AN IR-BASED QA SYSTEM FOR IMPACT OF SOCIAL DETERMINANTS OF HEALTH ON COVID-19

Priyanka Addagudi and Wendy MacCaull

Department of Computer Science, St. Francis Xavier University, Canada

ABSTRACT

Question Answering (QA), a branch of Natural Language Processing (NLP), automates information retrieval of answers to natural language questions from databases or documents without human intervention. Motivated by the COVID-19 pandemic and the increasing awareness of Social Determinants of Health (SDoH), we built a prototype QA system that combines NLP, semantics, and IR systems with the focus on SDoH and COVID-19. Our goal was to demonstrate how such technologies could be leveraged to allow decision-makers to retrieve answers to queries from very large databases of documents. We used documents from CORD-19 and PubMed datasets, merged the COVID-19 (CODO) ontology with published ontologies for homelessness and gender, and used the mean average precision metric to evaluate the system. Given the interdisciplinary nature of this research, we provide details of the methodologies used. We anticipate that QA systems can play a significant role in providing information leading to improved health outcomes.

KEYWORDS

Question Answering, Ontology, Information Retrieval, Social Determinants of Health, COVID-19.

1. INTRODUCTION

The world wide web has allowed researchers and decision-makers in medical and other domains to easily access data and documents and build vast repositories of related knowledge. However, determining the relevant information for a particular research problem or decision-maker is difficult. The increased power to perform computations has enabled the application of Artificial Intelligence techniques (NLP based QA systems [8], Machine Learning, etc.) to the problem of finding relevant information.

Coronavirus infection emerged in December 2019 in Wuhan city, China. The infection rapidly spread across parts of China and later worldwide. The COVID-19 pandemic is a worldwide crisis endangering the health of everyone on the planet. Droplets from mouth, nose and direct contact with a person infected with COVID leads to transmission of the virus. The Social Determinants of Health (SDoH) are socio-economic conditions that impact people's health. The authors of paper [11] discuss how the SDoH impact disadvantaged populations during times of crisis. Studying SDoH and how they impact crises like COVID can help decision-makers manage health emergencies so that everyone has an equal opportunity to stay healthy. A study done on impacts of SDoH on COVID-19 by [9] states preliminary evidence from surveillance and media reports have shown that SDoH contributes to high rates of COVID-19 infection, hospitalization, and mortality. Recently, the CTV news article [10] stated the socio-economic and health inequities are the topmost important factors out of five big lessons learned from the pandemic. Considering the importance of COVID-19 related topics, the Allen Institute for AI, in collaboration with other organisations and Kaggle, released the COVID-19 Open Research Dataset (CORD-19) [50]. The scientific papers on Covid-19 and related historical coronavirus research are growing, and CORD-19 is the resource for all of them. This dataset aims to enable researchers to work with new tools to analyse and overcome the problems associated with coronavirus. Overall evidence indicating a crucial role for SDoH and related social factors in shaping health has become so compelling that it cannot be ignored [57].

The objective of this work is to develop an integrated architecture (Domain SDoH and QA system) for a Question and Answering system dealing with a corpus related to "Impact of SDoH and associated risk factors on transmission and outcomes of COVID-19". The NLP based QA system allows the users to input the query in natural language, which reduces the technicality and time needed to get the information. According to our literature search, there is no QA system for the impact of SDoH factors on COVID-19. This has motivated the long-term goal of our research which is to create a QA system that can assist people in understanding the impact of SDoH on COVID-19 or future pandemics due to infectious diseases.

Question answering (QA), is a branch of information retrieval and natural language processing (NLP) [1]. QA systems automatically provide answers to questions posed in a natural language. They are different from Information Retrieval (IR) systems or search engines like Google that return a ranked list of relevant sources based on a set of keywords. A QA system finds and returns relatively short and concrete answers in the form of: a sentence, a paragraph, a fragment of the text, or even a word that answers a given question, by analysing many documents where the answers may be found. QA systems can reduce technical difficulties by enabling humans to interact with machines using natural languages instead of programming languages or text-based commands. QA systems are evolving worldwide and used in a wide range of application areas, from biomedicine to tourism [2]. Several literature reviews have focused on a variety of aspects of QA systems, e.g., domain [3][4], information retrieval paradigm [5], hybrid-based paradigm [6]. However, no established relationships exist between domains, algorithms, techniques, and systems [7].

