Journal of the American Medical Informatics Association, 30(2), 2023, 329–339
https://doi.org/10.1093/jamia/ocac222
Advance Access Publication Date: 17 November 2022
Research and Applications

Research and Applications

Epidemic Question Answering: question generation and entailment for Answer Nugget discovery

Maxwell A. Weinzierl and Sanda M. Harabagiu

Human Language Technology Research Institute, Department of Computer Science, University of Texas at Dallas, Richardson, Texas, USA

Corresponding Author: Maxwell A. Weinzierl, BS, University of Texas at Dallas, P.O. Box 830688; MS EC31, Richardson TX 75080-0688, USA; maxwell.weinzierl@utdallas.edu

Received 20 July 2022; Revised 31 October 2022; Editorial Decision 2 November 2022; Accepted 3 November 2022

ABSTRACT

Objective: The rapidly growing body of communications during the COVID-19 pandemic posed a challenge to information seekers, who struggled to find answers to their specific and changing information needs. We designed a Question Answering (QA) system capable of answering ad-hoc questions about the COVID-19 disease, its causal virus SARS-CoV-2, and the recommended response to the pandemic.
Materials and Methods: The QA system incorporates, in addition to relevance models, automatic generation of questions from relevant sentences. We relied on entailment between questions for (1) pinpointing answers and (2) selecting *novel* answers early in the list of its results.
Results: The QA system produced state-of-the-art results when processing questions asked by experts (eg, researchers, scientists, or clinicians) and competitive results when processing questions asked by consumers of health information. Although state-of-the-art models for question generation and question entailment were used, more than half of the answers were missed, due to the limitations of the relevance models employed.
Discussion: Although question entailment enabled by automatic question generation is the cornerstone of our QA system's architecture, question entailment did not prove to always be reliable or sufficient in ranking the answers. Question entailment should be enhanced with additional inferential capabilities.
Conclusion: The QA system presented in this article produced state-of-the-art results processing expert questions and competitive results processing consumer questions. Improvements should be considered by using better relevance models and enhanced inference methods. Moreover, experts and consumers have different answer expectations, which should be accounted for in future QA development.

Key words: COVID-19, Question Answering, information retrieval, deep learning

INTRODUCTION

Finding specific information for the COVID-19 disease and its causal virus SARS-CoV-2 during the pandemic was difficult both for biomedical researchers and health professionals as well as for the general public. The emergence of the COVID-19 pandemic has given rise to a multitude of questions, ranging from the characteristics of the new virus to the prevention or the treatment of the infection. The rapidly growing and changing stream of publications made it hard for clinicians, researchers, patients, and policy makers to stay updated with respect to the health, economic, and social-cultural consequences of the pandemic. Although Information Retrieval systems (eg, search engines such as Google, Bing or PubMed) are now go-to tools for finding health information, they did not prove to be ideal solutions for searching relevant information in the fast-changing circumstances brought about by the pandemic.[1] Furthermore, search engines retrieved hundreds or thousands of documents which needed to be inspected to satisfy information needs.[2] In contrast, Question Answering (QA), a language technology, aims to

alleviate the problem of finding pertinent information across thousands of documents, as it pinpoints the answers responding to questions expressed in natural language from large collections of electronic documents.[3] Research in QA has a long history, spearheaded by the National Institute of Standards and Technology (NIST) Text Retrieval Conferences (TREC), initiating in 1999 a QA track which enabled the evaluation of tens of QA systems capable to answer open-domain questions for a decade.[4] While being a long-standing problem in natural language processing, open-domain QA has recently regained interest due to the DrQA system,[5] which used Wikipedia data for distant supervised learning of the extraction of answers. Dense retrieval methods further enabled state-of-the-art QA results when using deep contextual embeddings.[6,7] However, when answering domain-specific questions, it is well known that open-domain QA systems are not ideal.[8]

Health-specific QA systems capable to find answers in biomedical articles were developed successfully before, for example for assisting Clinical Decision Support (CDS) systems to find answers to clinicians' health-related questions. From 2015 to 2016, TREC organized a special track on Clinical Decision Support (TREC-CDS),[9] addressing the challenge of evaluating QA systems capable of retrieving biomedical articles relevant to medical case descriptions generated by consulting Electronic Medical Records from MIMIC-II.[10] The articles containing the answer to the questions were available from PubMed Central (PMC). The QA systems developed for the TREC-CDS challenge were meant to benefit expert clinicians.[11–17] Furthermore, Consumer Health Information Question Answering (CHiQA), an online specialized QA system,[18] was designed to help consumers find answers to their health-related questions.

In response to the COVID-19 pandemic, the 13th Text Analysis Conference (TAC), hosted by NIST, organized in 2020 an evaluation challenging research teams to develop QA systems capable of answering ad-hoc questions about the COVID-19 disease, its causal virus SARS-CoV-2, the related coronaviruses, as well as the recommended response to the pandemic. QA systems participating in this TAC challenge, called EPIdemiC-QA (EPIC-QA), were provided with a new dataset[1] which incorporates both public-facing and expert-level documents containing information about COVID-19. The QA designed for the participation in the EPIC-QA challenge had to process both Expert Questions (EQs) and Consumer Questions (CQs). Moreover, each question could have multiple answers. Each of the answers to a question were considered an Answer Nugget (AN).

OBJECTIVE

This article presents a study that addresses the question whether the same QA architecture may be used for answering questions asked by health professionals, for example EQs, as well as questions asked by the public, for example CQs by using the data available from the EPIC-QA challenge. In addition to relying on passage retrieval methods, this QA system took advantage of (1) automatically generated questions from relevant sentences and (2) question entailment implemented using deep learning representations of language. Interestingly, this QA system produced the best results in the challenge evaluations for the EQs, while still competitive results were obtained for the CQs. The analysis of the results explains the difference in performance between processing EQs and CQs.

MATERIALS AND METHODS

We first detail the tasks used in the EPIC-QA Challenge, briefly describing the datasets used in each task. This is followed by the presentation of our QA system designed for the EPIC-QA Challenge. We first describe the architecture of the QA system and then we detail its modules.

