ISSN: 2474-9206

Research Article

Journal of Anesthesia & Pain Medicine

Large-Scale Knowledge Synthesis and Complex Information Retrieval from Biomedical Documents

Vishal Vaddina*, Shreya Saxena, Raj Sangani, Siva Prasad, Shubham Kumar, Mihir Athale, Rohan Awhad.

Quantiphi Inc.

***Corresponding Author**
Mr. Vishal Vaddina, Quantiphi Inc.

Submitted: 2023, Mar 09**; Accepted:** 2023, Mar 26**; Published:** 2023, May 16

Citation: Vaddina, V., Saxena, S., Sangani, R., Prasad, S., Kumar, S., et al. (2023). Large-Scale Knowledge Synthesis and Complex Information Retrieval from Biomedical Documents. *J Anesth Pain Med, 8*(3), 130-137.

Abstract

Recent advances in the healthcare industry have led to an abundance of unstructured data, making it challenging to perform tasks such as efficient and accurate information retrieval at scale. Our work offers an all-in-one scalable solution for extracting and exploring complex information from large-scale research documents, which would otherwise be tedious. First, we briefly explain our knowledge synthesis process to extract helpful information from unstructured text data of research documents. Then, on top of the knowledge extracted from the documents, we perform complex information retrieval using three major components- Paragraph Retrieval, Triplet Retrieval from Knowledge Graphs, and Complex Question Answering (QA). These components combine lexical and semantic-based methods to retrieve paragraphs and triplets and perform faceted refinement for filtering these search results. The complexity of biomedical queries and documents necessitates using a QA system capable of handling queries more complex than factoid queries, which we evaluate qualitatively on the COVID-19 Open Research Dataset (CORD-19) to demonstrate the effectiveness and valueadd.

Index Terms: Information-Retrieval, Knowledge-Synthesis, Semantic-Retrieval, Question-Answering, CORD-19.

Introduction

The healthcare sector stands tall with an enormous amount of unstructured text data in documents, articles, biomedical journals, and JSON files, as well as structured data like tables, and electronic health records, often leading to a severe information overload challenge. To researchers and health professionals, extracting relevant information from a huge corpus of biomedical data is a complex and tedious task that delays the research outcome and involves considerable capital. Therefore, there is an increased urgency for an information retrieval system in the biomedical domain to retrieve such complex information.

Information Retrieval (IR) is a core task in many real-world applications, such as digital libraries, expert finding, web search and others. Information retrieval aims at retrieving information relevant to a query from large data collections, which has been an active research area in the healthcare domain [1-3]. Traditional information retrieval systems rely on lexical retrievers such as Boolean Retrieval, BM25 and statistical language models, which aim to find an exact match between the query and documents but fail to handle the problem of vocabulary and semantic mismatch [4]. Earlier studies in neural IR handle the problem of vocabulary mismatch by taking a different approach, such as maximum inner product search (MIPS) between GLoVe or Word2vec embeddings of query and document terms [5,6]. The problem of semantic mismatch was solved by leveraging contextual embeddings with the introduction of language models [7]. Lexical systems might fail to capture the semantics of the concepts, especially in biomedical data with complex terms that sometimes are quite ambiguous. Semantic systems can handle this ambiguity well, but these systems often have difficulty dealing with longer contexts. Hence, we need a hybrid framework that can accommodate both of these mechanisms. This paper conceptualizes a framework to help users access meaningful information extracted from massive corpora in the biomedical domain. They can explore the information in the form of knowledge graphs (Section III.A), search for specific information, and get answers to complex questions i.e, questions that require multiple contexts to provide an answer (e.g. *"What virus was isolated from a patient who died from acute respiratory failure?"*).

We propose an all-in-one information retrieval framework using lexical and semantic approaches, shown in Fig.1, that combines multiple functionalities like passage retrieval; triplet retrieval from knowledge graphs and complex QA. We also include faceted navigation for filtering the triplet search results, making it easier for the user to explore relevant information through a large amount of data. Our question answering system can answer complex queries by integrating multi-hop dense retriever which uses a dense iterative retrieval method [8]. The following describes how the paper is structured: Background information is provided in Section-II. The methodology is then discussed in Section-III with distinct subsections for its various components, experiments are discussed in Section-IV, and Section-V concludes the paper.