This paper showcases our prototype QA architecture, AQuA, demonstrating how a QA system integrated with a domain, can answer questions related to severe outcomes of COVID-19 due to some SDoH factors (homelessness and gender). The QA system developed allow users (decision-makers) to understand and quickly determine some circumstances (related to SDoH) that can lead to transmission and severe outcomes associated with the Covid-19 infection. The motivating research for this work is MEANS, a QA system built to address the problems in the medical domain combining NLP techniques and semantic web technologies [12]. Our architecture differs from MEANS in various factors, namely: a) Named Entity Recognition (NER) and Relation Extraction (RE) processes are replaced with rule-based triple extraction compared to that work. b) Three layers for similarity measures are used for linking. c) The Resource Descriptive Framework (RDF) triple store is replaced by Elastic Search database, questions are converted to Elastic Search queries and the Elastic Search scoring model is used for answer ranking. We built an ontology for the two SDoH factors under consideration in this work (i.e., homelessness and gender) and COVID-19 to support the QA system.

The remainder of this paper is described as follows: section 2 outlines the background required for this research and related work, section 3 details the architecture and implementation, section 4 describes the evaluation results, section 5 concludes this work and outlines some future works.

2. BACKGROUND AND RELATED WORK

2.1. Background

NLP is a field that deals with the interactions between computers and humans, especially how to program computers to process and analyse large amounts of natural language data [13]. QA is a Computer Science discipline within the field of NLP, which is concerned with building systems that can automatically answer questions posed in a natural language [14]. The three most common essential modules in most QA systems are document retrieval module, question processing module, and answer extraction and formulation module. As stated in [15], QA systems are different from IR systems (aka, search systems) which allow users to input keywords to search and returns lists of documents in response. The difference between IR and QA systems is that QA systems internally have a stronger dependency on NLP techniques such as parsing, named-entity detection, semantic role labelling, tree-matching, and, in some implementations, logical inference. Any QA system typically has many modules, and its performance varies and depends on how the components are integrated within an algorithm. There are two types of QA systems: a) Open-domain question answering deals with everything (not specific to any domain) so there are many documents to retrieve answers from. b) Closed-domain question answering deals with questions on a specific domain, so there are a limited number of documents to retrieve answers from. Closed-domain QA has some advantages over open-domain QA, e.g., users have an idea about what domain they are using the system for and do not use it for general purposes. Also, the users can leverage domain properties to build vocabularies, ontologies and other models.

2.1.1. Components in NLP pipeline

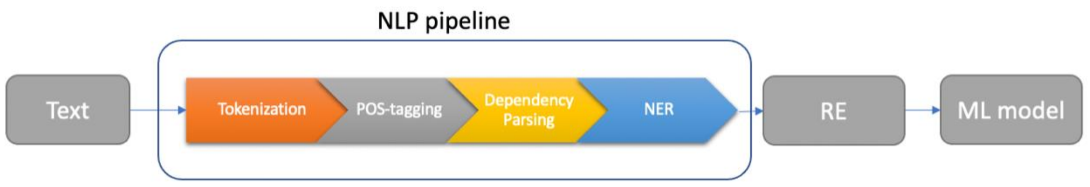


Figure 1: NLP pipeline

The NLP pipeline is shown in Figure 1. Tokenization is the task that splits the sentences in documents into sentence tokens or words in a sentence to meaningful word tokens. The tokens are identified and matched with the training data used for other processes in the pipeline. POS-tagging refers to assigning every word in a sentence with its part of speech (noun, verb, adverb, and so on) [16]. In general, Hidden Markov models [17] are used for POS-tagging task because information about the context of a word can be used to predict what the POS-tag might be. There are also rule-based POS-taggers, which use predefined rules to perform the tagging or learn rules from the corpus and rely less on probabilistic methods. Named entity recognition seeks to locate and classify entities mentioned in the text into pre-defined categories like chemical, symptom, disease etc. A named entity is a real-world object usually with a proper name - examples are Donald Trump, Italy, Facebook etc. Dependency Parsing is a way to analyse a sentence or break up a sentence to understand the structure of a sentence. Parsing is a technique that can be applied to any statement with formal grammar. RE is the task of extracting semantic relationships from text (usually between two entities) [18]. This cleaned and pre-processed data from the pipeline is converted to vectors and used to train the ML models.

