ICIS 2022 Proceedings

IS in Healthcare

Dec 12th, 12:00 AM

Designing a Healthcare QA Assistant: A Knowledge Based Approach

Prakash Chandra Sukhwal
Institute of Systems Science, isspcs@nus.edu.sg

Atreyi Kankanhalli
National University of Singapore, atreyi@comp.nus.edu.sg

Vaibhav Rajan
National University of Singapore, vaibhav.rajan@nus.edu.sg

Recommended Citation

Sukhwal, Prakash Chandra; Kankanhalli, Atreyi; and Rajan, Vaibhav, "Designing a Healthcare QA Assistant: A Knowledge Based Approach" (2022). *ICIS 2022 Proceedings*. 1.
https://aisel.aisnet.org/icis2022/is_health/is_health/1

Designing a Healthcare QA Assistant: A Knowledge Based Approach

Short Paper

Prakash Chandra Sukhwal
Institute of Systems Science
29 HMK Terrace, Singapore 119620
prakashs@nus.edu.sg

Atreyi Kankanhalli
National University of Singapore
15 Computing Drive, Singapore 117418
atreyi@comp.nus.edu.sg

Vaibhav Rajan
National University of Singapore
15 Computing Drive, Singapore 117418
vaibhav.rajan@nus.edu.sg

Abstract

Question answer (QA) assistants are vital tools to address users' information needs in healthcare. Knowledge graphs (KGs) and language models (LMs) have shown promise in building QA systems, but face challenges in their integration, and performance. Motivated thus, we take the case of a specific disease, skin eczema, to design a QA system combining KG and LM approaches. We present design iterations for systematically developing the KG, then fine-tuning a LM, and finally carrying out joint reasoning over both. We observe that while KGs are effective for fact finding, fine-tuned LMs perform better at answering complex queries. Initial results suggest that combining KG and LM approaches can improve the performance of the system. Our study contributes by laying out the design steps and developing a QA system that addresses various gaps in the related literature. Our future plan is to refine these techniques towards building a full-fledged healthcare QA assistant.

Keywords: QA assistant, Healthcare, Knowledge graph, Language model, Design

Introduction

In recent times, the Internet has become a popular source for people to seek information. For healthcare information, people often visit online forums and browse medical sites to find answers to their questions about ailments, symptoms, and remedies. In fact, around 4-5% of Internet searches are related to healthcare information (Lin et al. 2016). However, much of the content in online health forums is user-generated and may misguide users due to inaccuracies, failure to update the information, and lack of validation by healthcare professionals (John et al. 2016). Further, individuals often require specific information personalized to their health condition that may not be addressed by online search e.g., by Google. Thus, there is an unmet need for personal *healthcare question-answer (QA) assistants*, which refer to software that can automatically glean the required information from online knowledge resources and user's personal details, to present tailored answers to users' questions.

A key component of such QA assistants are QA systems. An automated *QA system* refers to a software that provides natural language answers to users' questions by acquiring and processing information from online sources and/or by building a structured knowledge base (Zhu et al. 2021). These QA systems differ from QA forums or online communities examined in several IS studies (e.g., John et al. 2016; Lin et al. 2016; Liu et al. 2019b; Khurana et al. 2019; Peng et al., 2020) where humans answer questions from other humans online. In contrast, our study is about designing an automated QA system for healthcare queries with no human involved in answering the questions.

The focal QA systems in our study also differ from chatbots for customer/patient service (e.g., booking appointments), work productivity (e.g., supporting collaborative work), or even social purposes that are often examined in the literature (Diederich et al. 2022). Compared to such chatbots, which usually have a well-defined set of rules (Bates 2019) or are of social nature (Zhou et al. 2020), the focal QA systems in our study need to acquire specialized domain knowledge from multiple technical sources in order to create a robust and updateable knowledge base and representation that can be efficiently browsed to find relevant answers to users' questions (Jiang et al. 2021).

While progress has been made in QA techniques, QA systems continue to face challenges in their integration (Singh et al. 2018), and performance (Jiang et al. 2021) that affect end-user experience. First, there are often ambiguous or incorrect answers (Janssen et al. 2021; Dugan et al. 2022) in system responses. Second, the systems are often unable to update their knowledge base to keep their information current (Janssen et al. 2021). Finally, the current systems do not customize the responses for different users (Zhang et al. 2021). These issues can be tackled by leveraging on advances in the field of natural language processing (NLP), particularly knowledge graphs and language models.