EPIC-QA Tasks: The EPIC-QA challenge involved 2 tasks:

- Task A: **Expert QA.** In this task, the QA systems processed 30 EQs. Examples of EQs are listed in Table 1a. Answers to EQs were searched through a collection of 236 034 biomedical articles from the document collection assembled for the COVID-19 Open Research Dataset Challenge (CORD-19). The CORD-19 dataset includes a subset of PMC as well as preprints from bioRxiv. In Task A, the answers were expected to provide information that is useful to researchers, scientists, or clinicians.
- Task B: **Consumer QA.** In this task, the QA systems processed 30 CQs to provide a ranked list of consumer-friendly answers. Examples of such CQs are listed in Table 1b. Answers to CQs were searched in a document collection which is a subset of articles used by the CHIQA service of the US National Library of Medicine (NLM).[18] This collection includes authoritative articles from the Centers of Disease Control and Prevention (CDC); the Generic and Rare Disease Information Center (GARD); the Genetics Home Reference (GHR), Medline Plus; the National Institute of Allergy and Infectious Diseases (NIAID); and the World Health Organization (WHO). In addition, the collection contained 256 Reddit threads from/*r/askscience* tagged with COVID-19, Medicine, Biology, or Human Body and filtered for COVID-19 questions asked by consumers. The collection also included a subset of the CommonCrawl News crawl from Janu-

Table 1. Examples of (a) Expert Questions and (b) Consumer Questions

How do cytokine pathways link sleep and immunity to infection and COVID-19?
What endocrine complications are linked to COVID-19?
Is the association between COVID-19 and diabetes driven by the dpp4 receptor?
Which interleukins and IL-inhibitors are involved in COVID-19 pathways?
How do mutations in SARS-CoV-2 impact its infectivity and antigenicity?
What computational predictions of SARS-CoV-2 mutations have been confirmed?
How are telehealth services used during the Covid-19 pandemic and what is their impact on the population health?
(a)
How long do COVID-19 antibodies stay in your system?
What anti-diabetic medications are the safest during the COVID-19 pandemic?
Why is COVID-19 more severe the second time?
Could a person's DNA explain why some get hit hard by the coronavirus?
What are recommendations and advice in case of COVID-19 resurgence?
How to build my child's social skills and prevent psychological harm during COVID-19?
Can science predict how coronavirus will change?
(b)

ary 1 to April 30, 2020, as used in the TREC Health Misinformation Track.[19]

In both EPIC-QA tasks, the answers returned by the QA systems were expected to be in the form of consecutive sentences extracted from a single *context* of a single document. Generally, contexts correspond to paragraphs defined by authors in their publications or by HTML sections on Web pages. Contexts do not contain more than 15 sentences. For Task A, participants in the EPIC-QA challenge were provided with 4 075 478 contexts, while for Task B, 430 876 contexts were provided.

Both EQs and CQs are complex questions, answered by potentially *multiple* ANs. The ANs are text snippets contained in sentences from expert or consumer contexts. Figure 1 illustrates 3 ANs responding to an EQ and 3 ANs responding to a CQ. Some sentences from a context may contain multiple ANs, while other sentences may contain no AN. The QA systems must provide a ranked list of sentences which contain ANs to a question, aiming to contain *novel* ANs earlier in the list. ANs were considered novel if not observed in higher-ranked sentences.

A question answering system for answer nugget identification

Although the EPIC-QA tasks were essentially health-specific QA tasks, they were different because (1) each operated on a different dataset and (2) the questions and answers are expected to be appropriate for 2 very different set of users: biomedical experts and general public. Nevertheless, we decided to develop a single QA system that handled both tasks, as illustrated in Figure 2. Given a question Q, a relevance model first ranks the list of contexts where the ANs for Q may be found. The relevance model relies either on a Research Index (RI) or a Consumer Index (CI). Both the RI and the CI were built by using the open-source Apache Lucene search engine library.[20] The relevance model implements the Okapi BM25 relevance model,[21] widely used in Information Retrieval.

While the BM25 ranking orders contexts by their relevance to a question, it is not sufficient for ranking sentences that may contain the question's ANs. Because the organizers of EPIC-QA have provided all track participants examples of questions and their annotated ANs, we utilized a BERT-Reranking module to *learn* how to rank the candidate sentences that may contain ANs. More specifically, BERT-Reranking estimates a reranking score r_i quantifying the relevance of the sentence s_i, potentially containing ANs for question Q. This is achieved by using the pre-trained BERT model[22] and fine-tuning it to the reranking task using a weighted cross-entropy loss:

$$\mathcal{L} = \sum_{j \in J_{POS}} w_j \times \log(r_j) - \sum_{j \in J_{NEG}} \log(1 - r_j) \qquad (1)$$

where J_{POS} represents the set of sentences known to contain ANs to the question Q, while J_{NEG} are sentences randomly sampled from the top 1000 sentences contained in the contexts ranked by BM25. The weight w_j quantifies the number of ANs in a sentence s_j, prioritizing the reranking of sentences containing a larger number of ANs. This reranking method was inspired by Nogueira and Cho,[23] where a cross-entropy loss, similar to the one from Equation (1), was used for learning the ranking of paragraphs, which resulted in improved relevance on a QA task operating on Wikipedia articles.

The reranking of sentences can further benefit from textual inference, for example *textual entailment*, to indicate the presence of ANs. We explored entailment between question Q and questions derived from the reranked sentences. Our intuition was that: (1) automatically generated questions have known answers and (2) if those questions are entailed by Q, their answers must also answer Q. Moreover, when entailment between Q and a generated question gq_i is established, a node representing gq_i is assigned in the Question Entailment Graph (QEG). Edges between any pair of nodes from the QEG are determined when entailment between the questions corresponding to the nodes is recognized.

In the design of the QA system illustrated in Figure 2, we believed that the final ranking of sentences returned by the QA system should maximize the evaluation metric of the challenge, namely the Normalized Discount Novelty Score (NDNS). To do so, we used another intuition that stipulates that only some of the automatically

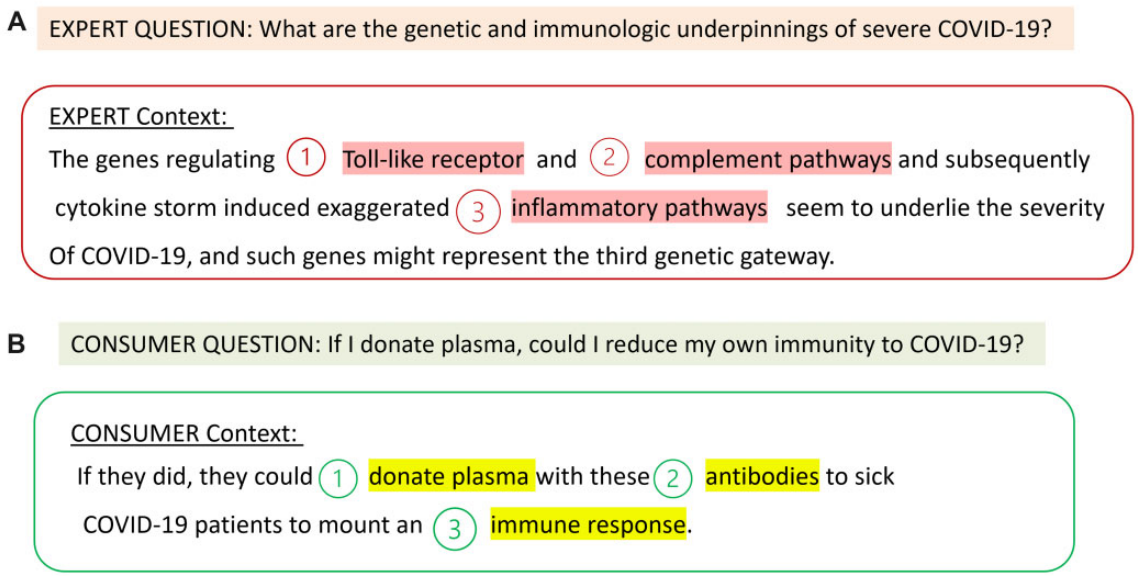

Figure 1. (A) Example of expert question and an expert context containing some of its Answer Nuggets; (B) example of Consumer Question and a consumer context containing some of its Answer Nuggets.