2.1.2. Semantic Web Technologies

Ontology (a branch of philosophy known as metaphysics) is the study of the nature of being, becoming, existence, or reality [19]. It deals with questions concerning what things exist or can be said to exist and how they can be grouped or classified based on similarities and differences. Ontologies, which are richer than terminologies [20] can be used to formally represent knowledge within a domain as a set of concepts, relations between concepts, instances of the concepts [21] and axioms. A pair of concepts or instances with a relation between them is known as 'semantic triple'. One semantic triple can be connected to another semantic triple whenever they share any common concept. In knowledge graphs, which are frequently used to represent ontologies, these concepts or instances aka entities are represented as nodes and the relations are represented as edges. Each ontology class and relation may have a Uniform Resource Identifier (URI) associated with it which is used to identify the concepts and relations. Resource Description Framework (RDF) expresses information about resources like people, physical objects, and abstract concepts [22]. RDF facilitates sharing of information on the web between different applications without loss of its meaning by providing a common framework for expressing the information. The attractive feature of RDF is the ability to distribute the development of an ontology (defining a set of data and their structure for other programs to use them) by basing terms in dereference able, globally unambiguous identifiers. To achieve this, RDF uses IRI (International Resource Identifier), which is a generalization of URIs whose scalability has been proven by the success of the web [23]. This generalization allows non-ASCII characters to be used in IRI character strings [24]. The Universal Resource Locator is one form of IRI. Another form of IRI provides an identifier for the resource but does not provide its location or explain how to access it. Triples in RDF are expressed in statements composed of a subject, object, and a predicate (i.e., relationship) between them.

2.1.3. Elastic Search

Search over RDF data is supported by two methods (a) By translating keyword queries to structured (SPARQL) queries [25] (b) By building or leveraging Information Retrieval Systems using classical IR methods for indexing and retrieval [27]. Our architecture is based on method (b) involving ElasticSearch, so the results or scoring/ranking is dependent on commonly used IR ranking functions.

ElasticSearch is open-source and one of the most popular search engines under the Apache 2.0 license [26]. ElasticSearch allows storage and searches a large amount of data offering a distributed architecture and emphasises scalability and reliability. ElasticSearch has a powerful query language known as Domain Specific Language (DSL), which supports advanced search features [27] such as search based on filter-context and query-context. Apache Lucene library can handle all types of data (textual, numerical, geospatial, structured, and unstructured), allowing the users of ElasticSearch were able to focus on scalability, usability, and performance. ElasticSearch provides an easy way to index documents enabling users to quickly query the nearest neighbour using a similarity metric based on TF-IDF [28]. Retrieval accuracy is affected by indexing and retrieval options in ElasticSearch. In ElasticSearch, the default model for similarity matching is BM25 [28] based on TF-IDF similarity measure (described in section 2.1.4). The data in ElasticSearch is stored in indices containing a different type of documents. This data stored can be searched and updated by ElasticSearch.

For indexing in ElasticSearch, each term will be used as a key to a multi-value map creating key-value pairs, which is called an 'inverted index'. The retrieval process depends on the i) query type, ii) weighing methods and iii) similarity models offered by ElasticSearch. There are two types of query clauses: a) filter-context - using exact match with query; b) query-context - using relevance scoring. As our interest is primarily free text search, we focus on query-context. Queries are further categorised into single-match queries, multi-match queries and boolean queries. As the name suggests, a single-match query is executed over a single field in a database, while a multi-match query is executed over multiple fields in a database. A Boolean query is the combination of single and multi-match query types. Boolean queries use clauses such as MUST or SHOULD. Weighing is an important factor for improving relevance at retrieval time. In this method, we usually apply weights on various fields. For instance, in ElasticSearch the field containing the object keyword is twice as important as the fields containing the subject and predicate keyword and has more weight. The way scoring works in real-time is different from generic scenarios. E.g., (as found in [29]) a document titled *Shard Selection Algorithm* contains the term algorithm; however, it is not relevant to the query *Algorithm for pathfinding*. Determining the relevancy to a query is a fundamentally hard problem. We can overcome this problem by implementing some of the most common scoring models like the TF-IDF scoring model.

The RDF datasets (aka web-of-data) are generally queried through a structured query language (such as SPARQL). But this is a difficult task for any person who is not proficient in SPARQL even though they are familiar with keyword search. Hence, there is a need for an effective method for keyword search over RDF datasets. Keywords are the phrases that a user type into search engines to retrieve an answer. Keyword search is used to optimise the search engine by adding weights to the keywords. In the example *Shard Selection Algorithm,* word *Shard* is given more weightage (using TF-IDF scoring model) than other two words so that search engine recognises the keyword for the search is Shard. There is widespread evidence [27] to the use of out-of-the-box IR systems like ElasticSearch in many contexts. The experiment by [27] investigated how such existing document-centric Information Retrieval Systems (IRS) can be used for enabling keyword search over RDF datasets, and how they perform compared to dedicated keyword search systems for RDF.