Knowledge graphs (KGs) are a means of representing knowledge by concepts and their relationships through a graphical structure of *nodes* (or entities) and *edges* (or relations) (Chiu et al. 2021). An edge and the two nodes it connects form a *triplet* representing a "fact", e.g., "*Siri* is a *digital assistant*" has two nodes (in italics) connected by the "is a" relation. KGs are useful for knowledge representation and retrieval tasks (Fecho et al. 2021). In healthcare QA systems, KGs have been used for answering questions on dietary recommendations (Chen et al. 2021), producing sub-graphs in response to biomedical research questions (Fecho et al. 2021), and generating rule-based and template-based short responses to medical questions (Jiang et al. 2021), among other uses. Nevertheless, we identified several gaps in the literature on KG-based QA systems. First, there has been a lack of attention to resolution of synonymous entities and abbreviations. Second, the disconnectedness of KGs (absence of edges between relevant/similar nodes) has not been addressed (Gong et al. 2021). Third, existing systems are unable to capture the semantic similarity of entities using graph embeddings, which facilitate downstream tasks for knowledge discovery (Chiu et al. 2021). We aim to address these issues as discussed later.

In contrast to KGs, *language models* (LMs) are probability-based models that learn statistical properties of the sequential distribution of words in text documents (Bengio 2008). By learning from large text corpora (collections) and encoding knowledge in a manner different from KGs, LMs are also useful for question answering. Petroni et al. (2019) note that there are several benefits of using LMs as knowledge bases for QA systems – they obviate the need for schema (cognitive frameworks), enable open-domain questions (while KG are usually domain restricted), are easy to extend with more data, and have less chances of yielding erroneous responses given quality data. Also once created, LMs can be adapted (or fine-tuned) with data from a different domain (Gururangan et al. 2020). In the healthcare context, a LM approach was used to create a QA system for COVID-19 infection (Graf et al. 2022). The system relied on a precompiled dataset and produced candidate answers using a pre-trained LM. It was not intended to account for knowledge updates. In the education domain, a LM approach was used for designing a student-centric QA system (Wambsganss et al. 2021). The system used text generated in the classroom, rather than a knowledge base. We identified the following gaps in the LM-based QA systems literature. First, the issue of incorrect or ambiguous answers has received less attention (Dugan et al. 2022). For a given question, a LM produces multiple candidate answers ranked by scores. The actual answer might get a lower score amongst the candidate answers. Second, there has been lack of domain adaptation (or finetuning) of LMs to obtain more relevant and precise answers (Gururangan et al. 2020). Our work aims to address these gaps.

KGs are relatively easy to update and good at finding facts due to their graphical schema, but struggle with complex queries. On the other hand, LMs are good at capturing context in answering complex queries, and work better with noisy data (Yasunaga et al. 2021) but they produce multiple candidate answers that need filtering. Recently researchers have started exploring a joint reasoning approach, where a QA system relies on both LM and KG. So far, this approach has been used for answering multiple-choice questions (MCQs) (Sun et al. 2021; Yasunaga et al. 2021). Thus, the design process of a joint-reasoning system for short or long QA that addresses the above issues remains unexplored.

Motivated thus, we present our initial work to answer the research question: *What are the design steps and their step-wise evaluation for integrating KG and LM approaches for a healthcare QA system?* To this end, we present the design iterations of: (i) developing a disease-specific KG for the QA system, (ii) fine-tuning LMs with annotated data, and (iii) joint reasoning over KG and LM for QA. While we describe and evaluate our design steps for a QA system for a single disease (skin eczema) to start with, these are extendable to other diseases. We gathered research articles on skin eczema (atopic dermatitis) from medical journals sourced from PubMed (a quality source) to create our KG. We then used annotated question-context-answer sets from the PubMed abstracts to fine-tune 4 popular LMs. Our preliminary investigations with these three iterations suggest that the second and third iterations performed better in terms of providing accurate answers for long sentence questions than using only a KG based approach. We conclude the paper by describing the study contributions and the next steps for advancing our work.

Background Concepts and Literature

Knowledge Graphs

KGs are constructed by systematically extracting concepts/entities (nodes), relations (edges) between pairs of entities, and attributes (properties of nodes/edges) from large bodies of text. KG is a structured representation connecting subject entities (or **head, h**) to object entities (or **tail, t**) via edges/ relations **(r)** using a schema, which provides the backbone semantics for further inference using graph mining techniques (Chiu et al. 2021).

