How Context or Knowledge Can Benefit Healthcare Question Answering?

Xiaoli Wang, Feng Luo, Qingfeng Wu, and Zhifeng Bao

Abstract—Healthcare question answering (HQA) is a challenging task as questions are generally non-factoid in nature. Traditional information retrieval techniques do not perform well on non-factoid questions. Recent neural question answering systems are reported to have performance gains over traditional methods. However, little attention has been given to HQA as datasets are generally too small to train a neural model from scratch. Recently, several systems have been proposed to learn context representations for HQA. Despite moderate progress, these systems have not been thoroughly compared with state-of-the-art neural models, and these neural models were tested only on self-created datasets. This makes it difficult for practitioners to decide which models should be used for various scenarios. To address the above challenges, we develop a new joint model to incorporate both context and knowledge embeddings into neural ranking architectures. First, we adapt context embedding pre-trained from large open-domain corpus to small healthcare datasets. Second, we learn knowledge embedding from knowledge graphs to provide external information for understanding non-factoid questions. To evaluate how context or knowledge embedding can benefit HQA, we adapt many state-of-the-art methods for general QA to HQA, by injecting the context or knowledge information only, or both of them. Extensive experiments are conducted to compare our approach with those adapted methods and current HQA systems. The results show that our approach achieves the state-of-the-art performance on both HealthQA and NFCorpus datasets.

Index Terms—Healthcare Question Answering, Context Embedding, Knowledge Embedding

1 Introduction

The last decade has witnessed the flourishing of online QA platforms, which provide precise answers instead of navigating through full documents [1], [2]. Recently, domain-specific QA platforms in healthcare have emerged, which allow users to seek help online instead of looking for traditional clinical services [3]. It is a challenging task for users to navigate through massive information for querying the required healthcare answers. Early systems are based on traditional information retrieval (IR) methods [4]–[6], and a comprehensive survey can be found in [7]. However, they were reported to have limited success, as users post healthcare questions that are abstract and non-factoid in nature, in which IR methods do not perform well [3].

Recent neural QA systems are reported to have performance gains over traditional IR methods [2], [8]–[17]. The key idea is to learn language representations of questions and answers that are used as input to neural ranking architectures. Some methods such as [9], [10], [13], [17] learn the syntactic and semantic representations of the question and the answer separately, and then score each question-answer pair based on the similarity of their representations. Others such as [2], [8], [9], [11], [12], [15], [16] exploit attention mechanisms to learn the interaction information between the question and the answer, which can better focus on relevant parts of the question-answer pair. Despite the above success in general QA, little effort has been made on more specific domains such as HQA, as datasets are generally too small to train a deep learning system from scratch.

To address the challenges, several systems have adopted neural ranking architectures to the healthcare domain [1], [3], [18], [19]. Among them, context embeddings are reported to have the best performance [1], [18], [19]. However, the appropriateness of several state-of-the-art models has not been thoroughly evaluated, and these models are tested without considering the adaption of context embedding on existing neural networks. This makes it unclear for practitioners to decide which part of the models leads to performance improvement. To address the problem, we comprehensively study the applicability of state-of-the-art neural networks for general QA to HQA. An ablation study is also conducted on the effectiveness of context embedding on existing neural networks.

The above systems only capture lexical, syntactic, and context information, but ignore the external knowledge from knowledge graph (KG) that plays a crucial role in QA [20]. Several systems extract graph representations for the question and answer from KG using query languages such as SPARQL [21]–[24]. However, it is too time-consuming to process huge KGs [2]. To resolve this problem, recent systems incorporate pre-trained knowledge embedding into existing language representations [20], [25]–[27]. These systems have shown that injecting extra knowledge information can significantly enhance the results. This observation raises two interesting questions: (i) whether the abilities of external knowledge on the open domain are transferable to the domain of healthcare; (ii) whether incorporating ex-

- *X. Wang is with the School of Informatics and National Institute for Data Science in Health and Medicine, Xiamen University, China. E-mail: xlwang@xmu.edu.cn*
- *F. Luo and Q. Wu are with the School of Informatics, Xiamen University. E-mail: jessyluo02@gmail.com, qfwu@xmu.edu.cn*
- *Z. Bao is with RMIT University, Melbourne, Australia Email: zhifeng.bao@rmit.edu.au*
- *Corresponding author: Qingfeng Wu.*

Manuscript received December 25, 2020; revised April 08, 2021; accepted in June 03, 2021.

ternal knowledge embedding into existing neural networks benefits HQA.

This paper comprehensively studies state-of-the-art neural networks and their applicability to HQA. We propose a novel neural ranking framework, namely CK-HQA, with a new joint model to incorporate context and knowledge embeddings into neural ranking architectures for HQA. We also design systematic experiments for answering the following four questions: (1) How does context embedding improve existing neural networks for HQA? (2) How does knowledge embedding improve existing neural networks for HQA? (3) Does domain-specific knowledge embedding benefit HQA compared against open-domain knowledge embedding? (4) Does our joint model with context and knowledge embedding improve existing neural networks for HQA? Our main contributions are as follows:

- Different from existing HQA systems that use only context embedding or knowledge embedding, we propose a new neural ranking framework with a joint model that combines context and knowledge embeddings into neural ranking architectures (Section 4).
- We adapt many state-of-the-art neural networks for general QA to HQA, by injecting context or knowledge information only, or both of them. We study the applicability of each of such adapted models to HQA. Extensive experiments are conducted to compare our CK-HQA with those adapted models and current HQA systems. The results show that both context and knowledge embeddings can improve existing neural networks significantly, achieving state-of-the-art performance on two real-world datasets under four benchmark metrics (Section 5).
- We have implemented a benchmark system for users to carry out extensive experiments on self-collected datasets. The full system is publicly available at [28], together with codes for replicating the results shown in this paper. All existing and adapted state-of-the-art neural networks for HQA are implemented in a library for users to conveniently apply to their HQA tasks.

2 Related Work

Question answering is a classical problem in information retrieval. Early works are based on traditional information retrieval techniques (e.g., [5], [6]). However, such techniques do not perform well, when applied to healthcare question answering with very abstract and non-factoid questions [3], [29]. Recent years have witnessed many successes in applying neural networks to QA [30]. In this paper, we comprehensively study existing neural network architectures, including feature-based models [3], [29], [31], context-based models [12], [32] and knowledge-based models [20], [25]–[27], and their applicability to HQA.

2.1 Feature-based Models

Feature-based models use pre-trained word embeddings to capture syntactic and semantic information from texts for ranking. We present several representative models that have been applied to HQA [3], [29], [31].

CDSSM [13] is a latent semantic model that uses convolutional layers on word trigram features, and **PACRR** [33] is a position-aware neural model that adopts convolutional layers on n-gram features. Both **Arc-I** and **Arc-II** [9] use convolutional architectures to create word representations of query and document, and compute their relevance using a feed-forward network. Arc-I generates document-level representations, while Arc-II adopts word-level interaction features. **MatchPyramid** [8] and **aNMM** [16] use the dot product between query and document word vectors as their interaction features. The difference is that MatchPyramid employs convolutional layers to compute the relevance while aNMM uses an attention network. **MV-LSTM** [14] uses cosine or bilinear operation over Bi-LSTM features, to compute the interaction features. **DRMMTKS** [34] uses word count based interaction features between query and document words. Both **Conv-KNRM** [15] and **KNRM** [35] use kernel pooling on interaction features to compute similarity scores. **DUET** [11] adopts both word-level interaction features and document-level semantic features as input embeddings that are fed into convolutional layers to compute the relevance. **HAR** [3] combines word-level, sentence-level, and document-level representations, and uses cross attention mechanism to compute the relevance.

These models are reported to have impressive performance gains over traditional Information Retrieval models for HQA [3]. However, their power is restricted by only using lexical or syntactic information that ignores contextual information which might be useful for domain-specific QA.

2.2 Context-based Models

For context-based models, high-level sentence or document encoders that generate contextual token representations have been pre-trained from unlabeled data and fine-tuned in downstream tasks, such as ELMo [36] and BERT [32]. ELMo [36] learns the deep contextualized word representation from a bidirectional LSTM. BERT [32] utilizes a multi-layer bidirectional transformer encoder which can generate a deep bidirectional representation to capture more contextual information. Motivated by this, recent works employ the contextual representations to improve the performance of traditional feature-based models on QA [2], [12], [37]. For example, Comp-Clip [37] employs ELMo along with Latent-Clustering for QA. BAS [2] and CEDR [12] both use BERT. BAS combines BERT with an answer type detector, while CEDR employs BERT with existing neural architectures, such as PACRR [33], KNRM [35], and DRMMTKS [34].

2.3 Knowledge-based Models

Despite moderate progress, both the feature-based model and the context-based model neglect the incorporation of external knowledge [20], [27]. Most **knowledge-based models** are semantic parsing approaches, which can be divided into two stages: structured query generation and answer retrieval in knowledge base (KB) [38]–[43]. Some works [39], [40] aim to transform the natural language question into structured query. They propose a neural semantic parser to generate the logic query form via two subtasks (i.e., entity linking and logic form generation), and simply employ the traditional search algorithms such as BFS to obtain answers. Other works [38], [41]–[44] focus on neural retriever model to derive answers from KB. For example, [41]–[44] utilize

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3090253, IEEE Transactions on Knowledge and Data Engineering

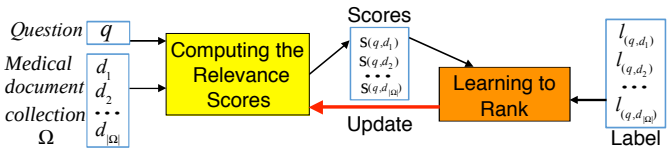


Fig. 1: Two procedures of CK-HQA

neural ranking model or logic reasoning framework to compute the relevance score between question and candidate answers.

However, these knowledge-based models are too time-consuming with huge knowledge base [2]. To resolve this problem, some knowledge-based models incorporate pre-trained knowledge embedding learned from the knowledge graph into existing language representations [20], [25]–[27]. These models have shown that injecting extra knowledge information from knowledge graph can significantly enhance the QA performance. Prompted by the recent success of contextualized and knowledge embeddings on language representation models, this paper considers incorporating external knowledge into contextualized language representation models to improve the HQA performance.