2.1.4. Linking

Linking is a process of mapping the extracted entities (subject or object) and relations (predicates) from the triple extraction process to the concepts and relations present in the knowledge bases (ontologies). In general, concepts or classes and relations in the knowledge base have unique identifiers. The combination of NLP and Semantic web technologies enables users to combine both structured and unstructured data. The linking process is difficult due to the high ambiguity of entity mentions and relations, which includes polysemy and multiword synonym [30]. Although this method is difficult, it can help search engines to disambiguate and retrieve the closest answer as the top ranked answer.

Linking can be performed in many methods [30]: a) Methods based on similarity - this can be done using Fuzzy matching, TF/IDF, Cosine similarity, and others. b) Methods based on machine learning - this can be done using classifiers like Support Vector Machine (SVM) classifiers and others. c) Methods based on graphs - this can be done by constructing a graph with nodes that are entities and edges which are relations. There are 2 main kinds of similarity measures, syntactic and semantic. Fuzzy matching is syntactic, cosine similarity is semantic; TF-IDF, N-gram are hybrid. Fuzzy logic helps in dealing with the problem of knowledge representation in an environment of uncertainty and imprecision. Fuzzy matching is a string-matching algorithm that measures the similarity between two strings using edit distance, also known as Levenshtein edit distance [31].

TF-IDF is also known as Term Frequency - Inverse Document Frequency [29]. TF-IDF can be used to calculate the similarity between two records by considering the frequency of the word in the data or documents. The first term in TF-IDF is Term Frequency, $Tf_{(t,\ d)}$ which means the frequency of the word t in the document d, calculated as Equation (1). Here, it assumes that a document having more than one match of the highest weighted term is a better match than a document having a single match. Document Frequency of term t, Dft, over all documents is calculated by counting the number of postings for a term in the inverted index, and the Inverse Document Frequency (IDF) for a term t in document d, Idf_t is calculated as Equation (2).

$$T\,f_{(t,\,d)} = \sqrt{f_t} \tag{1}$$

$$Idf_t = \log\left(\frac{N}{Dft\ +1}\right) \tag{2}$$

where N is the total number of documents. For scoring, the $T\,f_{(t,\,d))}$ weights are combined with Idf_t weights to multiplied by the score, as shown in Equation (3).

$$score(d,q) = \sum_{t\epsilon q} T\,f_{(t,d)} \times\ Idf_t \tag{3}$$

where q is the query. Soft TF-IDF is very similar to the TF-IDF concept which considers how frequently various combinations of words appear in data and calculated as Equation (4):

$$Sim_{SoftTFIDF}(x,y) = \sum_{\omega \in CLOSE(\theta,x,y)} \frac{((V(\omega,x))}{\sqrt{\sum_{\omega} V(\omega,x)^2}} \frac{((V(\omega,y))}{\sqrt{\sum_{\omega} V(\omega,y)^2}} D(\omega, y) \tag{4}$$

Where x, y are two strings from documents X, Y, ω is the word in the string and θ is the threshold defined for comparison, $V(\omega,x)$ and $V(\omega,y)$ are defined as the TF-IDF weight of token ω in string x and y given by frequency count in X and Y, $D(\omega,y)$ is essentially a normalizing coefficient, which dampens the impact on the Soft TF-IDF [67].

Context can be very important when working with textual data. We may sometimes lose the context in vector representations of a word, knowing only the count of every word. To some extent, N-grams, and in particular bi-grams, will help us solve this problem. An n-gram is a string of elements such as letters, words etc., that appear in a continuous sequence. We will be dealing with words being the item, but based on the use case, it could be letters, syllables, or sometimes, in the case of speech, phonemes. When n=2, it is called a bi-gram. Bi-grams in a text can be calculated using the conditional probability of a token with respect to its preceding token. Another way of calculating bi-grams is by choosing words that appear next to each other, but it is more effective to use bi-grams that are likely (using the conditional probability) to appear as a pair; such a bi-gram is called a collocation. Character n-gram (charNgram) is a character-based compositional model which is used to embed textual sequences [32]. In this method, each word is represented as a bag of character n-grams. The vector representation of a string or word is associated with each character n-gram. The end character embedding is the average of the distinct character n-gram embeddings. Using the character embeddings methodically and efficiently provides morphological features [33].

The model implemented in this work compares the scores from all the 3 models and finds the similarity between entities with ontology concepts. By using all three methods, we can deal with various categories of terms like: misspelled words, words in a similar context, large vocabularies, and many rare words.