Background

To researchers and health professionals, extracting relevant information from a huge corpus of medical research documents and texts is a complex, time-consuming, and tedious task that delays the research outcome and involves considerable capital, yet is a necessity. Therefore, there has been an increased urgency for an information retrieval system in the medical domain to retrieve such complex information [9]. Recent years have witnessed an increase in information retrieval systems in the healthcare domain, such as a medical information retrieval system for e-healthcare records retrieval of semantically similar questions in healthcare forums and a system that uses information retrieval with an added component for classifying breast cancer [10-12]. Due to the pandemic, information extraction around COVID19 data has emerged as an active research area [13], predominantly using knowledge graphs [14-15]. Complex question answering, especially in the medical domain, has also become prominent [16]. These systems try to solve problems like knowledge graph (KG) generation on structured data, factoid question answering and searching entities in the KG [17]. These systems fail to address these use cases comprehensively. For example, the KG created might be ontology specific and cannot capture facts from open text or might only represent metadata information in the form of a graph; these systems also fail to provide an efficient integrated search and QA functionality such as ours [18,19].

Recent transformer-based retrievers mostly rely on the maximum inner product search between the dense representation of the query and the documents, generated using transformer models. These retriever-based systems are often supported by a re-ranker, based on variants of transformer models like SBERT and BERT-based cross-encoders [20].

Previous work on Open Domain Question Answering is mainly based on retriever and reader architecture, Iterative Retriever, Reader, Reranker (IRRR), which captures an initial set of keywords from the query, expands it based on the passages it retrieves from the database and re-ranks them [21,22]. These re-ranked passages are then passed to the reader to generate answers, and the whole process is iteratively repeated until the answer is found with high confidence. All of this makes the IRRR based systems highly complex due to the large number of components involved, coupled with longer inference time and higher memory consumption. Other retrieval methods use graph-based knowledge along with transformer models to find multi-hop reasoning paths [23,24].

Methodology

In this section, we explain the proposed model architecture. We use CORD-19 as an example dataset for explaining the pipeline and process throughout this paper, although the entire framework is flexible and should be translatable to a variety of datasets in biomedical literature [25].

Knowledge Synthesis

In the biomedical domain, data may be present in the form of blogs, articles, research papers, clinical documents, etc. We adopt a knowledge synthesis process to extract information in the form of subject-relation-object triplets from such unstructured text documents.

a) *Knowledge graph construction:* We first clean and preprocess the text from research documents. This data is indexed for further use by the retrievers to retrieve relevant contexts. Then we pass this text through our knowledge graph construction pipeline, which is as follows:-

1) Coreference Resolution on the sentences.
2) Extracting triplets (subject, relation, object pairs) using the Open Information Extraction (OpenIE 6) System from sentences [26].
3) Canonicalization of the extracted relations.
4) Linking extracted entities to appropriate ontology.

The above pipeline results in the formation of a knowledge graph, which consists of canonicalized and linked triplets, extracted from the biomedical documents. We also include metadata such as Authors, Institutions, Publication Year, etc. along with textual phrases from the original documents.

B. Complex Information Retrieval

Our Complex Information Retrieval system consists of three main components:- *Paragraph Retrieval, Triplet Retrieval*, and *Complex Question Answering.* We also have a spell checker that corrects spelling errors in the query asked by the user.

Spell correction: Often queries include misspelled terms resulting in irrelevant results. Therefore, a spell correction module trained on biomedical text is deployed to correct the query before handing it over to the retrievers, enabling the system to handle adversarial examples of misspelled terms robustly. The module is based on which uses Levenshtein Distance (edit distance) and the probability of the word appearing in the document [27].

$$correction(w) = argmax_{c\in candidates} P(c|w) \qquad (1)$$

Out of all possible candidate corrections, having an edit distance of 2 or less, the algorithm finds the correction c that maximizes the probability that c is the intended correction, given the original word w.