However, KGs are well known for being incomplete. For example, the DrugBank KG has a significant number of missing drug-drug interactions, which impedes the generation of comprehensive medication related precautions for patients (Gong et al. 2021). A *KG completion* task can be understood as either: (i) an entity prediction task, where an entity h or t is missing **[(?, r, t) or (h, r, ?)]**, or (ii) a relation prediction task, where r is missing **(h, ?, t)** in a fact triplet. A common approach for KG completion is through the use of KG embeddings. *KG embeddings* are numeric vector based effective low-dimensional representations of KG nodes (Gong et al. 2021). The core idea here is to learn the properties of the KG and find a function that can map the entities and relations to representations (also called embeddings) in a vector space that preserve the KG relationships. While there are a number of embedding methods, in our work we used a common 1D (dimension) method, TransE (Bordes et al. 2013) and a common 2D method, ConvE (Dettmers et al. 2018) for comparison. TransE is a 1D distance-based embedding technique, whereas ConvE uses a convolutional neural network to capture non-linear relations and generate embeddings based on 2D graph convolutions. Since our aim was to embed the graph nodes and relations for QA purposes only, we experimented with both TransE and ConvE to make an initial design decision on the embedding technique. In future, we plan to evaluate other embedding methods.

Language Models

Using large volumes of textual data, LMs model the probability distribution of word sequences, along with the context (of words) surrounding words. Recent and popular LMs include, BERT (Devlin et al. 2018) from Google, RoBERTa (Liu et al. 2019) from Facebook, and ELECTRA (Clark et al. 2020) that differ in the way they are trained. The performance of these models has been tested on many downstream NLP tasks, such as for QA using the 'Stanford Question Answering Dataset' (SQuAD). After being trained on a large general text corpus (collection) such as Wikipedia, LMs can be made more domain-specific through *fine tuning* (Gururangan et al. 2020) on text from a particular domain, such as healthcare (using PubMed text). Fine tuning involves training the top neural network layers of a trained LM model so that the vectors learn and adjust using the task-relevant unlabeled data, such as SQuAD. In our work, we compared basic versions of BERT and RoBERTA trained using non-medical data, with variants of ELECTRA (BioM) and BERT (BioBERT) pre-enriched with biomedical data, in order to assess if there are any performance gains when we further fine tune a pre-enriched LM. We fine-tuned these models with our disease specific data and queried each of them using natural language questions to assess their performance.

Design Framework and Data Preparation

Our design framework is shown in **Figure 1**. First, we prepared our data for all design iterations i.e., KG generation (iteration 1), finetuning existing LMs (iteration 2), and joint reasoning over KG and LM (iteration 3). We explain the design steps, to address the issues and gaps mentioned in the introduction.

First, it is critical to acquire error-free healthcare information for the system design. To ensure high quality of data, we collected articles on skin eczema from scientific journals through the database PubMed DB. PubMed DB is a premier source of medical knowledge, being the largest collection of high quality research papers in this domain (Williamson and Minter 2019). This ensured the quality and reliability of our data corpus. We used the query - "*atopic eczema [OR] atopic dermatitis*" and searched PubMed using the R package *easyPubMed*. As a result, we obtained 500 abstracts, and filtered them for errors or duplication to obtain 425 relevant abstracts on skin eczema. Second, medical text is often full of abbreviations not known to lay users. To *increase their understanding*, we ran an *abbreviation detection* procedure in Python, to add the full-form of medical abbreviations over the whole data corpus using the Scispacy library (specially designed for biomedical text). As this step added repeated terms to the corpus, data redundancy was resolved through a de-duplication step later. *Third*, we ran *coreference resolution* on the individual sentences replacing all references of a main entity with the name of the main entity. This helped in *reducing ambiguity from individual sentences* and building a more connected KG, e.g., *"Eczema is a prevalent disease in Asia. It is particularly found in infants"* was changed to *"Eczema is a prevalent disease in Asia. Eczema is particularly found in infants"*. As a result, the sentences can be linked, with 'Eczema' as common entity.

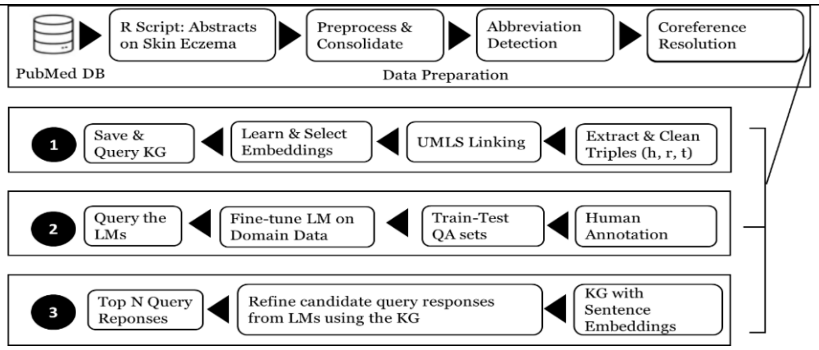
R Script: Abstracts on Skin Eczema
Preprocess & Consolidate
Abbreviation Detection
Coreference Resolution
PubMed DB
Data Preparation
1
Save & Query KG
Learn & Select Embeddings
UMLS Linking
Extract & Clean Triples (h, r, t)
2
Query the LMs
Fine-tune LM on Domain Data
Train-Test QA sets
Human Annotation
3
Top N Query Reponses
Refine candidate query responses from LMs using the KG
KG with Sentence Embeddings