2.1.5. Domain – SdoH

Existing social inequities in health may increase the risk of severe COVID-19 outcomes, such as hospitalization and death [9][10][11]. Racialized populations, Socioeconomic status, Homeless populations, Gender, Incarcerated populations, Education are some SDoH factors. Obesity, hypertension, diabetes, cardiovascular disease, and chronic respiratory disease and asthma are a few risk factors or conditions that may lead to co morbidities (overlap of different conditions) that may be associated with increased risk for severe outcomes from COVID-19. To help vulnerable people/groups in times of emergency, the focus should be on the roots of the problem. There is a need for all, not just decision-makers, to understand how social determinants and associated risk factors can impact the mortality rate as low mortality rates are associated with community support and cohesion. As stated in [11], pandemics are more of a social problem than a healthcare problem. Understanding factors like SDoH that play an important role in health and healthcare can facilitate access to medical and non-medical needs for everyone on a more equitable basis. Integration of SDOH into efforts to eliminate disparities in health and healthcare can be one solution to reduce the impact (transmission and outcomes) globally [11].

2.2. Related Work

BASEBALL [34] and LUNAR [35] are some of the oldest and well know QA systems that answer questions related to the US baseball league and geological analysis of rocks returned by the Apollo moon mission, respectively. The most noteworthy system is IBM Watson [36], which is the best example of a successful QA system. Other commercial products in the area of personal assistants include Apple's Siri (in 2011), Amazon's Alexa (in 2014) and Microsoft's Cortana (in 2014), Samsung's Bixby, and Google Assistant. The adoption of QA-based personal assistants (capable of answering variety of questions) has been observed over the last few years.

The QA Systems architecture depends on the underlying knowledge source like plain text, data graphs (RDF), or mixed (plain text and data graphs). The authors of [1] have done a comprehensive survey on the question answering system over RDF and Linked Data, documents, and mixtures of these to get a clear understanding of the QA systems. An RDF KB describes real-world entities as well as the relations between them. Integration of RDF KBs with other KBs or data sources (e.g., CSV, SQL tables) is necessary as it makes KBs able to support QA systems both in Open and Closed domains. Availability of tools or techniques relative to the integration of various databases makes the integration task easier. This data integration task can be done in many ways; some are: 1) interlinking different KBs based on ontology mapping methodologies [37] [38], 2) transforming and properly integrating other data source types into RDF ontologies using semantic labelling techniques [39] [40].

As our method is based on keyword search over RDF data by adapting an IR system using classical IR methods for indexing and retrieval, we will report related work to showcase the difference in our approach. The paper [27] presented a study that investigates the challenges and techniques to overcome them while querying RDF triples with ElasticSearch (explained in chapter 3). One of the initial systems that used IR system to query RDF data was Falcon [41]. In this paper, each document is mapped to with the textual description of the maximum subset of connected RDF triples. In contrast, our work uses URIs for mapping and querying the triples. The ranking of the documents for mapping is based on two factors. The first factor is cosine similarity which is used for mapping keyword terms to documents. The other factor for ranking is the popularity of each document. In our system we used three methods, fuzzy matching, TF-IDF and charNgram for similarity measure, and used the ElasticSearch score function for ranking the answers.

The author of [27] states that several related systems are evaluated in the entity search track of the SemSearch10 workshop [42]. These systems show variations in the TD-IDF weighting adapted for RDF data and returned a ranked list of entities. Although we follow a similar approach in our implementation, these systems are implemented on different datasets. [43] is an approach that uses inverted lists over terms that appear as entities, the keyword query is translated to a logical expression that returns the URIs of the matching entities; in contrast we indexed used ElasticSearch queries to retrieve the answers. [44] is the work that makes use of ElasticSearch, which is a text-based entry point to the Linked Data cloud. ElasticSearch was also used for indexing and querying Linked Bibliographic Data in JSON-LD format [45]. [45] introduces a solution for representing and indexing bibliographic resources by retaining the integrity and extensibility of Linked Data also supporting fast, customizable indexes in an application-friendly data format. The methodology uses JSON-LD to represent RDF graphs in JSON, which is suitable for indexing with ElasticSearch. Unlike this approach, we applied these methods to convert ontologies and RDF triples to JSON format and we query the data related to a confined domain rather than using a generic dataset.

3. ARCHITECTURE AND IMPLEMENTATION

Figure 2 presents the architecture (AQuA) we used for this research work. In this section, we briefly described each process and specify the details of our implementation.