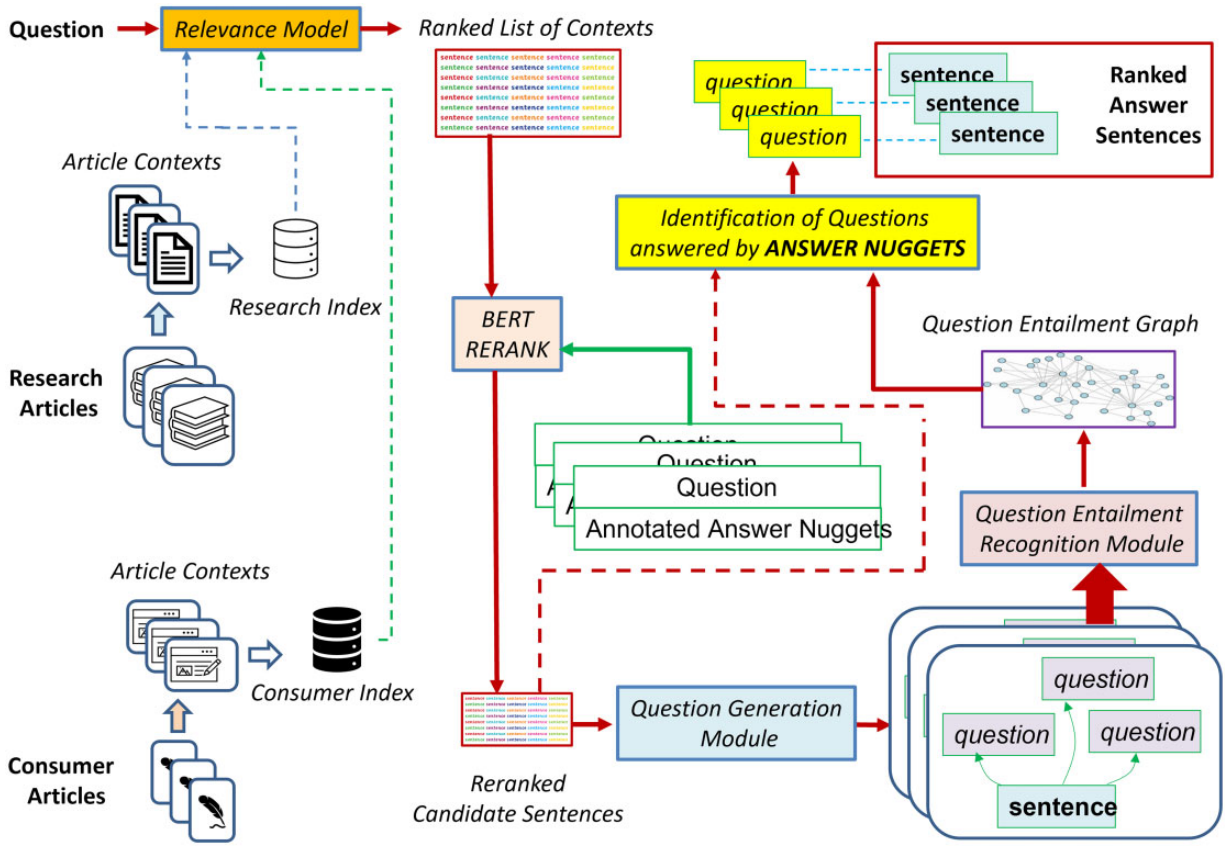

Figure 2. Architecture of a Question Answering system used for identifying Answer Nuggets for questions used in both tasks of the EPIC-QA evaluations.

generated questions are answered by the ANs of question Q. The discovery of these questions was made possible by the QEG. Therefore, essential to the operation of our QA system were (1) the question generation module and (2) the recognition of question entailment, which informed the final ranking of the sentences (FRS) containing ANs, as detailed below.

The question generation module

The generation of questions from each of the reranked candidate sentences was performed using the docTTTTTquery system,[24] which uses the sequence-to-sequence T5 model[25] for learning to predict questions from any text passage. The docTTTTTquery system was trained on the Microsoft Machine Reading Comprehension (MS MARCO) dataset[26] using 500 000 query–passage pairs, enabling it to learn a model able to predict which questions can be generated from a text passage. The model for generating questions from any text was trained for 4000 iterations, using batches of 256 MS MARCO passages.[24] When using this question generation model, we produced for each candidate reranked sentence questions similar to those illustrated in Figure 3.

However, not all questions generated automatically from the reranked candidate sentences are asking about information that may contain ANs for the original question Q processed by the QA architecture illustrated in Figure 2. To address this limitation, we have considered the automatic recognition of question entailment relations, which can be established between automatically generated questions GQ_i and the original question Q.

Recognizing question entailment

We assumed that entailment should be observed between the generated questions GQ_i that are answered by the ANs of question Q and question Q itself. To recognize entailment in the EPIC-QA tasks, we relied on the Question Entailment Recognition Module (QERM), illustrated in Figure 2, capable to identify entailment relations between pairs of questions Q_A and Q_B. Entailment relations between 2 questions are viewed as a logical inference performed on the text of the questions, in terms of possible worlds (or interpretations). If Q_A is the premise, while Q_B is the hypothesis, then Q_A is true in all the worlds where Q_B is true, which is resolved through neural textual entailment.[27] We considered 2 possible implementations of the QERM, illustrated in Figure 4, both of them fine-tuned on the Quora Question Duplication Detection dataset.[28]

In both implementations, Word Piece Tokenization (WPT) is applied to a question Q_A and a question Q_B. The tokens of each question along with a start token [CLS] and a separator token [SEP] placed between the tokens of the 2 questions and after the tokens of question Q_B are provided to either of the 2 language models used in the QERM. The first implementation, illustrated in Figure 4A, uses the BERT language model[22] to create deep contextual representations of Q_A and Q_B. The output of BERT, a single contextual embedding, corresponding to the [CLS] token, which is passed through a single-layer Feed Forward Neural Network (FFN) to predict either an *Entailed* or a *Not Entailed* relationship between Q_A and Q_B with a probability provided by the softmax function implementing the last stage of the entailment classifier.