Figure 1. Design Framework for Proposed QA System

Design Iteration 1: KG-based

Methodology

With the coreference resolved data from the data preparation step (see **Figure 1)**, we generated unique triplet sets (*h, r, t*) using the Stanford OpenIE library. In the data corpus, several synonyms exist for a particular term. These synonyms posed a problem as they can confuse users if they are not linked together. Thus, we observed that the initial KG had many disconnected nodes with synonymous entities (e.g. disease references: (*eczema, causes, itching*); (*atopic dermatitis, causes, skin rashes*)). To remedy this issue, we enriched the triplets using the unified medical language system (UMLS). UMLS codifies the knowledge of various biomedical standards and terminologies to provide a set of centralized biomedical vocabularies. We ran our coreferenced text to extract ConceptIDs (CUIs) for all the entities in our corpus that match registered entities in the UMLS database. Using Scispacy and UMLS, we injected 7,179 CUIs in our KG. Unlike earlier work (e.g., Chen and Li 2018; Jiang et al. 2021), we directly introduced the CUIs in the triplets (e.g. (*eczema, is, C6578*); (*atopic dermatitis, is, C5678*)) instead of using them as node properties. Through abbreviation resolution and UMLS linking, we were able to link multiple synonymous entities to obtain a more connected disease-specific KG (see **Figures 2.A, 2.B**).

Further, unlike prior research, we obtained semantically aware low-dimensional vector representations of the triples using the graph embedding techniques (described below) that facilitate downstream graph analytics and knowledge discovery tasks (Chiu et al. 2021). Using triplet corruption or the negative sampling technique, we learned the embeddings for our KG with TransE and ConvE methods. The training involved the following split of dataset: for training (51,094 triplets), for validation (500 triplets), and for testing (1000 triplets). A learning rate of 0.001, Adam optimizer, batch size of 10, max epochs of 1000 and negative log likelihood loss (NLL, a loss function for multi-class classification problems) were used for training. An early stopping criterion was used to stop training when no improvement was observed in the mean reciprocal rank (MRR) on the validation set.

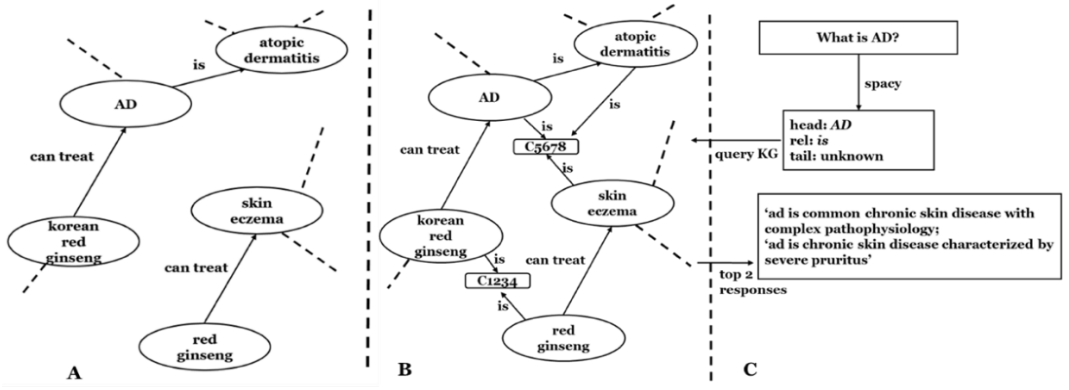

Figure 2. Enriching the KG with UMLS and Querying

Evaluation and Results

For measuring the accuracy of the graph embeddings in predicting subject or object entities, when one of them is missing, we used the popular metrics - *Hits@N* and *MRR* (Mohamed et al. 2020). Hits@N ranks triplets and computes the percentage of positive triplets ranked in the top N results. MRR measures the average of reciprocal ranks of all the triplets. Higher values of these metrics indicate better accuracy of graph embeddings. Accordingly, we chose to use TransE, as it performed better on both metrics (**Table 1**).