3 Problem Definition

We denote a healthcare question as q and an answer document as d. Given a question/document, we represent it as a token sequence of $\{T_1, \cdots, T_N\}$, whose length is N. The tokens are at the sub-word level. By aligning each token to an entity in knowledge graph, we have an entity sequence of $\{E_1, \cdots, E_{N'}\}$, whose length is N'. We formulate the healthcare question answering problem as a document ranking problem in Definition 1.

Definition 1. *Given a healthcare question q, let $\Omega = \{d_1, \cdots, d_{|\Omega|}\}$ be a set of candidate answer documents, and $|\Omega|$ is the total number of documents in Ω. We aim to compute the relevance score of $S(q, d_i)$ between q and each document d_i in Ω. $S(q, d_i) \in \Re$ represents a real-valued relevance score between q and d_i.*

To address the above problem, we develop a new neural ranking framework denoted as CK-HQA. Figure 1 shows two procedures in our framework: computing the relevance scores and learning to rank. In the first procedure, we calculate the relevance score of $S(q, d_i)$ between the question q and each document d_i. Then, the second procedure uses human-annotated labels for learning to optimize and update our ranking framework in the first procedure. We assume that the ground truth is known and labels are predefined. We use a triple of $(q, d_i, l_{(q,d_i)})$ to represent a ground truth of the relevance score between q and d_i, where $l_{(q,d_i)} \in \Re$ is the relevance score annotated by experts (We use several public datasets with human-annotated labels in our experiments in Section 5). The details of our proposed CK-HQA are shown in Section 4.

4 Methodology

Figure 2 illustrates the overall architecture of CK-HQA, which contains four main components: context encoder, knowledge encoder, joint model, and neural ranking model. Given a question, we aim to rank a set of candidate answer documents by computing the ranking score between each pair of question and document. Concretely, we first employ context encoder to learn the initial context embeddings of question and document, separately (Section 4.1). Then, we employ knowledge encoder to learn the knowledge embeddings of question and document, separately (Section 4.2). Afterwards, we introduce our proposed joint model to learn the final knowledge-enhanced context representations of question and document, separately (Section 4.3). Finally, the neural ranking model utilizes the knowledge-enhance context representations to calculate the final ranking score between question and document (Section 4.4).

4.1 Context Encoder

The context encoder aims to generate context embeddings of question and document. Recently, the pre-trained context language representation models, such as CoVe [45], ELMo [36], OpenAI GPT [46] and BERT [32], represent words in context by downstream tasks to capture higher-level context information such as disambiguation, syntactic structures, and semantic roles. Inspired by current success of BERT model (e.g., BioBERT [47] and SciBERT [48]), we adapt it in our context encoder to embed the question and document into three high-dimensional vector spaces.

Given a question consisting of N tokens denoted as $\{T_1^q, \cdots, T_N^q\}$ and a document consisting of M tokens denoted as $\{T_1^d, \cdots, T_M^d\}$. We first concatenate them into one token sequence. Then, we add $[CLS]$ token as the first token to the sequence and separate question and document sequence with $[SEP]$ token to get the input token sequence, i.e., $\{[CLS], T_1^q, \cdots, T_N^q, [SEP], T_1^d, \cdots, T_M^d\}$. After that, the sequence is fed into the context encoder. We use the BERT [32], a transformer model which is pre-trained with a large amount of corpus on sub-word level[1]. To generate the context representation for the input token sequence, we use the pooling of each attention layer. Suppose the total number of attention layers is L. In the i-th layer ($1 \leq i < L$), the output context embedding is denoted as

$$W^i = \{w^i_{[CLS]}, w^i_{T_1^q}, \cdots, w^i_{T_N^q}, w^i_{[SEP]}, w^i_{T_1^d}, \cdots, w^i_{T_M^d}\}.$$

Here, $W^i \in \mathbb{R}^{K \times D_c}$, where $K = N + M + 2$ and D_c is the attention hidden layer dimension. $w \in \mathbb{R}^{D_c}$ denotes a token representation. Then, we have

$$W^i = f(W^{i-1}),$$

where $f(\cdot)$ is the multi-head self-attention used in each layer.

Different from recent BERT models, we concentrate all the output context embeddings from L attention layers to generate the final context representation as below:

$$C = \begin{pmatrix} W^0 \\ W^1 \\ \vdots \\ W^{L-1} \end{pmatrix} \in \mathbb{R}^{L \times K \times D_c}. \quad (1)$$

1. In this paper, the corpus could be open-domain or domain-specific.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3090253, IEEE Transactions on Knowledge and Data Engineering

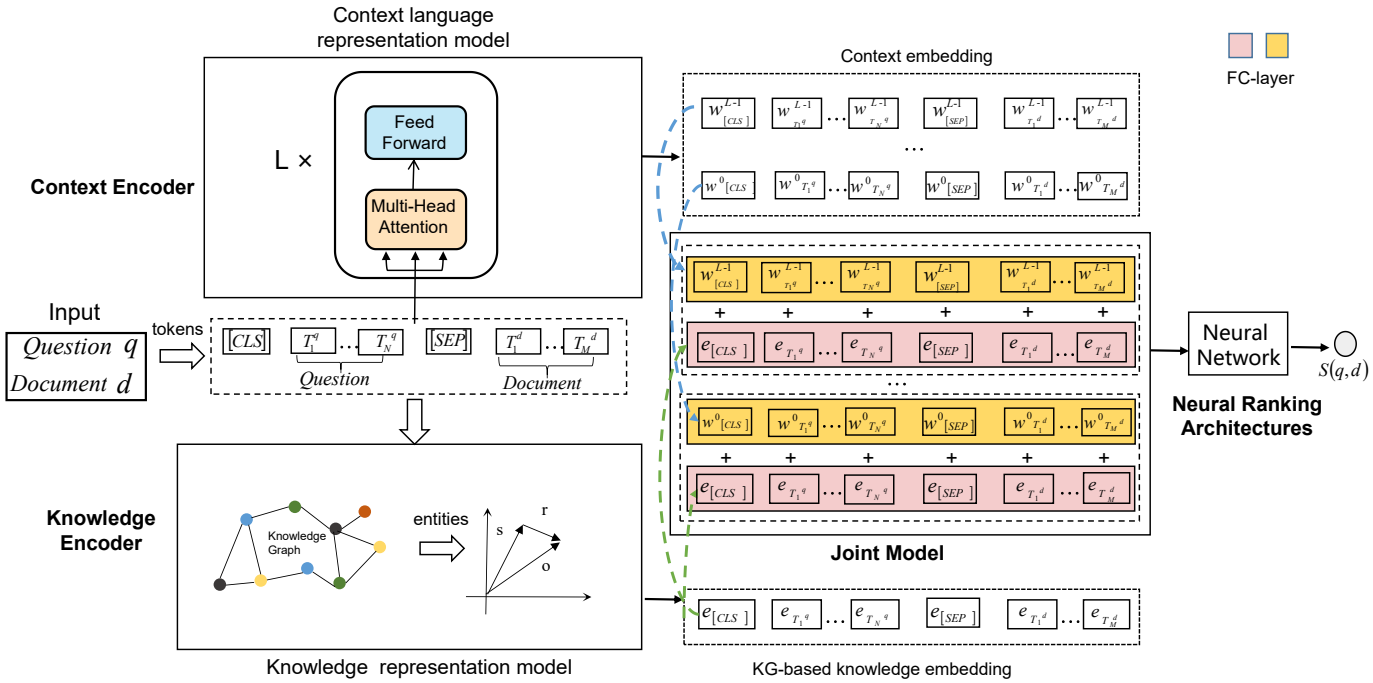


Fig. 2: Overview of our end-to-end CK-HQA framework. Given the question and document, the context encoder utilizes pre-trained context language representation model to generate the context embedding, while the knowledge encoder extracts entities from knowledge graph which are fed into knowledge representation model to generate the knowledge embedding. After getting the context embedding and knowledge embedding, the fusion layer in the joint model iterates L layers to combine context embeddings with knowledge embeddings to generate the knowledge-enhanced context representation for the neural ranking architectures.

From C, the inputs with question and document are embedded into three high-dimensional vector spaces. The embedding outputs of question and document are respectively concatenated to form a single representation: $CLR_q \in \mathbb{R}^{L\times N\times D_c}$ and $CLR_d \in \mathbb{R}^{L\times M\times D_c}$ respectively denote the context representation of question and document. Then, we have

$$CLR_q = \begin{pmatrix} w^0_{T^q_1} & \cdots & w^0_{T^q_N} \\ w^1_{T^q_1} & \cdots & w^1_{T^q_N} \\ \vdots & \ddots & \vdots \\ w^{L-1}_{T^q_1} & \cdots & w^{L-1}_{T^q_N} \end{pmatrix}, \quad (2)$$

$$CLR_d = \begin{pmatrix} w^0_{T^d_1} & \cdots & w^0_{T^d_M} \\ w^1_{T^d_1} & \cdots & w^1_{T^d_M} \\ \vdots & \ddots & \vdots \\ w^{L-1}_{T^d_1} & \cdots & w^{L-1}_{T^d_M} \end{pmatrix}. \quad (3)$$

Both representations combine the output of all layers to allow our contextualized language representation model to learn the importance of representation from different layers for better understanding the context information. The embedding of $[CLS]$ is also recorded in our framework and will be used as input in the ranking model (Section 4.4). It can also provide semantic information besides individual context information.

4.2 Knowledge Encoder

As shown in Figure 2, the knowledge encoder is designed to embed entities in KGs into a low-dimensional entity vector space by employing the knowledge representation model. To generate the entity sequences, existing works [20], [27], [49] incorporate the external knowledge information by performing entity mention detection using string matching methods [50] such as n-gram matching or entity linking methods. Following these works, given the question and document with their concatenated token sequence $\{[CLS], T^q_1, \cdots, T^q_N, [SEP], T^d_1, \cdots, T^d_M\}$, we first employ specific KG-related entity linking systems, such as Scispacy [51] or TagMe [52] for entity mention detection in our KGs (i.e., UMLS and Wikidata) as described in Section 5.1.4. Scispacy provides a fast and robust entity linker component to extract entities linked to UMLS. TagMe is a powerful tool that can conduct fast and accurate annotation of short texts with Wikidata entities. Due to the accuracy of entity linking methods on specific KGs, we align each token to a named entity by simply selecting the top-1 entity from the entity candidates of its corresponding entity mention. Note that not every token can be aligned to an entity. If a token does not have a corresponding entity, we align it with a pre-defined empty entity $[UNK]$. Consequently, the token sequence can be aligned to an entity sequence denoted as $\{[CLS], E^q_1, \cdots, E^q_N, [SEP], E^d_1, \cdots, E^d_M\}$, and our goal is to create a mapping of each entity into a low-dimensional entity vector space.