3.1. Ontology Augmentation

Ontology augmentation process involves selecting, extracting, and reorganizing content from various sources to produce an ontology meeting the specifications of a particular domain and/or task. [46] defines ontology augmentation as the process of enriching an ontology by: i) incorporating new concepts, instances and relations from external resources which include, other ontologies, text, and databases etc., and ii) adding axioms and properties to the ontology. We studied the related work on ontologies related to homelessness and gender factors. The CODO [47] ontology is a COVID-19 ontology which is a data model for publishing COVID-19 data on the web as a knowledge graph. The Homelessness and Clinical Data Recording [48] ontology is a respected conceptual framework used to define degrees of housing insecurity and homelessness internationally. The Gender, Sex, and Sexual Orientation (GSSO) ontology [49] is aimed at bridging gaps between linguistic variations inside and outside the healthcare environment.

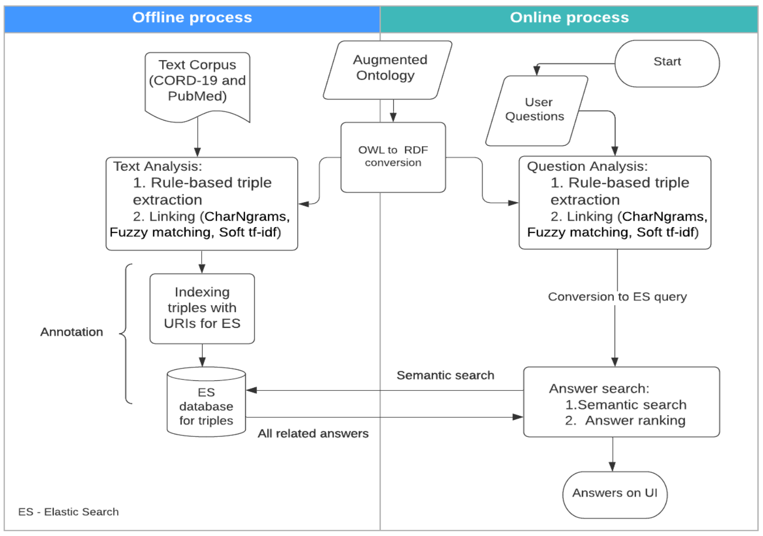

Offline process
Online process
Text Corpus (CORD-19 and PubMed)
Augmented Ontology
Start
User Questions
OWL to RDF conversion
Text Analysis: 1. Rule-based triple extraction 2. Linking (CharNgrams, Fuzzy matching, Soft tf-idf)
Question Analysis: 1. Rule-based triple extraction 2. Linking (CharNgrams, Fuzzy matching, Soft tf-idf)
Indexing triples with URIs for ES
Annotation
Conversion to ES query
Semantic search
ES database for triples
All related answers
Answer search: 1.Semantic search 2. Answer ranking
Answers on UI
ES - Elastic Search

Figure 2: AQuA – Architecture for Question and Answering

These ontologies are used to cover their respective domains but so far, the ontologies are not used to build a QA system or to investigate COVID-19. To produce an ontology meeting the specifications of this domain to obtain the domain coverage we merged and augmented the above-mentioned ontologies. In general, to meet the specifications of a new task, candidate ontologies may be either manually modified or semi-automatically extended by exploiting different types of information sources. This task is done manually for this research work; there are other ways to accomplish this task which are detailed in [46]. The ontologies are merged using the protégé tool. We also added some new concepts and relations along with equivalency axioms where needed to get a taxonomy which sufficiently covers the domain of two factors selected. 'Reasoning' is an important concept while working with ontologies. The reasoner is a software which is used to understand logical consequences from a collection of asserted facts or axioms. We used a Pellet reasoner while merging and augmenting the ontology. Enabling the reasoner before merging the ontologies ensures that the augmenting task is performed properly. For instance, there are no problems arising from definitions of relations, axioms, and equivalence classes; the super class and subclass hierarchy is correct, and the ontology is consistent.

3.2. Data extraction (text corpus)

Data extraction is the process of extracting domain specific textual data, in our case documents, which will be used for creating triples. To this end, (i) we manually extracted some documents from Kaggle - CORD-19 [50] database which are related to all SDoH factors and filtered out the documents related to homelessness and gender from the collected documents manually and (ii) collected more documents related to homelessness and gender from the PubMed library [51].

There are two main processes involved in our QA architecture (shown in Figure 2): ‘Offline Process’ and ‘Online Process’.