The second implementation of the QERM, illustrated in Figure 4B, relies on the intuition that questions sharing ANs must

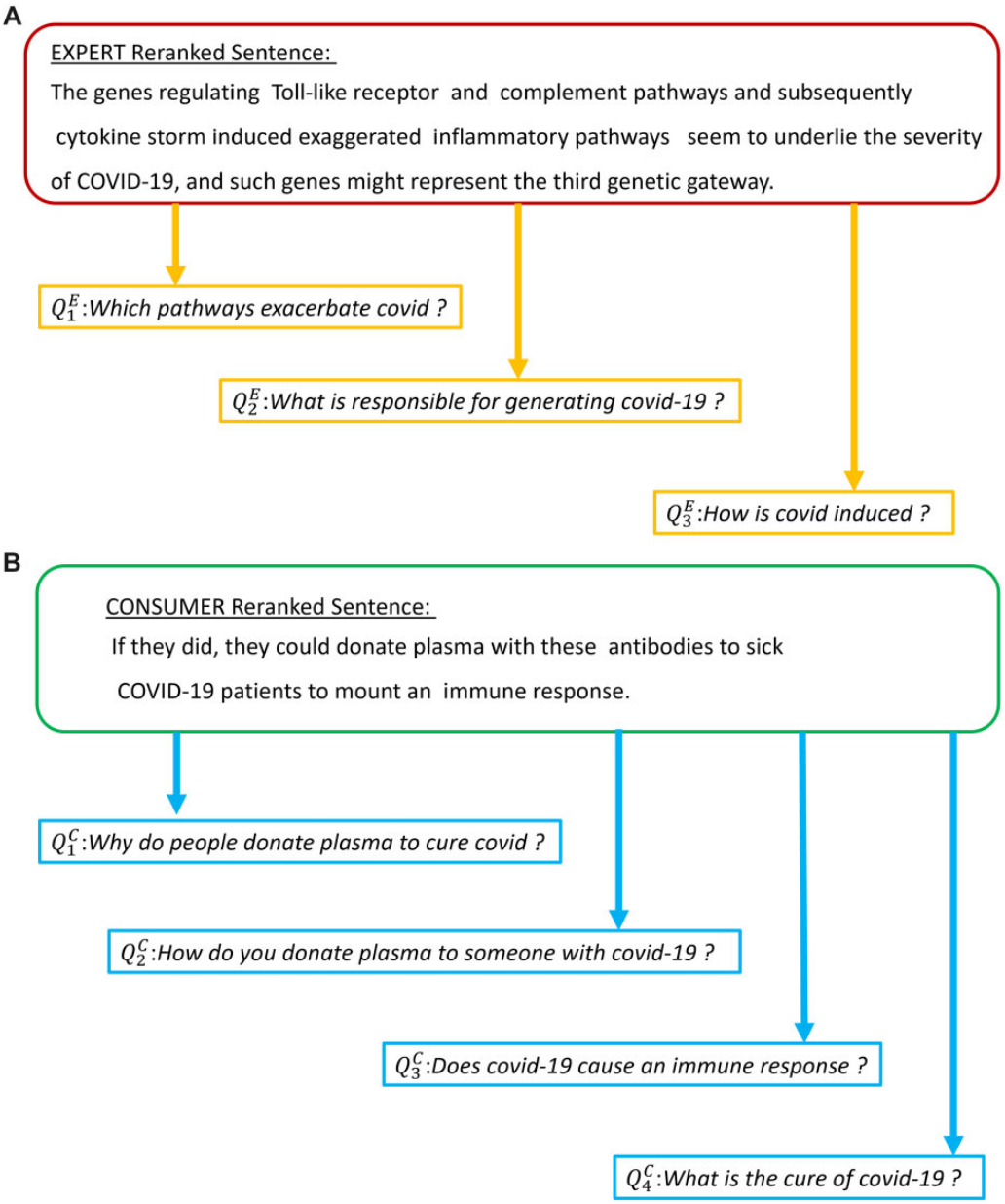

Figure 3. Example of questions automatically generated from the reranked candidate sentences. (A) Illustrates 3 questions automatically generated from a sentence showcased as an Expert Passage in Figure 1A, where the ANs are highlighted. We note that Q_1^E is answered by all 3 ANs shown in Figure 1A. Interestingly, Q_2^E is not answered by any of the ANs. Question Q_2^E articulates a causal explanation request that has no correct answer in the sentence, which uses the verb "induced" to establish the causal relation between the cytokine storm and the inflammatory pathways. (B) Illustrates 4 questions automatically generated from the sentence showcased as a Consumer Passage in Figure 1B. Interestingly, question Q_1^C is answered by the text snippet "*to mount an immune response*" covering the third answer snippet shown in Figure 1B. Question Q_2^C has no correct answer in the sentence, but the event reference "if they did" refers to information from preceding sentences in the Consumer Passage, which might contain the answer. Question Q_3^C is answered by the text snippet "plasma with these antibodies", which contains the second AN of the CQ illustrated in Figure 1B. Finally, question Q_4^C is correctly answered by the text snippet "*donate plasma with these antibodies to sick COVID-19 patients to mount an immune response*", which contains all 3 ANs of the CQ illustrated in Figure 1B.

also be involved in an entailment relation. Consequently, we have replaced the BERT language model with the MS MARCO-BioBERT-RERANK model, which was trained on question–answer pairs from the dataset provided in MS MARCO.[23] MS MARCO includes 100 000 questions sourced from real, anonymized queries posed on BING or CORTANA, and their answers judged by human crowdsourcing. This output of this language model is passed through a relevancy classifier layer to obtain a relevance score.

In addition to discovering the entailment relations between an original question Q and any of the automatically generated question GQ_i, both implementation of the QERM were also used to identify entailment between pairs of generated questions GQ_i and GQ_j. This allowed us to produce on the fly, for each original question Q, a QEG, similar to the one illustrated in Figure 5. Given the set of reranked sentences $s_1, \ldots, s_N$ resulting from BERT-Reranking, the corresponding automatically generated questions become nodes