To learn the KG-based knowledge embedding, we uti-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3090253, IEEE Transactions on Knowledge and Data Engineering

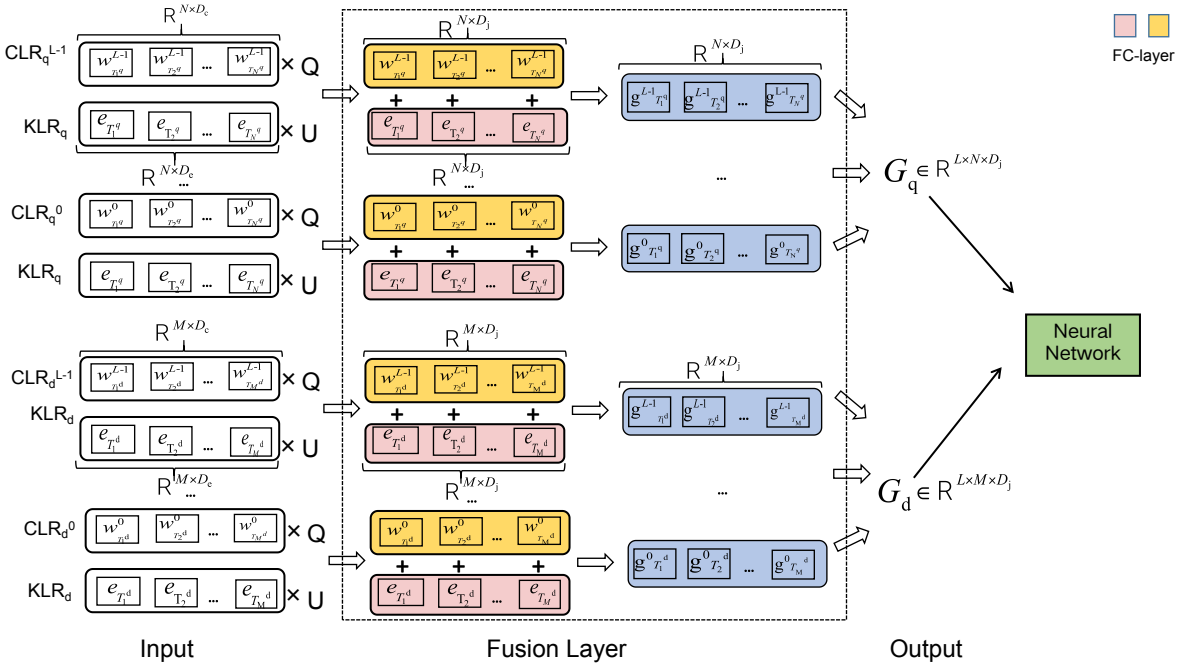


Fig. 3: Illustration of the fusion layer in our joint model. Given the context and knowledge embeddings, it iterates L layers context embedding with knowledge embedding to generate the knowledge-enhanced context representation.

lize the knowledge representation model to inject external knowledge information into knowledge representation. The KG is formed of a substantial number of triples, where each triple indicates a relation between two entities denoted as $(subject, relation, object)$. For each triple, the knowledge representation model learns vector embedding of it denoted as (s, r, o), where s is the embedding of the subject entity, r is the embedding of the relation and o is the embedding of the object entity. In this paper, we adopt the transE [53], which is an effective knowledge representation model, to pre-train the entity knowledge embedding. The idea is that the embedded entities s and o can be connected by r with low error, i.e., $s+r \approx o$. Thus, we can obtain the corresponding knowledge embedding for each entity in the entity sequence of question and document. To be specific, we encode the sequence of $\{[CLS], E_1^q, \cdots, E_N^q, [SEP], E_1^d, \cdots, E_M^d\}$ into the KG-based knowledge representation denoted as $\{[CLS], e_{T_1^q}, \cdots, e_{T_N^q}, [SEP], e_{T_1^d}, \cdots, e_{T_M^d}\}$, where $e \in \mathbb{R}^{D_e}$ and D_e is the dimension of entity embedding. The inputs with question and document are encoded into the entity vector space. The embedding outputs of question and document are respectively concatenated to form a single representation: $KLR_q \in \mathbb{R}^{N \times D_e}$ and $KLR_d \in \mathbb{R}^{M \times D_e}$ respectively denote the knowledge representation of question and document. N and M are the lengths of question and document entity sequence respectively. Then, we have

$$KLR_q = \{e_{T_1^q}, \cdots, e_{T_N^q}\},$$
$$KLR_d = \{e_{T_1^d}, \cdots, e_{T_M^d}\}.$$

4.3 Joint Model

A new joint model is proposed to incorporate both the context and knowledge embeddings into our language representation model. As shown in Figure 3, when obtaining the context and knowledge embeddings, the fusion layer in our joint model integrates them to generate the knowledge-enhanced context representation for each token and its corresponding entity. To aggregate both context and knowledge embeddings, we develop a new transformed strategy in the fusion layer denoted as CK-Trans. Algorithm 1 describes the procedure of CK-Trans. Given the input embeddings of CLR_q, CLR_d, KLR_q, and KLR_d, we first iterate the context embedding in each layer from the multi-layer context embeddings CLR_q and CLR_d. Then, in each layer, we use two transformation matrices $Q \in \mathbb{R}^{D_c \times D_j}$ and $U \in \mathbb{R}^{D_e \times D_j}$ with a bias vector $b \in \mathbb{R}^{D_j}$ to integrate the

Algorithm 1: CK-Trans

Input : context embedding CLR_q, CLR_d
knowledge embedding KLR_q , KLR_d
Output: knowledge-enhanced contexutalized representation G_q and G_d
Initialize transformation matrices Q, U and b;
Initialize the question representation list QL ;
Initialize the document representation list DL ;
for *i-th layer embedding* CLR_q^i *in* CLR_q **do**
  $G_q^i = \sigma(CLR_q^i * Q + KLR_q * U + b)$;
  add G_q^i to QL;
end
for *i-th layer embedding* CLR_d^i *in* CLR_d **do**
  $G_d^i = \sigma(CLR_d^i * Q + KLR_d * U + b)$;
  add G_d^i to DL;
end
stack QL to G_q;
stack DL to G_d;

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3090253, IEEE Transactions on Knowledge and Data Engineering

context embedding with knowledge embedding KLR_q and KLR_d for generating the knowledge-enhanced context representation, where D_j is the dimension of the knowledge-enhanced context representation. Finally, the representation in each layer is aggregated to generate the final output representation. As shown in Figure 3, the final output representations of question and document are $G_q \in \mathbb{R}^{L \times N \times D_j}$ and $G_d \in \mathbb{R}^{L \times M \times D_j}$.

4.4 Neural Ranking Architectures

Based on the final output of G_q and G_d aforementioned, we feed them into existing neural ranking models for HQA. Thus, the similarity representation can be represented as $s(q,d) \in \mathbb{R}^{L \times N \times M}$. Different structures are used to calculate the similarity representation and output the final ranking score as $S(q,d)$. In our implementation, we follow a most recent work in general QA [12] and apply three neural models of KNRM [35], PACRR [54], and DRMMTKS [34]. For each model, we concatenate the $[CLS]$ token embedding from Section 4.1 with the matching signals at the last hidden layer for benefiting HQA.

For training, the objective function of our framework follows most state-of-the-art neural ranking models [9], [14], [35], [55]. It uses the pairwise learning to rank and the loss is shown as follows:

$$Loss = \sum_{q} \sum_{d^+, d^-} \max(0, 1 - S(q, d^+) + S(q, d^-)),$$

where d^+ and d^- represent the pairwise relevant and irrelevant document respectively. Given the human-annotated ground truth $l_{(q,d^+)}$ and $l_{(q,d^-)}$ for d^+ and d^- respectively, we have $l_{(q,d^+)} > l_{(q,d^-)}$.

5 Evaluation

We design systematic experiments for answering the following four questions:

- How does context embedding improve existing neural networks for HQA?
- How does knowledge embedding improve existing neural networks for HQA?
- Does domain-specific knowledge embedding have benefits for HQA compared with open-domain knowledge embedding?
- Does the proposed joint model with context and knowledge embeddings together improve existing neural networks for HQA?

5.1 Setup

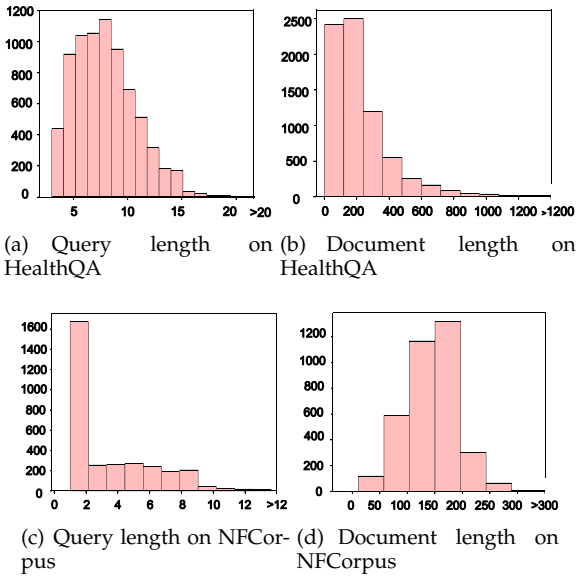
5.1.1 Datasets

We use two widely adopted datasets. Table 1 shows the statistics of documents and questions, and Figure 4 shows the distribution of the question and document length.

HealthQA [3] is a dataset used in recent work containing 7,517 questions and 7,355 documents. The documents are extracted from 1,235 healthcare articles from the Patient website[2]. Each section in the articles is selected as a document. Since the sections themselves are very long, most documents contain more than 100 words. Questions are human-created using the information in documents or rephrased from the document subtitles. We obtain this dataset from the authors of [3], and follow their settings to split all questions together with documents into the train, validation, and test datasets with a cardinality of 5,247, 1,109, and 1,134 respectively.