3.3. Offline process

In the offline process, we extract triples from the corpus and store them in a database. The corpus used in this process is natural language text documents. This process consists of two major steps: a) Text analysis on text corpus which includes 2 components namely, a rule-based method for extracting triples from sentences and linking. b) ElasticSearch indexing and storing the RDF triples in an ElasticSearch database

Various processes from the NLP pipeline are applied to the corpus to extract triples. In rule-based triple extraction process, pre-processing is the first step to break the documents to words as machines cannot understand the document as is. For this, we used spaCy, pdfminer and NLTK python libraries which perform operations like parsing/analysing the text, tokenization, POS tagging, etc., over pdf files. After pre-processing, we created patterns to recognise entities and relations between entities. Patterns are created for triple extraction. E.g., for entities: [{‘POS’: ‘NOUN’}; {‘POS’: ‘NOUN’}] which means, any noun followed by a noun can together be considered as an entity; example: corona vaccine. For relations: [{‘POS’: ‘VERB’}] which means any verb is considered as relation/predicate; example: in the sentence ‘He ate the mango’, ate (verb) is the relation. The advantages of rule-based information extraction technologies are, they are declarative, easy to comprehend, easy to maintain, easy to in-corporate with domain knowledge and errors are easy to fix [52]. There are also some disadvantages as this process is heuristic and requires tedious manual labour.

Rule-based Information Extraction (IE) such as triple extraction is valued in the commercial world for its interpretability, which makes IE programs easier to adopt, understand, debug, and maintain in the face of changing requirements [52]. Additionally, rule-based IE is valued as it allows researchers/users to incorporate domain knowledge easily. Other methods to perform NER and RE tasks are based on machine-learning or hybrid methods (combining rule-based and machine learning techniques). Though there are some pros for these methods, e.g., Machine learning and hybrid methods are trainable, adaptable, and reduce manual efforts, there are also some drawbacks. For instance, these methods require labelled data, the models need to be re-trained for domain adaption, and they are opaque [52]. Therefore, we adopted rule-based IE in our implementation because the domain (Impacts of SDoH on COVID-19) we choose is still evolving (and new social factors affecting Covid-19 are being discovered since the spread of Covid-19) and the data preparation for this task is tedious and requires some more manual support which makes the machine learning and hybrid methods difficult to implement.

There are many python libraries available to perform various NLP tasks. The fundamental purpose of python-based NLP libraries is to simplify text pre-processing. Also, python offers some powerful libraries for leveraging the power of NLP in research projects. We discuss two of the libraries below. spaCy is a python-based library [53] that has many tools that can be applied for various text processing applications in multiple languages. SpaCy uses deep neural networks and transformers in the background. The spaCy library supports various features like Tokenization, POS-Tagging, Dependency Parsing, NER, Rule-based Matching, Text Classification, Linking, etc. [54]. Natural Language ToolKit (NLTK) is another python-library that enables users to build applications in human languages. NLTK is a library for processing string data, which takes a string as input and the output returned is a single or a list of strings. Although spaCy performs better than NLTK in many tasks, NLTK outperforms spaCy in sentence tokenization (as mentioned in [55]). Therefore, we used NLTK for the sentence tokenization task.

Rule-based triple extraction method is done using human created patterns. We used the spaCy library for this operation. We created patterns manually based on scenarios we were interested in (homelessness and gender integrated with transmission and outcomes of COVID-19). We created patterns to extract subjects, predicates, and objects. We extracted 106599 triples from the filtered corpus using these patterns, some are valid, and some are invalid triples. Let us consider the sentence to understand valid triples "Individuals of low incomes are disproportionately likely to suffer from poor mental health". A valid triple extracted for this sentence, and which makes sense is shown in Table 1. Invalid triples are generated as the corpus have different types of values like special characters, numerical values, headers, footers etc. An example demonstrating an invalid triple is shown in Table 2. These invalid triples are filtered while indexing the triples. This filtering task can also be done using data cleaning method which we did not implement at this time.

Table 1: Valid triple

361	low incomes	suffer	poor mental health	Individuals of low incomes are disproportionately likely to suffer from poor mental health.

Table 2: Invalid triple

1192	¬©	CMAJ	8(4)	DOI:10.9778/cmajo.20200213 ¬© 2020 Joule Inc. orits licensors CMAJ OPEN 8(4) E627 OPENquantifying heterogeneity in, Äúwhat has happened, Äù a process often referred to as an epidemic appraisal.