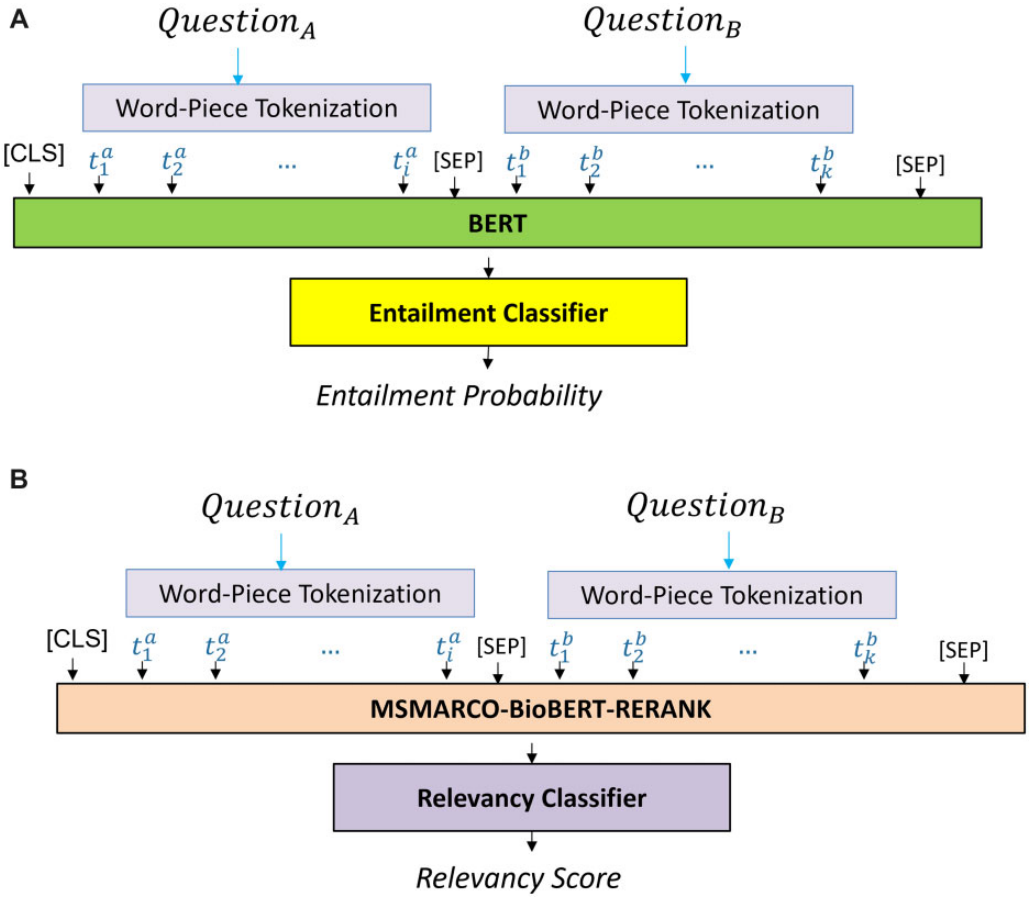

Figure 4. Implementations of the Question Entailment Recognition Module (QERM).

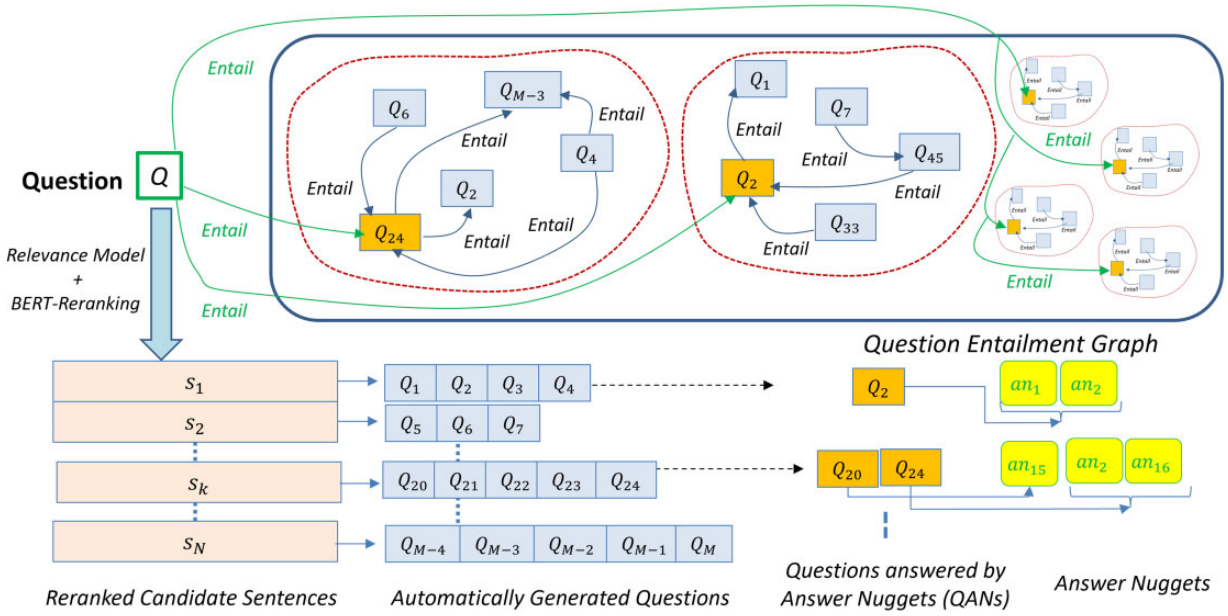

Figure 5. Using the Question Entailment Graph (QEG) for Recognizing Automatically Generated Questions that ask about the Answer Nuggets of a Question Q. Figure 5 shows how from s_1 4 questions were generated, but only q_2 was a QAN, answered by ANs an_1 and an_2. Although 3 questions were generated from s_2, none of them were QANs, and hence s_2 is a sentence with no ANs. However, in a lower-ranked sentence, s_k, 2 QANs were identified via the QEG, namely q_{20} and q_{24}. Some of the ANs may not be novel, as they have also been observed in other sentences, for example an_2 is observed both in s_1 and s_k. Consequently an_2 is considered a *novel* AN only in s_1.

in the QEG if they were participating in at least one entailment relation.

As shown in Figure 5, the entailment relations create connected components in the QEG, highlighted by dashed red lines. Within these connected components, some questions share 2 important properties: (1) they are entailed by the original question Q and (2) they are the most connected questions in a connected component of the QEG. These questions are colored in orange in Figure 5. We hypothesize that these questions are answered by the ANs of the original question Q because (1) they are entailed by Q, therefore their answers are also answers to Q; and (2) they are either entailed by or entail most of the other questions from the QEG component, therefore they share most answers with other related questions. Knowledge about Questions that are answered by the Answer Nuggets (QANs) informs the FRS returned by the EPIC-QA system for question Q.

Final ranking of sentences containing answer nuggets

The identification of QANs in the QEG allowed us to keep track, as shown in Figure 5, not only of the automatically generated questions GQ_j^i for each sentence s_i, but also of the ANs contained in the sentence and the corresponding QANs they answer.

In our experiments, we found that EQs were answered by an average number of 25.0 ANs per question, while CQs were answered by an average number of 13.9 ANs per question. Not only did the FRS returned by the QA system for each question Q need to cover all ANs recognized in the candidate sentences, but it also promoted the novel ANs at the top of ranked sentences. The FRS was obtained by generating an optimal value for the evaluation metric used in the EPIC-QA challenge, namely the modified Normalized Discount Cumulative Gain (NDCG), an evaluation metric used for assessing the ranking quality of search engines.[29] NDCG evaluates the usefulness, or the *gain*, of a sentence as_i (which is at rank i) in providing *novel* ANs for a question Q. The modification of NDCG that was used in the EPIC-QA challenge was the NDNS which computes the gain of adding sentence as_i to FRS through a Novelty Score for as_i, provided by:

$$NS_i = \frac{nan_i \times (nan_i + 1)}{nan_i + SF_i} \tag{2}$$

where nan_i represents the number of novel ANs observed in as_i, while SF_i is a sentence factor which penalizes the novelty score. However, in the FRS, there are 3 types of sentences: (1) sentences containing novel ANs (eg, s_1 or s_k from Figure 5)—we shall denote the number of such sentences as *nsnn*; (2) sentences containing only ANs that have been already seen—we denote the number of such sentences as *nssn*; and (3) sentences containing zero ANs (eg, s_2 from Figure 5)—we denote the number of such sentences as *nszn*. These 3 numbers inform 3 different formulations of SF_i, which leads to 3 different variants of the NDNS evaluation metric:

- NDNS-Relaxed, where answer sentences should contain only novel ANs, defining SF_i by:

$$SF_i = \min(nsnn, 1) + nssn + nszn$$

- NDNS-Partial, where answer sentences that contain no ANs are discarded, defining SF_i by:

$$SF_i = \min(nsnn, 1) + nssn$$

- NDNS-exact, which prefers a shorter FRS, defining SF_i by:

$$SF_i = nsnn + nssn + nszn$$

An optimal FRS was obtained by maximizing the NDNS score for the entire FRS, computed as:

$$NDNS(FRS) = \sum_{p=1}^{|FRS|} \frac{DCG(p)}{IDCG(p)} \tag{3}$$

where DCG_p, the Discounted Cumulative Gain at position p in the FRS is computed as:

$$-DCG_p = \sum_{i=1}^{p} \frac{NS_i}{log_2(i+1)} \tag{4}$$

while $IDCG_p$, the Ideal DCG_p, is computed as:

$$IDCG_p = \sum_{i=1}^{A_p} \frac{NS_i}{log_2(i+1)} \tag{5}$$

with A_p representing the list of sentences which contain novel ANs, ordered in descending order of the number of novel ANs they contain.