1) *Paragraph Retrieval:* To retrieve the most relevant piece of information from indexed documents, we introduce the paragraph retrieval functionality, where one paragraph is considered a unit of information and indexed. The retrieval combines four different search mechanisms viz, phrase search, bigram search, keyword search and semantic search. For all the mechanisms, we use ElasticSearch for indexing. We employ a cross-encoder to re-rank the results based on their relevance to the given query.

a) *Phrase Search:* Phrase search finds an exact match for the entire query or a part of the query, which can be specified by encoding the phrase in double quotes, e.g. *Human "SARSCoV"* infection where we retrieve the relevant documents by matching the exact phrase *"SARS-CoV"*.

b) *Bi-gram Search:* Bi-gram search splits the query into pairs of words, called bi-grams. These bi-grams are substrings of the query. The system searches the paragraph corpus for exact matches of these bi-grams (e.g. *Real-time PCR assay* ¿ [*'Real-time PCR', "PCR assay*']).

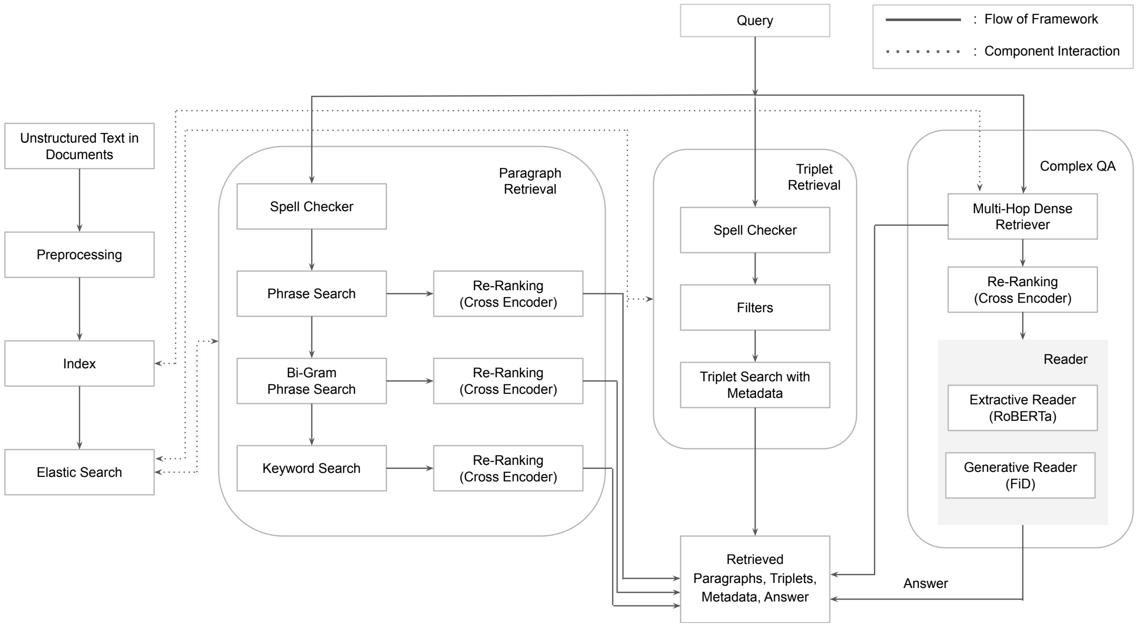
Query
Flow of Framework
Component Interaction
Unstructured Text in Documents
Preprocessing
Index
Elastic Search
Paragraph Retrieval
Spell Checker
Phrase Search
Bi-Gram Phrase Search
Keyword Search
Re-Ranking (Cross Encoder)
Re-Ranking (Cross Encoder)
Re-Ranking (Cross Encoder)
Triplet Retrieval
Spell Checker
Filters
Triplet Search with Metadata
Complex QA
Multi-Hop Dense Retriever
Re-Ranking (Cross Encoder)
Reader
Extractive Reader (RoBERTa)
Generative Reader (FiD)
Retrieved Paragraphs, Triplets, Metadata, Answer
Answer

Figure 1: Architecture of Complex Information Retrieval System. The query is passed through all three components of the framework. The paragraph retrieval combines results from the phrase, bigram, and keyword searches and retrieves relevant passages from the indexed data. The triplet retrieval retrieves related subject-object-relation pairs from the constructed knowledge graph. The complex question answering system gives an answer to the query along with the semantically retrieved passages from the Multi-hop Dense Retriever (MDR).

c) Keyword Search: This method tokenizes the query and searches through the corpus for matches and retrieves them in order of the count of matches in the specific paragraph. We use an Edge n-gram tokenizer with n being set to a minimum value of 4 and a maximum value of 30. The similarity function we use in this method is Okapi BM25 [5].