Method	Hits@1	Hits@10	Hits@100	MRR
TransE	**0.69**	**0.8**	**0.92**	**0.73**
ConvE	0.01	0.04	0.12	0.02

Table 1. Test Set Performance of Learned Embeddings for KG

A KG has a well-defined structure (h, r, t) and KG embeddings further capture the semantic relationships between nodes and edges facilitating the triplet completion task. We used this fact for querying our KG. *First*, from a given natural language question we extracted the structure, using a custom program in Python, to see if it is of the form < **h, r, ?** > OR < ?, **r, t** > OR < **h, ?, t** >. We then used this structure to translate a question to a triplet format to query our KG. For assessing answer accuracy, we tested our KG with factual questions (see **Figures 2.B, 2.C**). An answer would score 1 (if correct), 0.5 (if partially correct), or 0 otherwise. While the KG showed good accuracy of 70% for short QA, it performed poorly on long questions (e.g., what is the reason for eczema?) with 25% accuracy. This led us to the use of LMs in the next iteration.

Design Iteration 2: Fine-tuned LM

Queries for a KG have to be created with much care to ensure consistency with the triplet structure of relations in KGs, which makes it difficult for them to handle long questions in natural language. Thus, in the second design iteration, we utilized pre-trained LMs (BERT and RoBERTA), which are considered as a form of open knowledge graphs (Wang et al. 2020) acquired from huge volumes of unstructured data. We also used enriched versions of these LMs (BioM-ELECTRA and BioBERT) that had been further trained on medical data (BioASQ8B and PubMed data). In this iteration, we fine-tuned these SQuAD2.0 pre-trained LMs i.e., BERT-base-squad2, RoBERTA-base-squad2, BioM-ELECTRA-base-squad2-BioASQ8B, and bioBERT-pubmed-squad2, and evaluated their performance with and without tuning using our dataset of PubMed abstracts on eczema that we annotated as described below.

Methodology

To fine-tune a given LM on a question-answer task, we needed to provide a formatted input of question, followed by a context (a paragraph which contains the answer), which is followed by an answer. We created this question-context-answer dataset manually from our coreference resolved data using an online annotation tool (Haystack 2020). Annotation involved reading the cleaned data to create questions, tagging a portion of text as a context for the question, and tagging an answer for the question within the same context. Specifically, a set of 70 questions served as the training set and 23 questions served as the validation set. The entire set of 93 QA pairs was created by the same researcher to maintain consistency. The four LMs were fine-tuned on a GPU machine (Tesla P100-PCIE-16GB, compute capability: 6.0) using the annotated data. The fine-tuning process took around 1 hour (see **Table 2**) per model. The pipeline for fine-tuning was created using the infrastructure provided by haystack. In future, we aim to improve this component with automated QA generation, before using the data to fine tune a LM. This will make our pipeline fully automated and the fine-tuning will effectively happen in a one-off manner from time to time as needed.

Evaluation and Results

Three established metrics for measuring the performance of fine-tuned LMs are – (i) Exact Match (**EM**), (ii) **F1** score, and (iii) Semantic Answer Similarity (**SAS**) (Risch et al. 2021). EM is a binary metric that indicates how many of the answers overlap 100% with an already provided answer string. F1 score captures how many words overlap in the predicted and true answers. Both these metrics cover lexical similarity, and do not capture the semantic similarity between two given strings. For example, the strings 'twenty percent' and '20%' would be considered different in calculating EM and F1 scores. Thus, we also used SAS, which assesses the semantic similarity between strings, to compare the LMs.

	1	2	3	4	5	6	7	8
	BERT-base (SQuAD 2)	BERT-base (SQuAD2 + human)	RoBERTa - base (SQuAD2)	RoBERTa - base (SQuAD2 + human)	BioM-ELECTRA-Base-SQuAD2-BioASQ8B	BioM-ELECTRA-Base-SQuAD2-BioASQ8B + human	BioBERT-pubmed (SQuAD 2)	BioBERT-pubmed (SQuAD2 + human)
Time	-	1.15 Hrs.	-	1.25 Hrs.	-	1 Hrs.	-	1.23 Hrs.
EM	0	**0.18**	0	0.125	0	0	0	0
F1	0.401	0.418	0.351	**0.502**	0.259	0.4253	0.46	0.453
SAS	0.676	0.737	0.652	**0.785**	0.648	0.703	0.713	0.741

Table 2. Performance of Fine-tuned LMs with Annotated Question-context-answer Dataset

On the stringent EM metric (see **Table 2**) all models performed poorly. However, all models showed better performance for both F1 and SAS, especially after fine-tuning with annotated data (columns 2,4,6,8 as compared to 1,3,5,7). Of the four doubly fine-tuned models (columns 2, 4, 6, 8), we see that the RoBERTa-based model (column 4) performed better than the rest in terms of F1 score and SAS.