Maximizing the NDNS, and therefore finding the optimal FRS, requires knowing which ANs are present in each answer. The organizers of EPIC-QA chose to use a beam search,[30] with the number of beams being 10, over various FRS, maximizing the value of DCG_p to guide the search. Beam search is a heuristic-based or online search strategy which considers a limited (in this case 10) best successors of all nodes that can be expanded during search.[31] The ideal FRS produced the maximum value for NDNS over this search. Because we did not know which ANs are present in each sentence, we decided to use the QANs as replacement for ANs, allowing us to use the same beam search over all possible FRS.

We have evaluated 3 versions of the QA system: (1) a version that uses only the BM25 ranking; (2) a version that also employs the BERT-Reranking module; and (3) the version that adds the information from the QEG, using the entire architecture illustrated in Figure 2. Before using the EPIC-QA data, the IRB board at UT Dallas approved our research as meeting the criteria for IRB exemption. Code is made publicly available at the following GitHub repository: https://github.com/Supermaxman/epic_qa

RESULTS

The evaluation results obtained with our QA system in EPIC-QA are listed in Table 2, with best results in bold, where Human Language Technology Research Institute (HLTRI) indicates our QA system. Table 2 also lists the results of the other QA systems participating in the EPIC-QA challenge,[1] where description of all systems are provided. The results from Table 2a indicate that our full QA system which incorporates the QEG is generating promising results, as it obtained the best results when processing EQs. It scored the best values across all variants of NDNS. Table 2b shows that our system obtained competitive results when processing CQs, while not obtaining the best performance. However, the BERT-Reranking helped in placing the result for the CQs evaluation as second-best. We examine the possible reasons for this drop in performance in processing CQs in the Discussion section.

We also considered as evaluation metric the NDCG, which ignores the novelty of ANs, counting all ANs in the FRS. The results listed in Table 2 indicate that BERT-Reranking provides the best NDCG results for our system when processing either EQs or CQs, which is not surprising, as the QEG role is to prioritize novel ANs.

Table 2. Question Answering performance on (a) the EPIC-QA Expert Task and (b) the EPIC-QA Consumer Task

Team and system	NDNS-relaxed	NDNS-partial	NDNS-exact	NDCG
UPC_USMBA Best Run	0.127	0.126	0.148	–
nlm_lhc_qa Best Run	0.219	0.223	0.209	–
IBM Best Run	0.329	0.331	0.367	–
h2oloo Best Run	0.344	0.344	0.390	–
vigicovid Best Run	0.344	0.345	0.391	–
Yastil_R Best Run	0.362	0.361	0.410	–
HLTRI Unsubmitted Run: BM25	0.236	0.255	0.199	0.068
HLTRI Run 2: BM25 + BERT-Reranking	0.364	0.363	0.413	**0.074**
HLTRI Run 3: BM25 + BERT-Reranking + QEG	**0.371**	**0.370**	**0.421**	0.073

(a)

System	NDNS-relaxed	NDNS-partial	NDNS-exact	NDCG
UPC_USMBA Best Run	0.172	0.176	0.175	–
nlm_lhc_qa Best Run	0.184	0.186	0.183	–
IBM Best Run	0.264	0.268	0.282	–
h2oloo Best Run	**0.368**	**0.366**	**0.414**	–
HLTRI Unsubmitted Run: BM25	0.138	0.147	0.114	0.039
HLTRI Run 2: BM25 + BERT-Reranking	0.313	0.312	0.353	**0.046**
HLTRI Run 3: BM25 + BERT-Reranking + QEG	0.317	0.316	0.363	0.044

(b)

Furthermore, we were interested to evaluate the performance of the QERM, which was instrumental in generating the QEG. Table 3 presents the results of 3 baselines reported in prior work against the results obtained by the 2 implementations of the QERM. Not surprisingly, all systems using neural networks outperform the system using logistic regression. The QERM implementation illustrated in Figure 4A, using BERT,[22] outperformed all prior work, while QERM-RERANK, which represents the QERM implementation illustrated in Figure 4B, achieved state-of-the-art performance with an accuracy of 89.55. These results indicate that the QA system presented in this paper relied on state-of-the-art question entailment.

Table 3. Question entailment recognition on the Quora Question Duplication dataset

System	Accuracy
Logistic Regression[32]	67.79
Neural Network[33]	81.34
Neural Network + GloVe Embeddings[34]	83.62
QERM-BERT	88.94
QERM-RERANK	**89.55**

DISCUSSION

We conducted a detailed quantitative and qualitative analysis of the errors made by the QA system when processing EQs and CQs. We first assessed the number of unique ANs present in the top-500 contexts retrieved for each question by BM25, to find out how many ANs were missed, and thus could not contribute to the FRS. On average, BM25 found 73.6% of ANs across the candidate answers retrieved for the EQs, with a minimum of 23.1% ANs discovered when processing the EQ: "*What approaches are recommended for developing children's social and emotional coping skills during the COVID-19 pandemic?*", and a maximum of 100% ANs discovered when processing 5 of the EQs. On average, BM25 also found 77.3% of ANs across candidate answers retrieved for the CQs, with a minimum of 6.3% ANs discovered for the CQ: "*How long after I feel better from COVID-19 can I go back to work?*", and a maximum of 100% ANs discovered when processing 9 of the CQs. Missing 26.4% of EQ ANs and 22.7% of CQ ANs in the entire search for relevant contexts clearly restricts the possible performance by the BERT-Reranking and QEG systems, as they can never provide answers containing those unique ANs.

When analyzing the number of unique ANs present in the top-100 candidate sentences produced by the BERT-Reranking module, we found that only 40.9% of ANs were present across the top-100 reranked candidate sentences for the EQs and 42.7% of ANs were present across top-100 reranked candidate sentences for the CQs. However, for the EQ "*Is the association between COVID-19 and diabetes driven by the dpp4 receptor?*" we found all the ANs in the reranked candidate sentences. Similarly, for the CQ "*How to build my child's social skills and prevent psychological harm during COVID-19?*" we found all the ANs in the reranked candidate sentences. Therefore, the BERT-Reranking module missed out 59.1% of EQ ANs and 57.3% of CQ ANs, in large part due to (1) the BM25 function that did not rank sufficiently high sentences containing all the ANs and (2) its selection of 100 sentences from the 500 contexts provided by the BM25 ranking, missing out on an additional 32.7% of EQ ANs and 34.6% of CQ ANs.