Re-ranking the results To re-rank the retrieved results based on relevance to the query, we use a MiniLM cross-encoder trained on MS MARCO is used [28,29]. This model outputs a relevancy score between 0 and 1 for every paragraph paired with the query. The order is decided based on this score with 1 being the highest.

Retrieved Paragraphs The results list consists of a predefined number of paragraphs (r). The passages are retrieved using phrase, bi-gram, and keyword search, in the respective order. This ordering is based on descending precision for individual mechanisms. We combine these passages with the passages retrieved using semantic search, from the retriever of the complex QA system. The retrieval process continues until the length of the results list is less than r (e.g. r=20).

2) Triplet Retrieval: To retrieve the most relevant triplets for the query from our large knowledge graph we need the triplet retrieval system. This methodology retrieves triplets (subject-relation-object) which are constructed using the Knowledge Synthesis pipeline (Section III.A) for all the CORD-19 research papers. An additional feature of faceted refinement is added on top to refine the results further by specifying values for different facets. We consider the subject and object types and subtypes as facets and join multiple such facets using a boolean AND condition to filter the retrieved results.

a) Triplet Index Construction: While indexing the data into the ElasticServer we use the following custom settings and analyzer for preprocessing the raw JSON data:-

1) Tokenize the documents using the Edge n gram method. 2) Filter the tokens to lowercase and ASCII folding.

b) *Retrieval*: The triplet retrieval component consists of the same similarity functions and search mechanisms used before viz. phrase search, bi-gram search, and keyword search. The results here consist of a list of triplets each containing subject, relation, and object. Additionally, we utilize triplet metadata like aliases, types, subtypes, descriptions, etc. Higher weightage is given to the subject, object, and relation triplet as compared to the metadata. Here, the weights can be manually tuned or trained.

c) *Faceted Refinement:* Faceted refinement is employed to assist researchers to refine the information retrieved using the facet fields Subject and object types and subtypes are considered facets. Multiple facets are joined together using a boolean AND condition filtering the retrieved results.

d) *Knowledge Graph Querying:* We also store our knowledge graph in the Neo4j graph database AuraDB with a particular schema to run structured queries on top of it for retrieving triplets and subgraphs, using CypherQL [30].

virus found in rhinolophus bats	
Paragraph Retrieval	**Triplet Retrieval**
The discovery of SARS-related CoVs in Kenyan bats adds to the diversity and geographic range of CoVs in Rhinolophus bats.	**Subject**: rhinolophus bats **Relation**: harbor **Object**: wide diversity of covs
Our long-term surveillances suggest that Rhinolophus bats seem to harbor a wide diversity of CoVs.	**Subject**: Yan Zhu **Relation**: Authored **Object**: Characterization of a New Member of Alphacoronavirus with Unique Genomic Features in Rhinolophus Bats
Semantic Search	
Global Epidemiology of Bat Coronaviruses However, there are not sufficient data to establish the prevalence of SARS-like CoVs in different bat host species, especially the species under the genus Rhinolophus. Interestingly, geographical factor does contribute to the diversity of SARS-like CoVs.	
Bat Coronaviruses in China It was strongly suggested that SARS-CoV most likely originated from Yunnan Rhinolophus bats via recombination events among existing SARSr-CoVs. These studies revealed that various SARSr-CoVs capable of using human ACE2 are still circulating among bats.	

Figure 2: Results from Complex Information Retrieval Framework for phrase. The paragraph retrieval retrieves passages relevant to the detection of rhinolophus bats. The triplet retrieval results subject-relation-object pairs from the knowledge graph. They include entity-relation-entity triplets from the passages and metadata triplets like document-reference-documents. The semantic search results contain passages retrieved from Multihop Dense Retriever (MDR).

As the query is a phrase, there is no response from the question-answering pipeline.