Design Iteration 3: Joint Reasoning over KG and LM

The underlying idea is conceptually simple. First, a LM model is used to create embeddings for all the prepared data. This step is done using sentence embeddings i.e., vector representations of sentences (Wambsganss et al. 2021) that enable their comparison in vector space. Second, a sentence-level KG is prepared with linkages across all sentences where common entities are present. Third, based on the KG, the embeddings of the prepared data are improved via graph convolution techniques. Finally, an answer is retrieved from the embedded data based on similarity of entities in the query and those present in the sentence KG. In our initial efforts on joint reasoning for design iteration 3, we adapted from Gannon (2020) using our annotated dataset and design choices (of sentence embeddings) as described below.

Methodology

First, we developed a KG by systematically parsing every line of the coreference resolved data (see **Figure 1**). While reading each line we extracted entities present in a sentence and looked for their presence in the Wikidata KG using Google language services for named entity recognition. For each entity in our data, if a match existed with a Wikidata entity, we got an entity code which could be used to look up further information on the matched entity e.g., [('Q8054', 'protein'), ('Q7187', 'gene'), ('Q79455', 'immunoglobulin E'). Our KG generated in this manner had 8,228 nodes and 9,669 relations. Different from design iteration 1, each sentence, out of the 4,132 sentences in the coreferenced corpus, was represented by a node in this KG. If two sentences had a common entity (word or phrase) then they were connected in the KG via that entity acting as a bridge. To achieve this, the initial data preparation steps of abbreviation detection and coreferenced resolution played crucial roles.

Second, we embedded every sentence (4,132 in total) of the coreferenced dataset using a sentence transformer model (all-mpnet-base-v2) guided by the performance ranking on a leaderboard of pre-trained models (Reimers 2022). This design choice yielded better quality sentence embeddings. We then followed the procedure in Gannon (2020) to enrich the embeddings with information from neighbors on the KG. After generating the sentence embeddings, we created a matrix ***m*** of all the sentence embeddings, where m[i] represents the ith sentence encoded using a 768-dimensional dense vector. Our matrix was of size 4132 x 768. After obtaining the matrix ***m***, for each sentence node, we found out the neighbors (other linked sentence nodes) in the KG and took a weighted sum of the neighboring nodes with the central node to obtain a new embedded representation for each sentence node and further normalized it. Specifically, to refine the representation of each KG node, we adopted a 3-layer convolution with varying weight parameter lambda ($\lambda \in [0, 1]$). In our case we chose $\lambda = 0.7$ for first update of node embedding to get ***m1*** from ***m.*** Then $\lambda = 0.9$ for the second update to get ***m2*** since the neighbors in the 2nd pass are further away and assumed to be less similar to the current sentence node. This convolution step differs from ConvE, and is a simple yet useful way to learn joint representations for a given node with respect to its neighboring nodes. We compared the four LMs (columns 2,4,6,8) of iteration 2 against the joint reasoning approach in iteration 3 (see **Table 3).** Due to space limits, we show only three key questions and their answers in **Table 3.**

Question ➡	***What is AD?***	***What are some symptoms of atopic dermatitis?***	***Any cure for eczema?***
Model ⬇	*Top Answers*		
BERT base (SQuAD2.0 + human)	'AD means Atopic dermatitis multidimensional implications'	'There are also some common symptoms for all types of eczema : dry , scaly skin , redness , itching'	'conditions , suppressive medications aim at managing the symptoms of these diseases'
Roberta-base (SQuAD2.0 + human)	'Atopic dermatitis'	'dry , scaly skin , redness , itching , which may be intense'	'no'
BioM-ELECTRA-Base-SQuAD2-BioASQ8B + human	'Our findings clearly suggested that VYAC treat AD through inhibiting the inflammatory mediator productions and blocking mast cell degranulation via suppressing Syk mediated NF-κB pathway'	'There are also some common symptoms for all types of eczema : dry , scaly skin , redness , itching , which may be intense'	'there is no'
Biobert-pubmed (SQuAD2.0 + human)	'AD means atopic dermatitis'	'There are also some common symptoms'	'there is no definitive cure'
*Joint reasoning (**Iteration 3**)*	*'AD means Atopic dermatitis patients are heavily infected with Staphylococcus aureus on the skin'*	*'Skin manifestations ranged from atopic and seborrheic dermatitis to psoriasiform rash.'*	*'Eczema is a skin disease that often requires long-term treatment.'*

Table 3. Answers by 4 fine-tuned LMs and Joint Reasoning on Atopic Dermatitis

Evaluation and Results

Given a question in natural language form, it was processed to identify the entities in the question and then these entities were compared with the KG to find the maximum overlapping sentences in terms of matches found for the extracted entities in the question. The identified closest matching *n* sentences in ***m2*** then served as the answers. Training and validation sets used were the same as those in iteration 2. In our initial review of the answers, we found that the joint reasoning approach provided accurate, unambiguous, and natural looking full-sentence answers to all 3 key questions (see **Table 3**). Since the answers may still contain medical jargon owing to the source of the data corpus (i.e., PubMed), in future we intend to solve this issue by two means. First, we will add layers of paraphrasing to simplify the system answers. Second, we will add a module on dialogue management to make the QA session more interactive and natural.