We were further interested in the percentage of ANs present in the candidate sentences produced by the BERT-Reranking module. We found that only 40.9% of ANs for EQs were present in these sentences and 42.7% of ANs for CQs were present. Therefore, this analysis indicates that the quality of the retrieval of candidate contexts combined with the selection of the candidate sentences is responsible for missing out more than half of the ANs. This is not surprising, because the quality of retrieval is still the main barrier for QA systems.[35] Given that the novelty of the ANs was crucial in the evaluation of the QA systems, we analyzed how well the QEG informed an optimal FRS. The QEG informed the FRS better for the

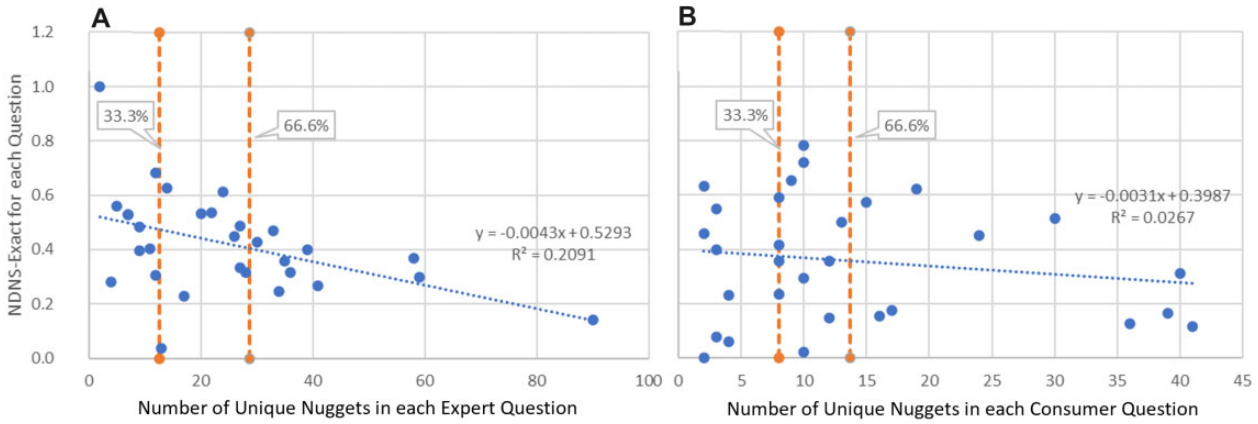

Figure 6. Performance of the Question Entailment Graph (QEG) informing the Final Ranking of Sentences (FRS) for the (A) Expert task and (B) Consumer task in EPIC-QA when compared to the number of unique Answer Nuggets judged to be present in each question.

Expert QA task than for the Consumer QA task. The average NDNS-Exact value of the FRS for the Consumer QA task was 0.363, with significant variance in performance for each question with a standard deviation of 0.221. The FRS the CQ: "*Can science predict how coronavirus will change?*" obtained the minimum NDNS-Exact value of 0.0, while for the CQ: "*What roles do interleukins and IL-inhibitors play in COVID-19?*" the FRS achieved a maximum NDNS-Exact value of 0.783.

We further analyzed the distribution of NDNS-Exact across EQs and CQs and compared these scores to the number of unique nuggets identified by annotators in the EPIC-QA task. The EQs had an average of 25 unique ANs for each question, while the CQs had an average of 14 unique ANs for each question. NDNS-Exact performance on each question is compared to the number of unique ANs judged to belong to each question, identifying one of the key differences in the tasks: The FRS informed by the QEG is highly dependent on the number of unique ANs judged to belong to each question. Figure 6 shows the distribution of NDNS-Exact across various numbers of ANs corresponding to a question. We further split the possible number of ANs into 3 tiers: Tier 1—corresponding to questions answered by the bottom 33.3% percentile of number of ANs; Tier 2—corresponding to questions answered by the 33.3–66.6% percentile of number of ANs; and Tier 3—corresponding to questions answered by the 66.6% and up percentile of number of ANs. Our QA system scored a NDNS-Exact of 0.517 for the bottom 33.3% of questions in the Expert task, 0.416 for the middle 33.3%, and 0.330 for the top 33.3%. The QEG module scored a NDNS-Exact of 0.300 for questions from Tier 1 in the Consumer task, 0.422 for questions from Tier 2, and 0.321 for the questions from Tier 3. Additionally, when the NDNS-Exact performance is only considered for questions with 25 ± 8 unique ANs (approximately the middle 20% of ANs in the Expert task), the performance of the QA system is equalized across EQs and CQs, with an Expert NDNS-Exact of 0.439 and a Consumer NDNS-Exact of 0.440. These differences in performance can therefore be partially attributed to the varying in numbers of ANs across the EQs and CQs, therefore performance of the QA system across tasks is similar only when answering questions having similar number of ANs.

Interestingly, we noticed that our QA system exceled when processing EQs with few unique ANs, while it performed poorly when processing CQs with few ANs, as is illustrated in Figure 6. Qualitatively, this indicates that the QEG is not always providing reliable information to the QA system. We inspected the QA system's processing of the following CQ: "*Why is COVID-19 more severe the second time?*" This question has been assigned 10 unique ANs: "Antibody Testing", "Vaccine Creation", "Vaccine Distribution", "Seasonality", "Influenza", "Contagious", "Reinfection", "Covid-19 Research", "Covid Protocols", and "Herd Immunity". However, in the QEG, we found that nearly all the generated questions were connected into a single QAN: "What are COVID-19 symptoms?" This is due to a slight topic shift across entailed questions: first generated question was "What are symptoms of severe COVID-19", a second generated question was "Is COVID-19 as serious a concern during the summer?", and a third generated question was "What are serious concerns regarding COVID-19?" Clearly the first question and the second question do not entail each other and should be considered separately in terms of representing a QAN. But, in this case, the third question entails the first question, and the third question entails the second question, leading to all 3 questions ending up in the same QAN. This is the primary reason why the QEG did not provide reliable information: the transitivity of the entailment relation does not hold in general, and when this is ignored, it can lead to a significant shift in the entailed questions within a QAN.

Moreover, the larger number of ANs led to an FRS with lower values for NDNS-Exact across both the Expert and Consumer tasks, as illustrated in Figure 6, indicating that the QEG did not provide sufficient information. In fact, when analyzing the QEGs, we noticed 2 issues: (1) the automatically generated questions did not answer all the ANs present in the candidate sentences; and (2) sometimes unrelated questions are represented in the same QEG, misinforming the FRS. These findings indicate that new question generation methods need to be developed such that they can address all question ANs, and question entailment needs to be augmented by additional inferential capabilities to improve the quality of QEGs.

CONCLUSION

Automatic question generation from sentences that may contain ANs enables the recognition of textual entailment relations and the creation of a QEG. The QEG plays an important role in identifying novel ANs and in producing a ranked list of sentences that contain early on the ANs of a question. The QA architecture that relies on a QEG produced state-of-the-art results when answering EQs and competitive results when answering consumers' questions about

COVID-19. Further improvements in the discovery of ANs can be achieved by contemplating additional inferential capabilities between the automatically generated questions.