TABLE 1: Dataset statistics

Parameters	HealthQA	NFCorpus
Number of document	7,355	3,593
Number of question	7,517	3,220
Average length of question	8.0	3.6
Average length of document	233.4	146.1



(a) Query length on HealthQA (b) Document length on HealthQA

(c) Query length on NFCorpus (d) Document length on NFCorpus

Fig. 4: The distribution of question and document lengths on datasets

NFCorpus [56] is a dataset used in previous work that contains thousands of full-text medical queries with relevant links to research articles on PubMed[3]. The medical queries are collected from an HQA website[4]. The documents are titles and abstracts extracted from articles. Three relevance levels are defined based on direct and indirect links of queries to articles. We treat the label with direct and indirect links as relevant while the others as irrelevant. The dataset can be downloaded from a public link[5]. We follow the settings from the original paper to split all the questions together with documents into a training set of 80%, a validation set of 10%, and a test set of 10% respectively.

5.1.2 Evaluation metrics

Given a query $q_j \in Q$ where Q is a set of queries, we return the top-K documents to q_j according to the relevance score computed by each model. Let A_{q_j} be the set of relevant documents in the top-K retrieved results for q_j, and $\hat{A_{q_j}}$ be the set of human-annotated relevant documents to q_j in the dataset. Similar to [3], we adopt the following evaluation

2. https://patient.info/

3. http://www.ncbi.nlm.nih.gov/pubmed/
4. http://www.nutritionfacts.org/
5. http://www.nutritionfacts.org/

metrics [57]. For each metric, its value is averaged by all the test queries.

- $Precision@K$ ($P@K$) is calculated to evaluate the precision of the retrieved results, which denotes the ratio of relevant documents in the top-K results, i.e., $Precision@K = \frac{\sum_{j=1}^{|Q|} \frac{|A_{q_j}|}{K}}{|Q|}$.
- $Recall@K$ ($R@K$) is computed as the ratio of relevant documents in the top-K results over human-annotated relevant documents in the dataset, i.e., $Recall@K = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{|A_{q_j}|}{|\hat{A_{q_j}}|}$.
- MAP is defined as the mean value of average precision for relevant documents to given queries. $MAP = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{\sum_{i=1}^{|A_{q_j}|} \frac{i}{rank(d_i)}}{|\hat{A_{q_j}}|}$, where $rank(d_i)$ is the ranking position of document d_i.
- $NDCG@K$ ($N@K$) is the normalized discounted cumulative gain, which measures the quality of the ranking order of retrieved results [57].

5.1.3 Competitors

Our competitors come from two bodies of work: (i) methods specifically designed for HQA; (ii) we adapt state-of-the-art general QA methods to the HQA domain.

HAR [3] is a recent neural model proposed for HQA. It employs a Bi-GRU encoder and a cross-attention layer to capture the similarity information between query and document at word-level. It also uses a hierarchical inner-attention mechanism over the document words and sentences to find interaction information for the query.

Neural Models for Text Retrieval – aNMM [16], **CDSSM** [13], **Arc-I** [9], **Arc-II** [9], **DUET** [11], **MV-LSTM** [14], **CONV-KNRM** [15] and **MatchPyramid** [8]. Most of them are also chosen as the baselines in recent work such as HQA [3].

DRMMTKS [34], **KNRM** [35] and **PACRR** [33] are three models we choose to adapt by injecting context or knowledge information only, or both of them. We study their applicabilities to HQA in Section 4.4.

5.1.4 Incorporating context and knowledge

To incorporate the context information into our models, we pre-train context embeddings. In our experiments, we use two pre-trained BERT models, BERT (Base, Uncased) [32] and SciBERT(Uncased) [48]. BERT (Base, Uncased) is trained on lower-based English texts, while SciBERT (Uncased) is trained on domain-specific scientific texts. The intuition is that the domain-specific context embedding may be helpful for HQA tasks.

To incorporate knowledge into our models, we recognize all the corresponding entities and pre-train the knowledge embeddings. Two knowledge graphs are used in our experiments: Wikidata [6] and UMLS [7]. Wikidata is open-domain containing 5,040,986 entities and 24,267,796 fact triples, while UMLS is domain-specific containing 4,258,810 entities and 11,882,429 fact triples. After recognizing all the entities from knowledge graphs, we utilize TransE [53] to learn their knowledge embeddings.

6. https://www.wikidata.org/
7. https://www.nlm.nih.gov/research/umls/

To evaluate our framework, we implement various models by incorporating either context or knowledge embeddings only, or both of them. We divide the models to be evaluated (in Tables 2 and 3) into four groups:

- **Original models**: aNMM, CDSSM, Arc-I, CONV-KNRM, Arc-II, MatchPyramid, DUET, MV-LSTM, HAR, DRMMTKS, KNRM, and PACRR.
- **Adapted knowledge-embedding models**: DRMMTKS-W, KNRM-W, and PACRR-W.
- **Adapted context-embedding models**: KNRM-B, KNRM-BC, DRMMTKS-B, DRMMTKS-BC, PACRR-B, and PACRR-BC.
- **Our CK-HQA (knowledge-enhanced context embedding models)**: DRMMTKS-BC-W, DRMMTKS-BC-U, DRMMTKS-SC-W, DRMMTKS-SC-U, KNRM-BC-W, KNRM-BC-U, KNRM-SC-W, KNRM-SC-U, PACRR-BC-W, PACRR-BC-U, PACRR-SC-W, and PACRR-SC-U.

In our model names, B refers to context embedding using BERT, and BC further takes [CLS] token representation into consideration when employing BERT. SC refers to domain-specific context embedding using SciBERT [48]. W denotes knowledge embedding from an open-domain KG named Wikidata; while U denotes that from domain-specific KG named UMLS.

5.1.5 Parameter settings

All methods are trained using Adam optimizer [58], with an initial learning rate of 0.001. The BERT layers are trained with a rate of $2e^{-5}$. All the methods are trained for 100 epochs, each with 32 batches of 16 training pairs following [12]. For HealthQA dataset, we set the maximum number of tokens in each query to 15 and in each document to 300 by following the previous work [3]. For the NFCorpus dataset, the maximum number of tokens is set to 10 in queries and 400 in documents.

We implement all competitors in Keras, with TensorFlow as the backend. The parameters are set the same as those used in [3]. For adapted models, the parameters are set the same as our implemented models in the CK-HQA framework for fair consideration to evaluate the impact of context embedding, knowledge embedding, and our knowledge-enhanced context embedding. We have implemented an interaction system [28] to help users design and evaluate HQA neutral ranking models for their future applications.

5.2 Experimental Results

The results on HealthQA and NFCorpus are summarized in Table 2 and Table 3. Researchers can reproduce the results following the source codes at https://github.com/emmali808/HQADeepHelper. For the NFCorpus dataset, some questions have no relevant documents, hence the metric Recall@K is not included. All the models labeled with CK can be considered as one of our CK-HQA based models. There are four parts in each table: in the first part we compare our CK-HQA models with state-of-the-art competitors; from the second to fourth part, we further study the effect of different embedding variants. Among our CK-HQA models, we explore the domain-specific context and knowledge graph.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3090253, IEEE Transactions on Knowledge and Data Engineering

TABLE 2: Ranking performance on HealthQA. In the Cat column (i.e., Category), F denotes original models, C denotes adapted models with context embedding, K denotes adapted models with knowledge embedding, CK denotes jointed models with context and knowledge embeddings together. The number in the parenthesis indicates the improvements compared with the first model in each line. The best results are indicated in bold.