3) Complex Question Answering: The Complex Question Answering system can handle factoid questions, e.g. *"Where was coronavirus first discovered?"* as well as multi-hop questions which require going through multiple passages to answer the question, e.g. *"What bats are the main reservoir of the virus which is transmitted to humans via ACE2 receptor?"*. We split the passages from COVID-19 related documents into a maximum length of 300 tokens. Then, we pass these fixed length passages through a transformer encoder to generate dense embeddings for each passage. We store these embeddings in a dense index for further retrieval.

a) Retriever: The retriever searches through the dense index of CORD-19 documents and retrieves passages relevant to the query. To deal with multi-hop questions, we make use of the Multi-hop Dense Retriever (MDR) [8], which is an iterative retriever that uses a single RoBERTa-base model [31] to encode queries and passages into the same vector space. It is trained to iteratively search and retrieve relevant documents from the database using Facebook AI Similarity Search (FAISS) [32]. We have set the number of iterations to 2 in our system, but it is tunable. MDR retrieves two passages, related to each other based on reasoning paths or information about the entities in question, constituting one chain of retrieved contexts. Top-k such chains are retrieved based on their semantic similarity scores. The chains are then sorted based on the combined similarity score of the hops and further re-rank the retrieved passages using the MiniLM crossencoder. Then we send the passages to the reader models to generate answers. We also merge these semantically retrieved results from the MDR with the results from paragraph retrieval (Section III.B.1).

b) Reader: The reader is responsible for providing an answer given a context. We use two readers in our framework: Extractive reader and Generative reader. Extractive readers extract continuous answer spans from the retrieved passage. In contrast, generative readers are capable of generating answers even though they may not find them in the context provided. For extractive QA we use the RoBERTa model. This model for question answering takes the question tokens and context tokens as inputs and predicts the answer start and end tokens. For generative reader we use the Fusion-in-decoder (FiD) [33].

Experiments

Dataset

CORD-19 is a corpus of academic papers about COVID-19 and related coronavirus research, curated and maintained by the Allen

Institute for AI. The dataset has grown to index over 1M papers and includes full-text content for nearly 370K papers. Documents from CORD-19 are indexed and information retrieval is done on top of this index. The reader models are fine-tuned on the MRQA [34] dataset that contains preprocessed subsets of other domain-related datasets, making it a more generalized and suitable benchmark. The reader is also fine-tuned on Covid-QA [35], a medical question answering dataset around COVID-19.

How many species exist of the mammals that are the main reservoir of coronaviruses?	
Paragraph Retrieval	**Triplet Retrieval**
High species diversity (about 1,150 in the world), high mobility and the fact that they represent a source of emerging infections for humans make bats one of the most epidemiologically relevant group of mammals to study disease ecology.	**Subject:** Isolation and characterization of a bat SARS-like coronavirus that uses the ACE2 receptor **Relation:** References **Object:** Detection of Novel SARS Coronaviruses in Bats
Bats are the second largest order of mammals, comprising more than 1200 different species	**Subject:** alpha-coronaviruses **Relation:** may be derived from **Object:** bat coronaviruses
Complex Question Answering	
QA Response: 1200	
Replication of MERS and SARS coronaviruses in bat cells offers insights to their ancestral origins Coronaviruses(CoVs) are important pathogens in animals and humans, responsible for a variety of respiratory, hepatic, and neurological diseases. Bats are an important reservoir of alpha coronaviruses and beta coronaviruses, which may jump to other species.	
New Adenovirus Groups in Western Palaearctic Bats Bats are the second largest order of mammals, comprising more than **1200 different species.** Their high vagility and the organization typically in social groups predispose them to infection and viral dissemination.	

Figure 3: Results from Complex Information Retrieval Framework for a multi-hop question. The paragraph retrieval retrieves passages relevant to the question. The triplet retrieval results subject-relation-object pairs from the knowledge graph. They include entity-relation-entity triplets from the passages and metadata triplets like document-reference-documents. The complex question answering system first retrieves a passage that talks about the main reservoir of coronavirus i.e bat and then retrieves a passage that talks about the number of species of bats.

Training

The extractive reader is a RoBERTa-base model, already pre-trained on WikiMultiHop. We initially fine-tune it on a generalized dataset, MRQA, and then fine-tune it further on Covid-QA for two epochs to learn the biomedical context. To avoid losing important information, we split the documents present in the CORD-19 dataset into chunks of size C, such that each chunk contains strides (overlap) of size S with the previous chunk. We make sure that C is less than 512, as most transformer models cannot process tokens more than 512 and S is set as 128 to overlap optimal information.