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

AUTHOR CONTRIBUTIONS

MAW and SMH drafted the manuscript. MAW designed the QA system with significant discussions with and contributions from SMH. MAW carried out the experiments and contributed to data collection and analysis. MAW and SMH edited and provided feedback on all drafts.

CONFLICT OF INTEREST STATEMENT

None declared.

DATA AVAILABILITY

The EPIC-QA shared-task data underlying this article was provided by TAC, hosted by NIST, with both the Expert and Consumer collections publicly available to download: https://bionlp.nlm.nih.gov/epic_qa/#collection. The Expert task collection is provided under the **Open COVID Pledge** compatible **Dataset License**, with additional licensing information available on the website. The Consumer task collection is provided without copyright due to being works produced by the federal government, with additional licensing information available on the website. Additionally, we utilized the Quora Question Duplication Detection dataset, which is licensed as subject to Quora's Terms of Service, allowing for non-commercial use, and is publicly available to download: https://www.kaggle.com/c/quora-question-pairs.

REFERENCES

1. Goodwin TR, Demner-Fushman D, Lo K, *et al.* Automatic question answering for multiple stakeholders, the epidemic question answering dataset. *Sci Data* 2022; 9 (1): 432.
2. Manning CD, Raghavan P, Schütze H. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press; 2005.
3. Strzalkowski T, Harabagiu SM. *Advances in Open Domain Question Answering (Text, Speech and Language Technology)*. Dordretch: Springer; 2006.
4. Voorhees EM, Tice DM. The TREC-8 question answering track evaluation. In: Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00); 2000.
5. Chen D, Fisch A, Westin J, Bodes A. Reading Wikipedia to answer open-domain questions. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL-2017); 2017: 1870–9.
6. Lee K, Chang M-W, Toutanova K. Latent retrieval for weakly supervised open domain question answering [published online ahead of print 2019]. *ArXiv*. abs/1906.00300.
7. Karpukhin V, Oguz B, Min S, *et al.* Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics; 2020; 6769–81; online.
8. Mollá D, Vicedo JL. Question answering in restricted domains: an overview. *Comput Linguist* 2007; 33 (1): 41–61.
9. Clinical Decision Support Track. Text Retrieval Conference (TREC). 2016. https://www.trec-cds.org/2016.html. Accessed February 1, 2020.
10. Lee J, Scott DJ, Villarroel M, Clifford GD, Saeed M, Mark RG. Open-access MIMIC-II database for intensive care research. *Annu Int Conf IEEE Eng Med Biol Soc* 2011; 2011: 8315–8.
11. McNamee P. A domain independent approach to clinical decision support. In: The Twenty-Fourth Text REtrieval Conference Proceedings (TREC-2015); 2015.
12. D'hondt EKL, Grau B, Zweigenbaum P. LIMSI @ 2015 clinical decision support track. In: The Twenty-Fourth Text REtreival Conference Proceedings (TREC-2015); 2015.
13. Hasan SA, Ling Y, Liu J, Farri O. Using neural embeddings for diagnostic inferencing in clinical question answering. In: The Twenty-Fourth Text REtrieval Conference Proceedings (TREC-2015); 2015.
14. Wang Y, Rastagar-Mojarad M, Komandur-Elayavili R, Liu S, Liu H. An ensemble model of clinical information extraction and information retrieval for clinical decision support. In: The Twenty-Fifth Text REtrieval Conference Proceedings (TREC-2016); 2016.
15. Chen W, Moosavinasab S, Rust S, Huang Y, Lin S. Evaluation of a machine learning method to rank PubMed Central articles for clinical relevancy: NCH at TREC 2016 clinical decision support track. In: The Twenty-Sixth Text REtrieval Conference Proceedings (TREC-2016); 2016.
16. Abacha AB. NLM NIH at TREC 2016 clinical decision support track. In: The Twenty-Fifth Text REtrieval Conference Proceedings (TREC-2016); 2016.
17. Goodwin TR, Harabagiu SM. Learning relevance models for patient cohort retrieval. *JAMIA Open* 2018; 1 (2): 265–75.
18. Demner-Fushman D, Mrabet Y, Abacha AB. Consumer health information and question answering: helping consumers find answers to their health-related information needs. *J Am Med Inform Assoc* 2020; 27 (2): 194–201.
19. Charles LA, Clark SR, Mark D, Smucker M, Maistro, G, Zuccon. Overview of the TREC 2020 health misinformation track. In: proceedings of the twenty-ninth text REtrieval conference, TREC-2020, vol. 1266. 2020; online.
20. Sharma A. Practical Apache Lucene 8: Uncover the Search Capabilities of Your Application. New York: APress; 2020.
21. Robertson SE, Zaragoza H. The probabilistic relevance framework: BM25 and beyond. *FNT in Information Retrieval* 2009; 3 (4): 333–89.
22. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: proceedings of NAACL-HLT; 2019: 16.
23. Nogueira R, Cho K. Passage re-ranking with BERT [published online ahead of print 2019]. ArXiv. abs/1901.04085.
24. Nogueira R, Yang W, Lin J, Cho K. Document expansion by query prediction [published online ahead of print 2019]. CoRR. abs/1904.8375.
25. Raffel C, Shazeer N, Roberts A, *et al.* Exploring the limits of transfer learning with a unified text-to-text transformer. *J Mach Learn Res* 2020; 21 (140): 1–67.
26. Nguyen T, Rosenberg M, Song X, *et al.* MS MARCO: A Human Generated MAchine Reading COmprehension Dataset [published online ahead of print 2016]. ArXiv. abs/1611.09268.
27. Kang D, Khot T, Sabharwal A, Clark P. Bridging knowledge gaps in neural entailment via symbolic models. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP-2018); 2018: 4940–5.
28. Quora Question Duplication Detection Dataset. https://www.kaggle.com/c/quora-question-pairs. Accessed February 1, 2020.
29. Wang Y, Wang L, Li Y, Liu T-Y. A theoretical analysis of normalized discounted cumulative gain (NDCG) ranking measures. 2013.
30. Meister C, Cotterell R, Vieira T. Best-first beam search. *Trans Assoc Comput Linguist* 2020; 8: 795–809.
31. Russell SJ, Norvig P. Artificial intelligence—a modern approach. 2nd ed. In: Prentice Hall Series in Artificial Intelligence. 2003.

32. Hosmer DW Jr, Rodney SL, Sturdivant X. *Applied Logistic Regression*; *Wiley Series in Probability and Statistics*. Hoboken, NJ: John Wiley & Sons; 2013.
33. Schmidhuber J. Deep learning in neural networks: an overview. *Neural Netw* 2015; 61: 85–117.
34. Pennington J, Socher R, Manning CD. GloVe: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP-2014); 2014: 1532–43.
35. Clark P, Etzioni O, Khot T, *et al*. Combining retrieval, statistics, and inference to answer elementary science questions. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16); 2016: 2580–6.