Cat	Model	MAP	P@K		Recall@k		NDCG@k	
			P@3	P@5	R@3	R@5	N@3	N@5
F	ARC-I [9]	30.62	10.49	10.19	31.48	50.97	22.88	30.85
	CDSSM [13]	53.25 (73.9%)	22.08 (110.4%)	16.74 (64.2%)	66.23 (110.4%)	83.69 (64.2%)	51.9 (126.9%)	59.09 (91.5%)
	ARC-II [9]	59.55 (94.5%)	24.37 (132.2%)	17.65 (73.2%)	73.1 (132.2%)	88.27 (73.2%)	59.23 (158.9%)	65.5 (112.3%)
	MV-LSTM [14]	64.85 (111.8%)	30.92 (194.7%)	19.59 (92.2%)	92.77 (194.7%)	97.97 (92.2%)	80.32 (251.1%)	82.5 (167.4%)
	MatchPyramid [8]	72.24 (135.9%)	29.25 (178.7%)	19.58 (92%)	87.74 (178.7%)	97.88 (92%)	74.06 (223.8%)	78.35 (154%)
	aNMM [16]	75.45 (146.4%)	30.28 (188.5%)	19.68 (93.1%)	90.83 (188.5%)	98.41 (93.1%)	77.87 (240.4%)	81.07 (162.8%)
	DUET [11]	76.02 (148.3%)	30.1 (186.8%)	19.38 (90.1%)	90.3 (186.8%)	96.91 (90.1%)	78.22 (241.9%)	80.95 (162.4%)
	HAR [3]	86.2 (181.5%)	31.72 (202.2%)	19.75 (93.8%)	95.15 (202.2%)	98.77 (93.8%)	87.74 (283.5%)	89.25 (189.3%)
	CONV-KNRM [15]	**86.94 (183.9%)**	**32.33 (208.1%)**	**19.89 (95.2%)**	**97 (208.1%)**	**99.47 (95.2%)**	**89.03 (289.2%)**	**90.07 (192%)**
F	PACRR [54]	71.6	28.81	19.17	86.42	95.86	73.36	77.30
K	PACRR-W	80.85 (12.9%)	31.16 (8.2%)	19.63 (2.4%)	93.47 (8.2%)	98.15 (2.4%)	83.09 (13.3%)	85.05 (10%)
C	PACRR-B	91.4 (27.7%)	32.63 (13.3%)	19.89 (3.8%)	97.88 (13.3%)	99.47 (3.8%)	92.74 (26.4%)	93.41 (20.8%)
	PACRR-BC	91.58 (27.9%)	32.66 (13.4%)	19.88 (3.7%)	97.97 (13.4%)	99.38 (3.7%)	92.92 (26.7%)	93.51 (21%)
CK	PACRR-BC-W	91.68 (28%)	32.83 (14%)	19.91 (3.9%)	98.5 (14%)	99.56 (3.9%)	93.22 (27.1%)	93.66 (21.2%)
	PACRR-BC-U	92.56 (29.3%)	32.77 (13.7%)	19.96 (4.1%)	98.32 (13.8%)	99.82 (4.1%)	93.76 (27.8%)	94.4 (22.1%)
	PACRR-SC-W	92.93 (29.8%)	33.07 (14.8%)	19.98 (4.2%)	99.21 (14.8%)	99.91 (4.2%)	94.41 (28.7%)	94.72 (22.5%)
	PACRR-SC-U	**94.11 (31.4%)**	**33.13 (15%)**	**20 (4.3%)**	**99.38 (15%)**	**100 (4.3%)**	**95.36 (30%)**	**95.62 (23.7%)**
F	KNRM [35]	75.29	29.63	19.37	88.89	96.83	77.05	80.36
K	KNRM-W	80.42 (6.8%)	31.28 (5.6%)	19.84 (2.5%)	93.83 (5.6%)	99.21 (2.5%)	82.81 (7.5%)	85.08 (5.9%)
C	KNRM-B	87.17 (15.8%)	31.95 (7.8%)	19.82 (2.3%)	95.86 (7.8%)	99.12 (2.4%)	88.73 (15.2%)	90.11 (12.1%)
	KNRM-BC	91.43 (21.4%)	32.69 (10.3%)	19.96 (3.1%)	98.06 (10.3%)	99.82 (3.1%)	92.81 (20.5%)	93.55 (16.4%)
CK	KNRM-BC-U	89.15 (18.4%)	**33.1 (11.7%)**	19.93 (2.9%)	**99.29 (11.7%)**	99.91 (3.2%)	**95.47 (23.9%)**	**95.73 (19.1%)**
	KNRM-SC-U	92.61 (23%)	32.98 (11.3%)	**20 (3.3%)**	98.94 (11.3%)	**100 (3.3%)**	94.07 (22.1%)	94.51 (17.6%)
	KNRM-BC-W	92.72 (23.2%)	32.92 (11.1%)	19.91 (2.8%)	98.77 (11.1%)	99.56 (2.8%)	94.11 (22.1%)	94.44 (17.5%)
	KNRM-SC-W	**93.14 (23.7%)**	**33.1 (11.7%)**	**20 (3.3%)**	**99.29 (11.7%)**	**100 (3.3%)**	94.61 (22.8%)	94.9 (18.1%)
F	DRMMTKS [34]	76.76	30.10	19.65	90.30	98.24	78.68	82.01
K	DRMMTKS-W	77.62 (1.1%)	30.19 (0.3%)	19.51 (-0.7%)	90.56 (0.3%)	97.53 (-0.7%)	79.44 (1%)	82.37 (0.4%)
C	DRMMTKS-B	92.14 (20%)	32.86 (9.2%)	19.93 (1.4%)	98.59 (9.2%)	99.65 (1.4%)	93.57 (18.9%)	94.02 (14.6%)
	DRMMTKS-BC	94.21 (22.7%)	33.1 (10%)	**20 (1.8%)**	99.29 (10%)	**100 (1.8%)**	95.4 (21.3%)	95.69 (16.7%)
CK	DRMMTKS-SC-U	94.11 (22.6%)	33.13 (10.1%)	**20 (1.8%)**	99.38 (10.1%)	**100 (1.8%)**	95.36 (21.2%)	95.62 (16.6%)
	DRMMTKS-BC-U	94.13 (22.6%)	33.1 (10%)	19.98 (1.7%)	99.29 (10%)	99.91 (1.7%)	95.35 (21.2%)	95.61 (16.6%)
	DRMMTKS-SC-W	94.25 (22.8%)	**33.19 (10.3%)**	**20 (1.8%)**	**99.56 (10.3%)**	**100 (1.8%)**	**95.54 (21.4%)**	**95.73 (16.7%)**
	DRMMTKS-BC-W	**94.3 (22.9%)**	33.1 (10%)	19.98 (1.7%)	99.29 (10%)	99.91 (1.7%)	95.47 (21.3%)	**95.73 (16.7%)**

5.2.1 Our CK-HQA vs. competitors

We have the following observations. (1) Among the competitors, CONV-KNRM and aNMM perform the best on HealthQA and NFCorpus respectively. (2) Our CK-HQA can substantially outperform all competitors on all metrics. For example on the HealthQA, the best MAP result of CK-HQA is 94.30% (i.e., DRMMTKS-BC-W), beating CONV-KNRM by 8.5%; on NFCorpus, the best MAP result for CK-HQA (i.e., DRMMTKS-SC-U) outperforms aNMM by 6.8%. Such an increase is attributed to the superiority of our CK-HQA framework over existing models. Using traditional word embedding such as GloVe, existing models are limited by the power of language representation and fail to capture the deep context information. However, our framework can provide the bi-direction context information and knowledge, which are crucial for HQA. Other observations are as follows:

- ARC-I and CDSSM do not work well for the HQA task. This is because they generate the representation of query and document separately and simply compute the relevance score using feedforward network or cosine similarity. Although the representation provides the semantic information for query and document, it fails to capture the important matching information between query and document.
- ARC-II has low performance, probably because it computes similarity matrix in an early stage. Therefore, it only captures the matching information at word-level but loses the structural information such as phrases and sentences.
- CONV-KNRM and aNMM outperform other baselines for both capturing fine-grained matching information based on the matching matrix, using kernel-pooling layer or value-shared weighting schema. With more

TABLE 3: Ranking performance on NFCorpus. The number in the parenthesis indicates the improvements compared with the first model in each line. The best results are indicated in bold.

Cat	Model	MAP	P@K			NDCG@K		
			P@1	P@3	P@5	N@1	N@3	N@5
F	ARC-I [9]	15.69	13	10.22	9.29	13	13.16	14.37
	ARC-II [9]	15.98 (1.8%)	13.93 (7.1%)	11.15 (9.1%)	9.54 (2.7%)	13.93 (7.1%)	14.24 (8.2%)	14.96 (4.1%)
	DUET [11]	16.27 (3.7%)	12.38 (-4.8%)	10.11 (-1%)	10.09 (8.7%)	12.38 (-4.8%)	13.14 (-0.2%)	14.89 (3.6%)
	CDSSM [13]	17.06 (8.7%)	15.79 (21.4%)	11.87 (16.2%)	10.03 (8%)	15.79 (21.4%)	15.33 (16.5%)	15.92 (10.8%)
	MV-LSTM [14]	17.57 (12%)	15.17 (16.7%)	11.35 (11.1%)	10.22 (10%)	15.17 (16.7%)	15.41 (17.1%)	16.54 (15.1%)
	HAR [3]	19.76 (25.9%)	17.03 (31%)	13.31 (30.3%)	11.58 (24.7%)	17.03 (31%)	18.08 (37.4%)	19.17 (33.4%)
	CONV-KNRM [15]	21.93 (39.8%)	21.98 (69%)	16.41 (60.6%)	13.31 (43.3%)	21.98 (69%)	21.79 (65.6%)	22.07 (53.6%)
	MatchPyramid [8]	22.02 (40.3%)	21.67 (66.7%)	17.23 (68.7%)	14.30 (54%)	21.67 (66.7%)	22.24 (69%)	22.76 (58.4%)
	aNMM [16]	**23.5 (49.8%)**	**24.15 (85.7%)**	**17.96 (75.8%)**	**14.74 (58.7%)**	**24.15 (85.7%)**	**24.16 (83.6%)**	**24.43 (70%)**
F	PACRR [54]	22.05	22.6	17.03	13.93	22.6	22.57	22.93
K	PACRR-W	22.22 (0.8%)	21.36 (-5.5%)	16.2 (-4.9%)	13.44 (-3.5%)	21.36 (-5.5%)	22.12 (-2%)	22.76 (-0.7%)
C	PACRR-B	23.48 (6.5%)	23.53 (4.1%)	18.16 (6.6%)	15.05 (8%)	23.53 (4.1%)	24.09 (6.7%)	24.62 (7.4%)
	PACRR-BC	23.81 (8%)	23.84 (5.5%)	18.06 (6%)	**15.29 (9.8%)**	23.84 (5.5%)	24.16 (7%)	24.86 (8.4%)
CK	PACRR-SC-W	23.56 (6.8%)	22.29 (-1.4%)	18.06 (6%)	14.98 (7.5%)	22.29 (-1.4%)	23.52 (4.2%)	24.23 (5.7%)
	PACRR-BC-W	23.77 (7.8%)	**25.39 (12.3%)**	17.85 (4.8%)	14.55 (4.5%)	**25.39 (12.3%)**	24.21 (7.3%)	24.57 (7.2%)
	PACRR-BC-U	23.98 (8.8%)	24.46 (8.2%)	18.68 (9.7%)	14.98 (7.5%)	24.46 (8.2%)	24.68 (9.3%)	24.9 (8.6%)
	PACRR-SC-U	**24.67 (11.9%)**	24.77 (9.6%)	**18.78 (10.3%)**	15.17 (8.9%)	24.77 (9.6%)	**24.99 (10.7%)**	**25.32 (10.4%)**
F	KNRM [35]	20.64	20.43	15.38	13.25	20.43	20.42	21.43
K	KNRM-W	22.4 (8.5%)	23.22 (13.6%)	16.82 (9.4%)	13.31 (0.4%)	23.22 (13.6%)	22.9 (12.1%)	22.79 (6.3%)
C	KNRM-B	23.7 (14.8%)	25.08 (22.7%)	17.85 (16.1%)	14.67 (10.7%)	25.08 (22.7%)	24.23 (18.6%)	24.57 (14.6%)
	KNRM-BC	23.9 (15.8%)	24.15 (18.2%)	18.16 (18.1%)	15.05 (13.6%)	24.15 (18.2%)	24.47 (19.8%)	24.86 (16%)
CK	KNRM-BC-W	23.42 (13.4%)	24.15 (18.2%)	17.23 (12.1%)	14.37 (8.4%)	24.15 (18.2%)	23.32 (14.2%)	24.09 (12.4%)
	KNRM-SC-W	24.35 (17.9%)	24.15 (18.2%)	17.75 (15.4%)	14.43 (8.9%)	24.15 (18.2%)	24.14 (18.2%)	24.58 (14.7%)
	KNRM-BC-U	24.49 (18.6%)	**26.01 (27.3%)**	**18.47 (20.1%)**	14.8 (11.7%)	**26.01 (27.3%)**	**25.16 (23.2%)**	25.25 (17.8%)
	KNRM-SC-U	**24.78 (20%)**	25.39 (24.3%)	18.27 (18.8%)	**15.29 (15.4%)**	25.39 (24.3%)	24.69 (20.9%)	**25.37 (18.4%)**
F	DRMMTKS [34]	23.59	24.46	17.96	14.55	24.46	24.19	24.6
K	DRMMTKS-W	21.74 (-7.9%)	21.67 (-11.4%)	16.31 (-9.2%)	13.13 (-9.8%)	21.67 (-11.4%)	21.91 (-9.4%)	22.13 (-10%)
C	DRMMTKS-B	23.82 (1%)	24.77 (1.3%)	18.27 (1.7%)	14.55 (0%)	24.77 (1.3%)	24.3 (0.5%)	24.65 (0.2%)
	DRMMTKS-BC	24.34 (3.2%)	24.46 (0%)	18.58 (3.5%)	15.36 (5.6%)	24.46 (0%)	**24.97 (3.2%)**	25.42 (3.3%)
CK	DRMMTKS-BC-U	23.96 (1.6%)	24.46 (0%)	18.47 (2.9%)	**15.54 (6.8%)**	24.46 (0%)	24.42 (1%)	25.3 (2.8%)
	DRMMTKS-BC-W	24.59 (4.2%)	25.7 (5.1%)	18.68 (4%)	14.98 (2.9%)	25.7 (5.1%)	24.91 (3%)	25.18 (2.4%)
	DRMMTKS-SC-W	24.65 (4.5%)	24.46 (0%)	**18.78 (4.6%)**	15.42 (6%)	24.46 (0%)	24.77 (2.4%)	25.39 (3.2%)
	DRMMTKS-SC-U	**25.1 (6.4%)**	**26.01 (6.3%)**	18.47 (2.9%)	15.36 (5.6%)	**26.01 (6.3%)**	24.9 (3%)	**25.62 (4.1%)**