The generative reader is the Fusion-in-Decoder model, with T5-base architecture, already pre-trained on TriviaQA [36]. We fine-tune FiD on MRQA and then on Covid-QA, for a total of 45000 steps with a batch size of 8.

Results

We evaluate our framework qualitatively on the CORD-19 dataset. We use two kinds of queries to test the performance of various components in our framework. First, for the phrase *“virus found in rhinolophus bats”*, we get a list of passages from paragraph retriever and multi-hop dense retriever along with multiple triplets that talk about rhinolophus bats (as shown in Fig.2). In case of a complex question like-*”How many species exist of the mammals that are the main reservoir of coronaviruses?”*, the complex QA system reasons over the passages retrieved by our multi-hop retriever and the reader gives us the correct answer. It first retrieves a passage that talks about the main reservoir of coronavirus i.e, bats, followed by a passage that talks about the number of species of bats (1200), which can be seen in Fig.3. We also observe that our passage retrieval mechanism retrieves highly relevant passages. They contain the keywords in the query and are contextually similar to the query asked. The triplet retrieval also retrieves the best set of triplets related to the query. Overall our system can provide the user with the most relevant information to the query asked using lexical as well semantic retrievers unlike similar information extraction systems around COVID19, such as that uses only BM25 for retrieval and not an iterative retriever like ours that also enables our question answering system to reason over more than one document and provide the answer [20]. Supports keyword and entity search, it fails to accommodate phrase search, bi-gram search and semantic search like our search system [21]. Both of these systems do not perform triplet retrieval on knowledge graphs.

Evaluation

We evaluate our framework on related open-source datasets due to the unavailability of labeled data for CORD-19. We evaluate the paragraph retrieval pipeline on another COVID19 related dataset, TREC-COVID [37]. Here we use Precision and NDCG as the metric. NDCG is the ratio of the Discounted Cumulative Gain (DCG) of a recommended and ideal order. It is evident that phrase search with MiniLM-L-6-v-2 re-ranker yields better results when compared to results without reranking, as shown in Fig.4.

We evaluate the performance of the reader models on the MRQA-dev data split by calculating the exact match and the F1 scores for all subsets of the dataset. We see that the model's performance varies massively depending on the kind of data as seen in Table.1.

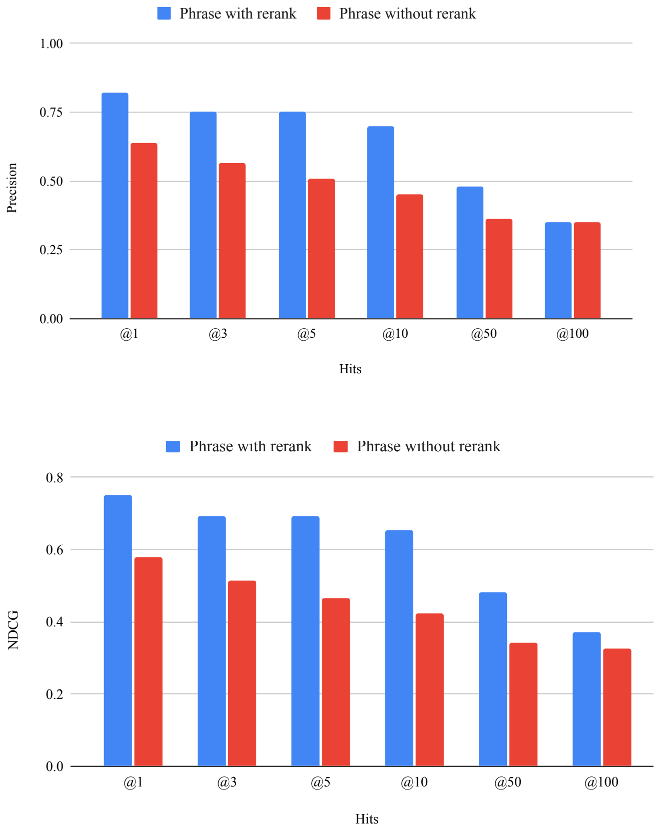

Figure 4: Evaluation of Paragraph Retrieval on TREC-COVID. Phrase with reranker (denoted by blue) outperforms phrase without re-ranker (denoted by red) across different top-k comparisons in both Precision and NDCG metrics.

