representations to retrieve semantically equivalent questions. In *ACL-IJCNLP*, Vol. 2. 694–699.

[9] Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive Pooling Networks. In *arXiv:1602.03609*.

[10] Cıcero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive Pooling Networks. *arXiv preprint arXiv:1602.03609* (2016).

[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*. 226–231.

[12] Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *ASRU*. IEEE, 813–820.

[13] Travis R Goodwin and Sanda M Harabagiu. 2016. Medical question answering for clinical decision support. In *CIKM*. ACM,

297–306.
[14] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *ACL)*, Vol. 1. 221–231.
[15] Sadid A Hasan, Siyuan Zhao, Vivek V Datla, Joey Liu, Kathy Lee, Ashequl Qadir, Aaditya Prakash, and Oladimeji Farri. 2016. Clinical Question Answering using Key-Value Memory Networks and Knowledge Graph. In *TREC*.
[16] Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *NAACL-HLT*. 937–948.
[17] Michael Heilman and Noah A Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *NAACL-HLT*. Association for Computational Linguistics, 1011–1019.
[18] Peter Jansen, Mihai Surdeanu, and Peter Clark. 2014. Discourse complements lexical semantics for non-factoid answer reranking. In *ACL*, Vol. 1. 977–986.
[19] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. In *arXiv:1412.6980*.
[20] Baichuan Li and Irwin King. 2010. Routing questions to appropriate answerers in community question answering services. In *CIKM*. ACM, 1585–1588.
[21] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Text. In *EMNLP*.
[22] Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2015. Natural language inference by tree-based convolution and heuristic matching. *arXiv preprint arXiv:1512.08422* (2015).
[23] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *EMNLP*. 2249–2255.
[24] Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. 2003. Towards a theory of natural language interfaces to databases. In *IUI*. ACM, 149–157.
[25] Xipeng Qiu and Xuanjing Huang. 2015. Convolutional Neural Tensor Network Architecture for Community-Based Question Answering.. In *IJCAI*. 1305–1311.
[26] Dragomir R Radev, Hong Qi, Zhiping Zheng, Sasha Blair-Goldensohn, Zhu Zhang, Weiguo Fan, and John Prager. 2001. Mining the web for answers to natural language questions. In *CIKM*. ACM, 143–150.
[27] Jinfeng Rao, Hua He, and Jimmy Lin. 2016. Noise-Contrastive Estimation for Answer Selection with Deep Neural Networks. In *CIKM*. 1913–1916.
[28] Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *ACL*. Association for Computational Linguistics, 41–47.
[29] Harrisen Scells, Guido Zuccon, Bevan Koopman, Anthony Deacon, Leif Azzopardi, and Shlomo Geva. 2017. Integrating the Framing of Clinical Questions via PICO into the Retrieval of Medical Literature for Systematic Reviews. In *CIKM*. ACM, 2291–2294.
[30] Aliaksei Severyn and Alessandro Moschitti. 2013. Automatic feature engineering for answer selection and extraction. In *EMNLP*. 458–467.
[31] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *SIGIR*. ACM, 373–382.
[32] Aliaksei Severyn, Alessandro Moschitti, Manos Tsagkias, Richard Berendsen, and Maarten De Rijke. 2014. A syntax-aware re-ranker for microblog retrieval. In *SIGIR*. ACM, 1067–1070.
[33] Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2008. Learning to rank answers on large online QA collections. In *ACL-HLT*.
[34] Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *ACL-IJCNLP*, Vol. 1. 1556–1566.
[35] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *ACL*, Vol. 1. 464–473.
[36] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. LSTM-based Deep Learning Models for Non-factoid Answer Selection. In *arXiv:1511.04108*.
[37] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. LSTM-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108* (2015).
[38] Yi Tay, Minh C Phan, Luu Anh Tuan, and Siu Cheung Hui. 2017. Learning to rank question answer pairs with holographic dual LSTM architecture. In *SIGIR*. ACM, 695–704.
[39] Rafael M Terol, Patricio Martínez-Barco, and Manuel Palomar. 2007. A knowledge based method for the medical question answering problem. *Computers in biology and medicine* 37, 10 (2007), 1511–1521.
[40] Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering. In *ACL-IJCNLP*, Vol. 2. 707–712.
[41] Mengqiu Wang and Christopher D Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *COLING*. Association for Computational Linguistics, 1164–1172.
[42] Shuohang Wang and Jing Jiang. 2016. A compare-aggregate model for matching text sequences. *arXiv preprint arXiv:1611.01747* (2016).
[43] Shuohang Wang and Jing Jiang. 2017. A Compare-Aggregate Model for Matching Text Sequences. In *ICLR*.
[44] Georg Wiese, Dirk Weissenborn, and Mariana Neves. 2017. Neural Domain Adaptation for Biomedical Question Answering. In *CoNLL*. Association for Computational Linguistics, 281–289.
[45] Xiaobing Xue, Jiwoon Jeon, and W Bruce Croft. 2008. Retrieval models for question and answer archives. In *SIGIR*. ACM, 475–482.
[46] Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, Maya Ramanath, Volker Tresp, and Gerhard Weikum. 2012. Natural language questions for the web of data. In *EMNLP-CoNLL*. Association for Computational Linguistics, 379–390.
[47] Liu Yang, Qingyao Ai, Jiafeng Guo, and W Bruce Croft. 2016. anmm: Ranking short answer texts with attention-based neural matching model. In *CIKM*. ACM, 287–296.
[48] Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *EMNLP*. 2013–2018.
[49] Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *NAACL-HLT*. 858–867.
[50] Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *ACL*, Vol. 1. 1744–1753.
[51] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. *arXiv preprint arXiv:1412.1632* (2014).
[52] Weiguo Zheng, Hong Cheng, Lei Zou, Jeffrey Xu Yu, and Kangfei Zhao. 2017. Natural Language Question/Answering: Let Users Talk With The Knowledge Graph. In *CIKM*. ACM, 217–226.
[53] Lei Zou, Ruizhe Huang, Haixun Wang, Jeffrey Xu Yu, Wenqiang He, and Dongyan Zhao. 2014. Natural language question answering over RDF: a graph data driven approach. In *SIGMOD*. ACM, 313–324.