fine-grained matching info, the relevance score could be more precise.

5.2.2 Effect of embedding variants

To demonstrate the efficacy of our CK-HQA framework, we compare the following three embedding variants: context embedding (the adapted models affixed by "-B" and "-BC"), knowledge embedding (the adapted models affixed by "-W"), and knowledge-enhanced context embedding (the models affixed by "-BC-W", "-SC-W", "-BC-U" and "-SC-U") that incorporate both context and knowledge embeddings. We observe that:

- With knowledge embedding, all adapted models achieve better performance on HealthQA. For instance, the increase of $NDCG@3$ can be up to 13.3%. On NFCorpus, some unexpected results happened on PACRR-W and DRMMTKS-W. This is caused by queries that are very short without mapping entities in KGs. No external knowledge can be captured for these queries. While for long documents, external knowledge is captured, which results in more noises are introduced for each query word. This is harmful to PACRR and DRMMTKS, which capture the strongest matching signals with query dimension and are less adaptable to noises.
- The context embedding significantly boosts the performance. For instance, $P@3$ increases by 13.3% for PACRR-B on HealthQA and 16.1% for KNRM-B on NFCorpus.
- With context embedding, we achieve better performance by adding the [CLS] token. The $P@3$ of KNRM-B increases from 31.95% to 32.69% for KNRM-BC on HealthQA and from 17.85% to 18.16% on NFCorpus.
- When both context and knowledge embeddings are incorporated, the performance has a significant increase. For example, the MAP result of PACRR-BC-U increases by 29.3% on HealthQA, more compared with PACRR-

BC (i.e., 27.9%) and PACRR-W (i.e., 12.9%). This finding confirms the efficacy of our joint model, which incorporates context embeddings with knowledge embeddings.
- The models with the context embedding perform much better than the models with knowledge embedding. For example, the MAP of PACRR-W increases by 12.9%, while 27.9% of PACRR-BC on HealthQA.

5.2.3 Effect of domain-specific context and knowledge

To analyze the effect of domain-specific knowledge graphs, we follow Section 5.1.4 and choose domain-specific SciBERT along with BERT as the context language representation models and the medical knowledge graph UMLS and open-domain knowledge graph Wikidata in the knowledge representation models. The results are reported as the CK category in Tables 2 and 3. In most cases, SciBERT+UMLS outperforms other associations (i.e., PACRR-SC-U, KNRM-SC-U, DRMMTKS-SC-U in NFCorpus and PACRR-SC-U in HealthQA). The result is unsurprising as SciBERT provides the bio-medical context information, while UMLS further introduces the medical knowledge beyond the context. This shows that the combination of domain-specific context language representation and domain-specific knowledge representation does contribute to HQA. However, simply using the domain-specific context language representation or domain-specific knowledge representation does not always perform better. It could be attributed to the context or knowledge information. Sometimes the knowledge information may contribute more, therefore the introduction of domain-specific context information seems to have less effect and vice versa.

5.2.4 Effect of the Question Length on NFCorpus

To investigate the poor performance on NFCorpus, we study the performance w.r.t. question lengths over NFCorpus. Figure 5 presents the results on three evaluation metrics of MAP, $NDCG@3$ and $Precision@3$. We evaluate the performance by comparing DRMMTKS, DRMMTKS-W, DRMMTKS-BC and DRMMTKS-BC-W. From the results, we observe that there is a strong gap for different question lengths. All the models perform poorly for single-word questions, but incur a significant improvement when question lengths become longer. For instance, the MAP result is far less than 10% for single-word questions, but more than 50% for questions which contain more than 5 words. This is because the shorter questions fail to provide enough information for retrieving relevant answers, while for longer questions more context and knowledge information can be utilized. However, a high proportion of questions in NFCorpus only contains one single word, which leads to the poor performance in Table 3.

5.3 Efficiency

Table 4 shows the time cost and MAP result based on DRMMTKS ranking architecture with different embeddings. The training time is the time cost of model training and the query time is the time for answering a query. We choose DRMMTKS as the baseline and compare it with other models both on time cost and MAP performance. Incorporated with both the context embedding and knowledge embedding, DRMMTKS-BC-W, DRMMTKS-BC-U, DRMMTKS-SC-W, and DRMMTKS-SC-U achieve the best performance, at the expense of off-line training and query time. For instance, the MAP result increases by 22% with almost 20 times training time cost and 40 times query time cost on HQA. We also find that the context embedding takes much more time than the knowledge embedding. For example, the training time is only 4 times with knowledge embedding while 16 times with context embedding on HealthQA. This is because pre-training deep context embedding is more complex. However, context embedding is more powerful than knowledge embedding. In HealthQA, the MAP result gets 20.04% absolute rise with context embedding in DRMMTKS-B while gets 1.13% improvement with knowledge embedding in DRMMTKS-W.

5.4 Case Study

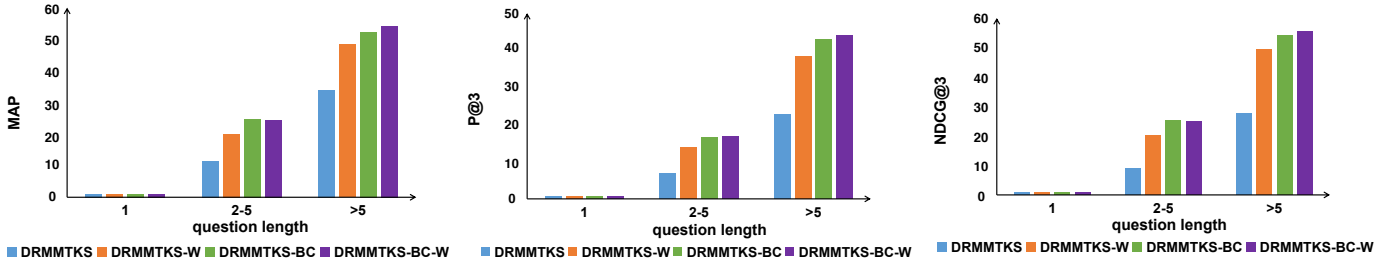
DRMMTKS provides an intuitive way to inspect the soft-alignment between the question and answers by visualizing the attention weight. The attention weight comes from the interaction matrix in DRMMTKS [34]. We finally visualize the attention weight with a relevant question-answer pair from HealthQA in Figure 6. The depth of colors indicates the relevance degree of token pair. The darker the color is, the more relevant the token pair is.

We summarize the observations as below. First, DRMMTKS performs the worst among the four models and fails to highlight relevant token pairs. Second, by incorporating the knowledge embedding, DRMMTKS-W can find the relatedness between word "cancer" and "cancerous" that DRMMTKS ignores. Third, by incorporating the context embedding, DRMMTKS-BC highlights more token pairs and shows a stronger relevance between the question and answer. Finally, the relevant token pairs found in Figure 6(d) are fewer than those in Figure 6(c). This is because DRMMTKS-BC-W pays much attention to the relevant token pairs which have a strong correlation in KG. For example, "bone cancer" is highly connected to "mali ##gant" (tokens for "malignant") and "tu ##mour" (tokens for "tumour") in KG, while the token sequences such as "what causes" and "starts from" seem to have no external knowledge information in KG.

5.5 Qualitative Results

Figure 7 shows an example of a question and the retrieved top-3 answers, respectively returned by DRMMTKS, DRMMTKS-W, DRMMTKS-BC and DRMMTKS-BC-W. The results show the impact of knowledge embedding, context embedding, and our knowledge-enhanced context embedding. We observe that except for DRMMTKS, other three methods DRMMTKS-W, DRMMTKS-BC, and DRMMTKS-BC-W all return relevant answers to the question (The test question has only one relevant answer in the human-annotated dataset). DRMMTKS-BC and DRMMTKS-BC-W both rank the correct answer as the most relevant (i.e., TOP-1), while DRMMTKS-W ranks the correct answer as the third relevant answer in the top-3 results. This is caused by the non-factoid nature of the test question. DRMMTKS simply utilizes traditional features, hence it may focus more

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3090253, IEEE Transactions on Knowledge and Data Engineering



(a) Map by question length (b) P@3 by question length (c) N@3 by question length

Fig. 5: Performance by question length on the NFCorpus dataset

TABLE 4: Time cost and evaluation results on DRMMTKS ranking model with different embeddings. The TT (training time) is measured in minutes and the QT (query time) is measured in milliseconds. For MAP, the number in the parenthesis indicates the improvements compared with DRMMTKS.