Table I: EVALUATION OF READERS ON MRQA-DEV SUBSETS

Subset	No. of Questions	Extractive Reader		Generative Reader
		Exact Match (%)	F1 score (%)	Exact Match (%)
SQUAD	10507	83.76	90.48	69.8
Trivia-QA-web	7785	12.76	14.24	45.7
Search QA	16980	10.2	10.94	60.4
Hotpot QA	5901	60.78	76.74	41.5
NQ Short	12836	64.78	76.74	48.9
News QA	4212	52.84	66.62	36.2

Conclusion

In this paper, we presented a complex information retrieval framework built on COVID-19 related biomedical documents that can perform both lexical and semantic search and retrieve paragraphs along with a knowledge graph consisting of triplets extracted from unstructured text. We also use faceted refinement to filter the results. We demonstrate our complex QA system, which gives the researcher a pinpoint answer to the query asked. We find that this framework makes it easier for the researcher to search for specific information from massive corpora. In our future work, we plan to add functionalities like query expansion and query intent classification along with scalable semantic retrieval on top of the knowledge graph.

Acknowledgment

We thank Varun V, Advaith Shankar, Nim Sherpa and Saisubramaniam Gopalakrishnan for their assistance with figures, and useful suggestions that helped improve the manuscript.

References

1. Sakji, S., Dibad, A. D., Kergourlay, I., Darmoni, S., & Joubert, M. (2009, April). Information retrieval in context using various health terminologies. In 2009 Third International Conference on Research Challenges in Information Science (pp. 453-458). IEEE.
2. Al-Qahtani, M., Katsigiannis, S., & Ramzan, N. (2020). Information Retrieval from Electronic Health Records. Engineering and Technology for Healthcare, 117.
3. Weiming, W., Shihong, C., Xi, C., & Fan, Z. (2008, December). Knowledge-based document retrieval in medical domain. In 2008 International Symposium on Knowledge Acquisition and Modeling (pp. 226-230). IEEE.
4. Jones, K. S., Walker, S., & Robertson, S. E. (2000). A probabilistic model of information retrieval: development and comparative experiments: Part 2. Information processing & management, 36(6), 809-840.
5. Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).
6. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
7. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
8. Xiong, W., Li, X. L., Iyer, S., Du, J., Lewis, P., Wang, W. Y., ... & Oğuz, B. (2020). Answering complex open-domain questions with multi-hop dense retrieval. arXiv preprint arXiv:2009.12756.
9. Jackson, R., Kartoglu, I., Stringer, C., Gorrell, G., Roberts, A., Song, X., ... & Dobson, R. (2018). CogStack-experiences of deploying integrated information retrieval and extraction services in a large National Health Service Foundation Trust hospital. BMC medical informatics and decision making, 18(1), 1-13.
10. Sengan, S., Kamalam, G. K., Vellingiri, J., Gopal, J., Velayutham, P., & Subramaniyaswamy, V. (2020). Medical information retrieval systems for e-Health care records using fuzzy based machine learning model. Microprocessors and Microsystems, 103344.
11. Wang, Y., Mehrabi, S., Mojarad, M. R., Li, D., & Liu, H. (2015, October). Retrieval of semantically similar healthcare questions in healthcare forums. In 2015 International Conference on Healthcare Informatics (pp. 517-518). IEEE.
12. Kumari, M., & Ahlawat, P. (2022). Intelligent Information Retrieval for Reducing Missed Cancer and Improving the Healthcare System. International Journal of Information Retrieval Research (IJIRR), 12(1), 1-25.
13. Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Deep Learning applications for COVID-19. Journal of big Data, 8(1), 1-54.
14. Wise, C., Ioannidis, V. N., Calvo, M. R., Song, X., Price, G., Kulkarni, N., ... & Karypis, G. (2020). COVID-19 knowledge graph: accelerating information retrieval and discovery for scientific literature. arXiv preprint arXiv:2007.12731.
15. Esteva, A., Kale, A., Paulus, R., Hashimoto, K., Yin, W., Radev, D., & Socher, R. (2021). COVID-19 information retrieval with deep-learning based semantic search, question answering, and abstractive summarization. NPJ digital medicine, 4(1), 68.
16. Jin, Q., Yuan, Z., Xiong, G., Yu, Q., Ying, H., Tan, C., ... & Yu, S. (2022). Biomedical question answering: a survey of approaches and challenges. ACM Computing Surveys (CSUR), 55(2), 1-36.
17. Lan, Y., He, G., Jiang, J., Jiang, J., Zhao, W. X., & Wen, J. R. (2022). Complex knowledge base question answering: A survey. IEEE Transactions on Knowledge and Data Engineering.
18. Su, D., Xu, Y., Yu, T., Siddique, F. B., Barezi, E. J., & Fung, P. (2020). CAiRE-COVID: A question answering and query-focused multi-document summarization system for COVID-19 scholarly information management. arXiv preprint arXiv:2005.03975.
19. Ambavi, H., Vaishnaw, K., Vyas, U., Tiwari, A., & Singh, M. (2020, October). CovidExplorer: A multi-faceted AI-based search and visualization engine for COVID-19 information. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (pp. 3365-3368).
20. Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.
21. Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S., & Chua, T. S. (2021). Retrieving and reading: A comprehensive survey on open-domain question answering. arXiv preprint arXiv:2101.00774.
22. Qi, P., Lee, H., Sido, O., & Manning, C. D. (2020). Answering open-domain questions of varying reasoning steps from text. arXiv preprint arXiv:2010.12527.
23. Saxena, A., Tripathi, A., & Talukdar, P. (2020, July). Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In Proceedings of the 58th annual meeting of the association for computational linguistics (pp. 4498-4507).
24. Wang, Z., Li, L., Zeng, D. D., & Chen, Y. (2018, November). Attention-based multi-hop reasoning for knowledge graph. In 2018 IEEE International Conference on Intelligence and Security Informatics (ISI) (pp. 211-213). IEEE.
25. Wang, L. L., Lo, K., Chandrasekhar, Y., Reas, R., Yang, J., et al. (2020). Cord-19: The covid19 open research dataset.
26. Kolluru, K., Adlakha, V., Aggarwal, S., & Chakrabarti, S. (2020). Openie6: Iterative grid labeling and coordination analysis for open information extraction. arXiv preprint arXiv:2010.03147.
27. Norvig, P. (2009). How to Write a Spelling Corrector, v. 10.8. 2007.
28. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). Minilm: Deep self-attention distillation for task-ag-