Initial Contributions and Future Work

In this short paper, we presented our in-progress work towards building a QA assistant for healthcare. Through this work, we offer initial contributions on designing QA systems for healthcare in the following ways. *First*, we explicate the data procurement, cleaning, and enrichment process for a healthcare QA system. *Second,* we present the detailed steps and choices for designing a disease-specific KG and fine-tuning existing LMs for the QA system. Compared to KGs like NELL and Wikidata which require much human intervention (Wang et al. 2020), our pipeline for KG generation is largely automatic, and has been adapted for healthcare through new design elements (e.g., medical abbreviation detection; UMLS linking). *Third*, we provide a joint reasoning methodology, using a novel synthesis of techniques that effectively utilize both KG and LM, to enhance QA system performance. To our knowledge, such an approach has not been used in previous studies that have used either KGs or LMs independently. *Last*, we propose ways to evaluate the system performance in each design iteration. Collectively, these add to IS research on the design of QA systems and assistants. Our data and design choices also have direct practical implications for enabling accurate answers and providing scalable KG design.

In future, we plan to do the following to advance our work. First, we will refine the design of the KG and LM as discussed earlier in the paper and perform the integration of our KG from step 1 in the joint reasoning design iteration. Second, to enhance the answer quality, we will improve the LM by more fine-tuning (which will be scaled up by automating data annotation) (Gururangan et al. 2020), and perform answer simplification to remove jargon. Third, to enhance interactivity we will add a module for dialogue management to make the QA session more natural. Fourth, we will add a module for user-level personalization using their past conversations and user profiles (Zhang et al. 2021). Finally, to evaluate the quality of answers, we will design user evaluation scales (Chen et al. 2021) involving lay users and medical professionals. For lay users we will measure perceptions such as relevance and comprehensibility of answers. For medical practitioners we assess perceptions such as the completeness, informativeness, and truthfulness of answers of answers (Zhu et al. 2009). In addition, we plan to apply the findings to other domains to understand the transferability of our work.

Acknowledgements

This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-016).

References

Bates, M. 2019. "Health care chatbots are here to help", IEEE Pulse, 10(3), pp. 12-14.

Bengio, Y. 2008. "Neural net language models", Scholarpedia, 3(1), p. 3881

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. 2013. "Translating Embeddings for Modeling Multi-Relational Data," in Neural Information Processing Systems, pp. 1-9.

Chen, Q., and Li, B. 2018. "Retrieval Method of Electronic Medical Records Based on Rules and Knowledge Graph," ICEB, 42.

Chen, Y., Subburathinam, A., Chen, C. H., and Zaki, M. J. 2021. "Personalized Food Recommendation as Constrained Question Answering over a Large-scale Food Knowledge Graph," in Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 544-552.

Chiu, B., See-To, W.K., and Ngai, E.W. 2021. "Knowledge Graph Construction and Applications in E-Retailing: A Review of Literature," PACIS 2021 Proceedings, 244.

Clark, K., Luong, M.T., Le, Q.V., and Manning, C.D. 2020. "Electra: Pre-training Text Encoders as Discriminators Rather than Generators," arXiv:2003.10555.

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. 2018. "Convolutional 2D Knowledge Graph Embeddings," In 32nd AAAI Conference on Artificial Intelligence Proceedings, (32:1).

Devlin, J., Chang, M.W., Lee, K., and Toutanova, K. 2018. "Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv:1810.04805.

Diederich, S., Brendel, A.B., Morana, S., and Kolbe, L. 2022. "On the Design of and Interaction with Conversational Agents: An Organizing and Assessing Review of Human-Computer Interaction Research." JAIS, 23(1), pp. 96-138.

Dugan, L., Miltsakaki, E., Upadhyay, S., Ginsberg, E., Gonzalez, H., Choi, D., and Callison-Burch, C. 2022. "A feasibility study of answer-agnostic question generation for education," arXiv:2203.08685.

Fecho, K., Bizon, C., Miller, F., Schurman, S., Schmitt, C., Xue, W., and Tropsha, A. 2021. "A Biomedical Knowledge Graph System to Propose Mechanistic Hypotheses for Real-World Environmental Health Observations: Cohort Study and Informatics Application," JMIR Medical Informatics, 9(7), e26714.

Gannon D. 2020. "Building a Tiny Knowledge Graph with BERT and Graph Convolutions". DOI: 10.13140/RG.2.2.23560.44804

Gong, F., Wang, M., Wang, H., Wang, S., and Liu, M. 2021. "SMR: Medical Knowledge Graph Embedding for Safe Medicine Recommendation," Big Data Research, 23, 100174.