Cat	Model	HQA			NFCorpus		
		TT	QT	MAP	TT	QT	MAP
F	DRMMTKS	6.38	0.48	76.76	6.60	0.48	23.59
K	DRMMTKS-W	25.4	3.02	77.62 (1.13%)	25.14	2.14	21.74 (-7.85%)
C	DRMMTKS-B	107.86	16.46	92.14 (20.04%)	108.88	21.52	23.82 (0.96%)
	DRMMTKS-BC	98.92	16.51	94.21 (22.74%)	98.48	16.55	24.34 (3.17%)
CK	DRMMTKS-BC-W	129.99	19.42	94.3 (22.86%)	116.83	19.49	24.59 (4.23%)
	DRMMTKS-BC-U	122.98	19.84	94.13 (22.64%)	117.44	19.65	23.96 (1.56%)
	DRMMTKS-SC-W	125.64	24.81	94.25 (22.79%)	125.81	19.70	24.65 (4.48%)
	DRMMTKS-SC-U	116.89	19.47	94.11 (22.61%)	124.11	19.65	25.10 (6.39%)

on the keywords in the query and fail to discriminate similar answers whose topic is related to the keywords. With knowledge embedding, DRMMTKS-W can incorporate external knowledge to capture the relevant information between the question and answers, which can help understand the non-factoid question. With context embedding, DRMMTKS-BC and DRMMTKS-BC-W utilize the deep context information to capture the matching information between the question and answers, which is crucial for HQA.

6 Conclusions and Future Work

In this paper, we comprehensively studied state-of-the-art neural network models and their applicability to HQA. Then we proposed a new neural ranking framework with a joint model, which combines context and knowledge embeddings into existing neural ranking architectures for HQA. The experimental results show that our joint model achieves the best performance against state-of-the-art HQA systems. We also provide new insights about domain-specific context and knowledge implementation in our framework and the trade-off between performance and time cost. Researchers can select appropriate models for various HQA scenarios based on our findings.

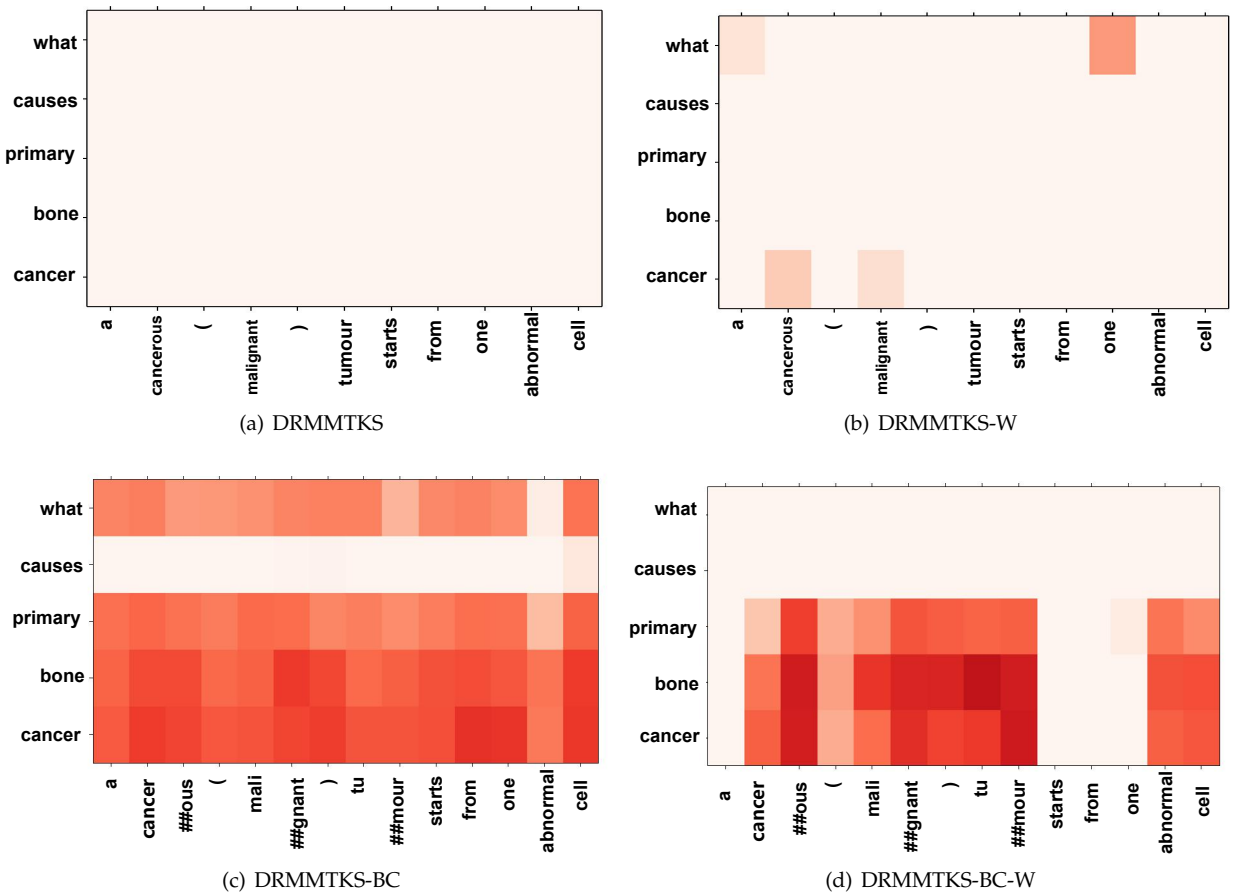
We summarize several HQA model design guidelines for future research: (I) interaction feature extraction between the query and answer can help improve the performance for long documents, but degrade the performance for short questions; (II) incorporating external knowledge into existing neural networks can enhance the HQA performance, but domain-specific KGs does not always boost the performance compared against open-domain KGs; (III) significant performance gains can be achieved by incorporating context embedding with existing neural networks; (IV) the joint model that combines knowledge and context embeddings can further boost the performance of existing models.

Acknowledgments

Xiaoli Wang was supported in part by the National Natural Science Foundation of China under Grant No. 61702432.

References

[1] T. Almeida and S. Matos, "Calling attention to passages for biomedical question answering," in *ECIR*, 2020, pp. 69–77.
[2] J. Mozafari, A. Fatemi, and M. A. Nematbakhsh, "BAS: an answer selection method using BERT language model," *CoRR*, vol. abs/1911.01528, 2019.
[3] M. Zhu, A. Ahuja, W. Wei, and C. K. Reddy, "A hierarchical attention retrieval model for healthcare question answering," in *WWW*, 2019, pp. 2472–2482.
[4] G. Luo, C. Tang, H. Yang, and X. Wei, "Medsearch: a specialized search engine for medical information retrieval," in *CIKM*, 2008, pp. 143–152.
[5] R. M. Terol, P. Martinezbarco, 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.
[6] H. Yu, M. Lee, D. Kaufman, J. Ely, J. A. Osheroff, G. Hripcsak, and J. Cimino, "Development, implementation, and a cognitive evaluation of a definitional question answering system for physicians," *Journal of biomedical informatics*, vol. 40, no. 3, pp. 236–251, 2007.



(a) DRMMTKS (b) DRMMTKS-W

(c) DRMMTKS-BC (d) DRMMTKS-BC-W

Fig. 6: Model visualization. In this example, the question is "what causes primary bone cancer" and the answer is "a cancerous (malignant) tumour starts from one abnormal cell". Each row corresponds to a token in the question and each column corresponds to a token in the answer. The darker areas indicate higher relevance.

[7] S. J. Athenikos and H. Han, "Biomedical question answering: A survey," *Computer Methods and Programs in Biomedicine*, vol. 99, no. 1, pp. 1–24, 2010.

[8] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition," in *AAAI*, 2016, pp. 2793–2799.

[9] B. Hu, Z. Lu, L. Hang, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *NIPS*, 2014, pp. 2042–2050.

[10] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *CIKM*, 2013, pp. 2333–2338.

[11] B. Mitra, F. Diaz, and N. Craswell, "Learning to match using local and distributed representations of text for web search," in *WWW*, 2017, pp. 1291–1299.

[12] S. MacAvaney, A. Yates, A. Cohan, and N. Goharian, "CEDR: contextualized embeddings for document ranking," in *SIGIR*, 2019, pp. 1101–1104.

[13] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "Learning semantic representations using convolutional neural networks for web search," in *WWW*, 2014, pp. 373–374.

[14] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng, "A deep architecture for semantic matching with multiple positional sentence representations," in *AAAI*, 2016, pp. 2835–2841.

[15] C. Xiong, J. Callan, and Z. Liu, "Convolutional neural networks for so-matching n-grams in ad-hoc search," in *WSDM*, 2017, pp. 126–134.

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

[17] M. Esposito, E. Damiano, A. Minutolo, G. D. Pietro, and H. Fujita, "Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering," *Information Sciences*, vol. 514, pp. 88–105, 2020.

[18] S. Arnold, B. van Aken, P. Grundmann, F. A. Gers, and A. Löser, "Learning contextualized document representations for healthcare answer retrieval," in *WWW*, 2020, pp. 1332–1343.

[19] W. Yoon, J. Lee, D. Kim, M. Jeong, and J. Kang, "Pre-trained language model for biomedical question answering," in *PKDD*, 2019, pp. 727–740.

[20] Y. Shen, Y. Deng, M. Yang, Y. Li, N. Du, W. Fan, and K. Lei, "Knowledge-aware attentive neural network for ranking question answer pairs," in *SIGIR*, 2018, pp. 901–904.

[21] W. Cui, Y. Xiao, H. Wang, Y. Song, and W. Wei, "Kbqa: Learning question answering over qa corpora and knowledge bases," vol. 10, no. 5, 2017, pp. 565–576.

[22] M. H. Namaki, Q. Song, Y. Wu, and S. Yang, "Answering why-questions by exemplars in attributed graphs," in *SIGMOD*, 2019, pp. 1481–1498.

[23] Q. Song, M. H. Namaki, and Y. Wu, "Answering why-questions for subgraph queries in multi-attributed graphs," in *ICDE*, 2019, pp. 40–51.

[24] W. Zheng, L. Zou, X. Lian, J. X. Yu, S. Song, and D. Zhao, "How to build templates for rdf question/answering: An uncertain graph similarity join approach," in *SIGMOD*, 2015, pp. 1809–1824.

[25] Y. Deng, Y. Xie, Y. Li, M. Yang, N. Du, W. Fan, K. Lei, and Y. Shen, "Multi-task learning with multi-view attention for answer selection and knowledge base question answering," in *AAAI*, 2019, pp. 6318–6325.