nostic compression of pre-trained transformers. Advances in Neural Information Processing Systems, 33, 5776-5788.

29. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., & Deng, L. (2016). MS MARCO: A human generated machine reading comprehension dataset. choice, 2640, 660.
30. Francis, N., Green, A., Guagliardo, P., Libkin, L., Lindaaker, T., Marsault, V., ... & Taylor, A. (2018, May). Cypher: An evolving query language for property graphs. In Proceedings of the 2018 international conference on management of data (pp. 1433-1445).
31. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.
32. Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with gpus. IEEE Transactions on Big Data, 7(3), 535-547.
33. Izacard, G., & Grave, E. (2020). Leveraging passage retrieval with generative models for open domain question answering. arXiv preprint arXiv:2007.01282.
34. Fisch, A., Talmor, A., Jia, R., Seo, M., Choi, E., & Chen, D. (2019). MRQA 2019 shared task: Evaluating generalization in reading comprehension. arXiv preprint arXiv:1910.09753.
35. Möller, T., Reina, A., Jayakumar, R., & Pietsch, M. (2020, July). COVID-QA: A question answering dataset for COVID-19. In Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020.
36. Joshi, M., Choi, E., Weld, D. S., & Zettlemoyer, L. (2017). Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. arXiv preprint arXiv:1705.03551.
37. Voorhees, E., Alam, T., Bedrick, S., Demner-Fushman, D., Hersh, W. R., Lo, K., ... & Wang, L. L. (2021, February). TREC-COVID: constructing a pandemic information retrieval test collection. In ACM SIGIR Forum (Vol. 54, No. 1, pp. 1-12). New York, NY, USA: ACM.
38. Saxena, S., Sangani, R., Prasad, S., Kumar, S., Athale, M., Awhad, R., & Vaddina, V. (2022, December). Large-Scale Knowledge Synthesis and Complex Information Retrieval from Biomedical Documents. In 2022 IEEE International Conference on Big Data (Big Data) (pp. 2364-2369). IEEE.