Graf, J., Lancho, G., Zschech, P., and Heinrich, K. 2022. "Where Was COVID-19 First Discovered? Designing a Question-Answering System for Pandemic Situations," ECIS 2022 Proceedings.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N.A. 2020. "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks," arXiv:2004.10964.

Haystack by deepset, 2020. https://haystack.deepset.ai/overview/intro

Janssen, A., Passlick, J., Cardona, D. R., and Breitner, M. H. 2020. "Virtual Assistance in Any Context", Business & Information Systems Engineering, 62(3), 211-225.

Jiang, Z., Chi, C., and Zhan, Y. 2021. "Research on Medical Question Answering System Based on Knowledge Graph," IEEE Access, 9, 21094-21101.

John, B.M., Chua, A.Y., Goh, D.H.L., and Wickramasinghe, N. 2016. "Graph-based Cluster Analysis to Identify Similar Questions: A Design Science Approach," JAIS, 17(9), pp.2.

Khurana, S., Qiu, L., and Kumar, S. 2019. "When a Doctor Knows, It Shows: An Empirical Analysis of Doctors' Responses in a Q&A Forum of an Online Healthcare Portal," ISR, 30(3), pp. 872-891.

Lin, Y.L., Chung, C.Y., Kuo, C.W., and Chang, T.M. 2016. "Modeling Health Care Q&A Questions With Ensemble Classification Approaches," AMCIS 2016 Proceedings, 6.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., and Stoyanov, V. 2019. "Roberta: A Robustly Optimized Bert Pretraining Approach". arXiv:1907.11692.

Liu, X., Li, Y., Liu, F., Cai, Z., and Lim, E. 2019b. "Reinventing the Wheel: Explaining Question Duplication in Question Answering Communities," ICIS 2019 Proceedings, 27.

Mohamed, A., Parambath, S., Kaoudi, Z., and Aboulnaga, A. 2020. "Popularity Agnostic Evaluation of Knowledge Graph Embeddings," in Conference on Uncertainty in Artificial Intelligence, pp. 1059-1068.

Peng, C.-H., Yin, D., and Zhang, H 2020. "More than Words In Medical Question-And-Answer Sites: A content-context congruence perspective", ISR, 31, 3, pp. 913-928.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.H., and Riedel, S. 2019. "Language Models as Knowledge Bases?" arXiv:1909.01066.

Reimers, N. 2022. "Pretrained Models". https://www.sbert.net/docs/pretrained_models.html

Risch, J., Möller, T., Gutsch, J., and Pietsch, M. 2021. "Semantic Answer Similarity for Evaluating Question Answering Models," arXiv preprint arXiv:2108.06130.

Singh, K., Radhakrishna, A.S., Both, A., Shekarpour, S., Lytra, I., Usbeck, R., and Auer, S. 2018. "Why Reinvent the Wheel: Let's Build Question Answering Systems Together," WWW 2018 Proceedings.

Sun, Y., Shi, Q., Qi, L., and Zhang, Y. 2021. "JointLK: joint reasoning with language models and knowledge graphs for commonsense question answering," arxiv:2112.02732 (https://arxiv.org/abs/2112.02732).

Wambsganss, T., Haas, L., and Soellner, M. 2021. "Towards the Design of a Student-Centered Question-Answering System in Educational Settings," ECIS 2021 Proceedings, 7.

Wang, C., Liu, X., and Song, D. 2020. "Language Models are Open Knowledge Graphs". arXiv:2010.11967.

Williamson, P.O, and Minter, C.I.J. 2019. "Exploring PubMed as a Reliable Resource for Scholarly Communications Services," Journal of the Medical Library Association, 107(1), pp. 16-29.

Yasunaga, M., Ren, H., Bosselut, A., Liang, P., and Leskovec, J. 2021. "QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering," arXiv:2104.06378.

Zhang, L., Li, W., Bai, Q., and Lai, E. 2021. "Graph-based Self-Adaptive Conversational Agent," In Proceedings of the 20th AAMAS, pp. 1791-1793.

Zhou, L., Gao, J., Li, D., and Shum, H. Y. 2020. "The Design and Implementation of Xiaoice, An Empathetic Social Chatbot", Computational Linguistics, 46(1), 53-93.

Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S., and Chua, T.S. 2021. "Retrieving and Reading: A Comprehensive Survey on Open-domain Question Answering". arXiv:2101.00774.

Zhu, Z., Bernhard, D., and Gurevych, I. 2009. "A Multi-Dimensional Model for Assessing the Quality of Answers in Social Q&A Sites". In ICIQ, pp. 264-265.