[26] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *TKDE*, vol. 29, no. 12, pp. 2724–2743, 2017.

[27] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3090253, IEEE Transactions on Knowledge and Data Engineering

Question How common is Henoch-Schönlein Purpura

Answers	DRMMTKS	DRMMTKS-W	DRMMTKS-BC	DRMMTKS-BC-W
TOP-1	Henoch-Schönlein purpura (HSP) is an immune-mediated condition. This means that it develops because of an abnormal reaction of the body's defence (immune) system. It is not clear exactly what causes this reaction but it is thought that something acts as a trigger for HSP. For example, the trigger may be a particular infection or certain medicines, such as certain antibiotics.	Someone with Henoch-Schönlein purpura (HSP) will often have had an upper respiratory tract infection within the few weeks before they develop the condition. So, for example, they may have had a cough, runny nose, and high temperature (fever) and have been feeling tired.	HSP is not very common. Between 8 and 20 in 100,000 people will develop HSP each year. HSP mostly affects children, especially children under the age of 10 years. But HSP can also affect older children and adults. It is more common in boys than in girls. Children under the age of 2 years tend to develop milder symptoms. Adults with HSP tend to develop more severe symptoms and are more likely to develop complications.	HSP is not very common. Between 8 and 20 in 100,000 people will develop HSP each year. HSP mostly affects children, especially children under the age of 10 years. But HSP can also affect older children and adults. It is more common in boys than in girls. Children under the age of 2 years tend to develop milder symptoms. Adults with HSP tend to develop more severe symptoms and are more likely to develop complications.
TOP-2	Someone with Henoch-Schönlein purpura (HSP) will often have had an upper respiratory tract infection within the few weeks before they develop the condition. So, for example, they may have had a cough, runny nose, and high temperature (fever) and have been feeling tired.	enoch-Schönlein purpura (HSP) is an immune-mediated condition. This means that it develops because of an abnormal reaction of the body's defence (immune) system. It is not clear exactly what causes this reaction but it is thought that something acts as a trigger for HSP. For example, the trigger may be a particular infection or certain medicines, such as certain antibiotics.	Norovirus is the most common virus causing infection of the gut (gastroenteritis) in adults in the UK. However, norovirus infection can occur in anyone of any age. You can get norovirus infection more than once because your body is not able to maintain immunity to norovirus infection for a long time once you have had it.	Norovirus is the most common virus causing infection of the gut (gastroenteritis) in adults in the UK. However, norovirus infection can occur in anyone of any age. You can get norovirus infection more than once because your body is not able to maintain immunity to norovirus infection for a long time once you have had it.
TOP-3	In many people with HSP, no complications develop. But, complications sometimes develop. They can include the following: Kidney involvement - in around half of people with HSP, the kidneys become affected. If immune complexes are deposited in the kidneys, this can lead to inflammation of the kidneys, known as nephritis. This complication usually develops within one month after the rash starts but can sometimes develop up to six months afterwards.	HSP is not very common. Between 8 and 20 in 100,000 people will develop HSP each year. HSP mostly affects children, especially children under the age of 10 years. But HSP can also affect older children and adults. It is more common in boys than in girls. Children under the age of 2 years tend to develop milder symptoms. Adults with HSP tend to develop more severe symptoms and are more likely to develop complications.	Henoch-Schönlein purpura (HSP) is an immune-mediated condition. This means that it develops because of an abnormal reaction of the body's defence (immune) system. It is not clear exactly what causes this reaction but it is thought that something acts as a trigger for HSP. For example, the trigger may be a particular infection or certain medicines, such as certain antibiotics.	Someone with Henoch-Schönlein purpura (HSP) will often have had an upper respiratory tract infection within the few weeks before they develop the condition. So, for example, they may have had a cough, runny nose, and high temperature (fever) and have been feeling tired.

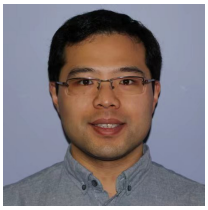
Fig. 7: An example of a question and its top-3 results. The correct answers are colored in red.

enhanced language representation with informative entities," in *ACL*, 2019, pp. 1441–1451.
[28] F. Luo, X. Wang, Q. Wu, J. Liang, X. Qiu, and Z. Bao, "Hqadeep-helper: A deep learning system for healthcare question answering," in *WWW*, 2020, pp. 194–197.
[29] G. Wiese, D. Weissenborn, and M. L. Neves, "Neural domain adaptation for biomedical question answering," in *CoNLL*, 2017, pp. 281–289.
[30] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W. B. Croft, and X. Cheng, "A deep look into neural ranking models for information retrieval," *ArXiv*, vol. abs/1903.06902, 2019.
[31] T. Sagara and M. Hagiwara, "Natural language neural network and its application to question-answering system," *Neurocomputing*, vol. 142, pp. 201–208, 2012.
[32] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.
[33] K. Hui, A. Yates, K. Berberich, and G. de Melo, "PACRR: A position-aware neural IR model for relevance matching," in *EMNLP*, 2017, pp. 1049–1058.
[34] Z. Yang, Q. Lan, J. Guo, Y. Fan, X. Zhu, Y. Lan, Y. Wang, and X. Cheng, "A deep top-k relevance matching model for ad-hoc retrieval," in *Information Retrieval*, 2018, pp. 16–27.
[35] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power, "End-to-end neural ad-hoc ranking with kernel pooling," in *SIGIR*, 2017, pp. 55–64.
[36] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," pp. 2227–2237, 2018.
[37] S. Yoon, F. Dernoncourt, D. S. Kim, T. Bui, and K. Jung, "A compare-aggregate model with latent clustering for answer selection," in *CIKM*, 2019, pp. 2093–2096.
[38] Y. Hua, Y.-F. Li, G. Haffari, G. Qi, and W. Wu, "Retrieve, program, repeat: Complex knowledge base question answering via alternate meta-learning," *arXiv preprint arXiv:2010.15875*, 2020.
[39] T. Shen, X. Geng, T. Qin, D. Guo, D. Tang, N. Duan, G. Long, and D. Jiang, "Multi-task learning for conversational question answering over a large-scale knowledge base," *arXiv preprint arXiv:1910.05069*, 2019.
[40] T. Shen, X. Geng, T. Qin, G. Long, J. Jiang, and D. Jiang, "Effective search of logical forms for weakly supervised knowledge-based question answering," *arXiv preprint arXiv:1909.02762*, 2019.
[41] Y. Deng, Y. Xie, Y. Li, M. Yang, N. Du, W. Fan, K. Lei, and Y. Shen, "Multi-task learning with multi-view attention for answer selection and knowledge base question answering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6318–6325.
[42] Q. Bao, L. Ni, and J. Liu, "Hhh: an online medical chatbot system based on knowledge graph and hierarchical bi-directional attention," in *Proceedings of the Australasian Computer Science Week Multiconference*, 2020, pp. 1–10.
[43] Y. Zhang, S. Qian, Q. Fang, and C. Xu, "Multi-modal knowledge-aware hierarchical attention network for explainable medical question answering," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1089–1097.
[44] P. Tong, Q. Zhang, and J. Yao, "Leveraging domain context for question answering over knowledge graph," *Data Science and Engineering*, vol. 4, no. 4, pp. 323–335, 2019.
[45] P. Michel, O. Levy, and G. Neubig, "Are sixteen heads really better than one?" in *Advances in Neural Information Processing Systems*, 2019, pp. 14 014–14 024.
[46] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.
[47] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, 2019.
[48] I. Beltagy, A. Cohan, and K. Lo, "Scibert: Pretrained contextualized embeddings for scientific text," *arXiv preprint arXiv:1903.10676*, 2019.
[49] Y. Deng, Y. Shen, M. Yang, Y. Li, N. Du, W. Fan, and K. Lei, "Knowledge as a bridge: Improving cross-domain answer selection with external knowledge," in *Proceedings of the 27th international conference on computational linguistics*, 2018, pp. 3295–3305.
[50] D. N. Milne and I. H. Witten, "Learning to link with wikipedia," in *CIKM*, 2008, pp. 509–518.
[51] M. Neumann, D. King, I. Beltagy, and W. Ammar, "Scispacy: Fast and robust models for biomedical natural language processing," *arXiv preprint arXiv:1902.07669*, 2019.
[52] D. Rao, P. McNamee, and M. Dredze, "Entity linking: Finding extracted entities in a knowledge base," in *Multi-source, Multilingual Information Extraction and Summarization*, 2013, pp. 93–115.
[53] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *NIPS*, 2013, pp. 2787–2795.
[54] A. Aizawa, "An information-theoretic perspective of tf-idf measures," in *Information Processing and Management*, 2003, pp. 45–65.
[55] J. Guo, Y. Fan, Q. Ai, and W. B. Croft, "A deep relevance matching model for ad-hoc retrieval," in *CIKM*, 2016, pp. 55–64.
[56] V. Boteva, D. G. Ghalandari, A. Sokolov, and S. Riezler, "A full-text learning to rank dataset for medical information retrieval," in *ECIR*, 2016, pp. 716–722.
[57] C. V. Gysel and M. de Rijke, "Pytrec_eval: An extremely fast python interface to trec_eval," *CoRR*, vol. abs/1805.01597, 2018. [Online]. Available: http://arxiv.org/abs/1805.01597
[58] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR abs/1412.6980*, 2014.

XIAOLI WANG received the B.S. degree from the Northeastern University of China, and the Ph.D. degree from the National University of Singapore, both in computer science. She is currently an associate professor at School of Informatics in Xiamen University, Xiamen, China. Her research interests mainly focus on database and data mining relevant issues, including indexing and query processing on complex structures, big healthcare data, and social media computing.

FENG LUO received her Master degree in software engineering from Xiamen University in 2021. She received her B.S. degree in software engineering from FuJian Normal University in 2018. Her research interests include natural language processing and big healthcare data.

QINGFENG WU received the B.S. and Ph.D. degrees from the School of Informatics, Xiamen University, Xiamen, China. He is currently a professor at School of Informatics in Xiamen University. His research interests include big data and cloud computing, digital media technology and bioinformatics.



ZHIFENG BAO received the Ph.D. degree in computer science from the National University of Singapore in 2011 as the winner of the Best PhD Thesis in school of computing. He is currently an Associate Professor at RMIT University. He is also an Honorary Senior Fellow with University of Melbourne in Australia. His current research interests include data usability, spatial database, graph data analytics and data integration.