Linking deals with matching the entities (subject/object) and relations (predicates) recognized earlier with the ontology classes (concepts) and relations. Since we are working with two processes (offline and online), interoperability between these processes is crucial; this is improved by the use of ontologies. For this purpose, we need to compare terms (subjects or predicates or objects) from extracted triples to the ontology concepts and relations using similarity models. In general, concepts and relations in the knowledge base (ontology) have unique identifiers (URIs).The ontologies are converted from OWL file format to RDF format for comparison and mapping using "RDFlib" python library which enables users to perform operations over RDF data. Later we use the same library to convert the RDF file into JSON format so that the ontology concepts and relations along with their respective URIs are indexed and stored in ElasticSearch. This indexing is important for mapping URIs to the terms in triples.

The next step in linking process is to determine the similarity between terms of triples (subjects or predicates or objects) and ontology concepts and relations to map the URIs. To find the similarity between two strings, we built three layers of comparison metrics, CharNgrams, Levenshtein distance/Fuzzy matching, Soft tf-idf. The application finds all the scores and compares them with the defined threshold of 0.9 (to show strong similarity). If any of the three layers produces a match with a score more than the threshold, then the application will map the URI associated with the ontology concepts and relations to the terms of the triples using ElasticSearch. Once the match is found, a keyword-based search is performed using ElasticSearch to fetch the URI and map it to the triple term. This linking process is demonstrated in Figure 3.

ElasticSearch provides an easy way to index documents or triples, enabling users to quickly query the nearest neighbour using a similarity metric. The semantic search (using URIs) is performed over this database to retrieve the answers for the questions in the online process. All the triples which have URIs mapped for subject, and object are filtered out first before indexing. This task is performed to avoid storing lot of data in the database which may hamper the time taken for answer retrieval. After the filtration process, we created URIs for those predicates which do not already have URIs as there was no match with any of the relations in the ontology. To create URIs for the predicate, we used the word as is unlike the regular URIs (Example - The URI created for the word Flourished would be Flourished but not “http://purl.obolibrary.org/obo/Flourished” which is a regular URI pattern). Once we get all the above steps done, all the triples with the mapped URIs are then indexed (using inverted index). The indexed triples are stored as a separate ElasticSearch database. We index and create two databases in our architecture; one is to save ontology concepts and relations with URIs so that this search for mapping URI to the triple term can be facilitated and mapping can be done. The other is to index and store triples, as mentioned above.

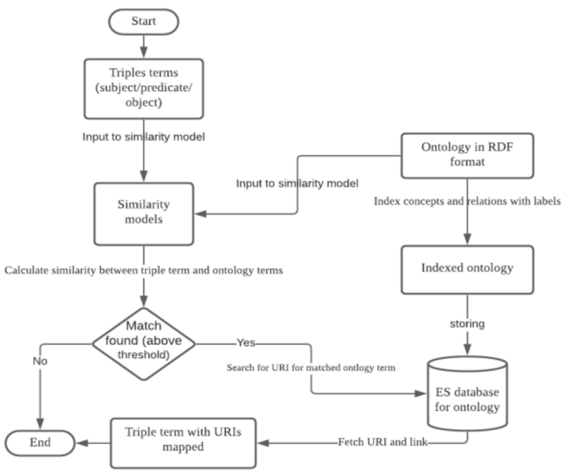

Start
Triples terms (subject/predicate/ object)
Input to similarity model
Ontology in RDF format
Input to similarity model
Similarity models
Index concepts and relations with labels
Indexed ontology
Calculate similarity between triple term and ontology terms
storing
Match found (above threshold)
Yes
No
Search for URI for matched ontlogy term
ES database for ontology
End
Triple term with URIs mapped
Fetch URI and link

Figure 3: Linking process

3.4. Online process

In online process, we apply question analysis to the natural language question provided to the system in a text format (in natural language) and query the database to retrieve the answer. The online process consists of the following components: a) question analysis, consisting of two NLP methods, namely, a rule-based method of extracting triples from natural language questions and linking the URIs of ontology concepts or relation to terms in the triples; b) converting triples with URIs to ElasticSearch query (question conversion); c) answer search

To generate user questions, we studied various resources available for homelessness and gender [56][57][58][59] and numerous other articles from our dataset to understand the domain better and to create questions to test the application. To this end, we investigated the domain and analysed triples created by the rule-based triple extraction method in offline process and picked some valid triples (20 triples) to create 20 questions on them.

The pre-processing task in the online process is applied on user questions which is the only difference from the offline process. We used the same libraries which are used in the offline process for this task in the online process. The rest of the rule-based triple extraction process is same in both online and offline processes. The same patterns we created in the offline process are used in the online process to extract triples from pre-processed questions. The linking process implementation remains same for both the processes with a small change i.e., we apply the same code in indexing and mapping the URIs with triples extracted from questions from offline process, we do not index the ontology file again in online process.

To perform search over ElasticSearch database, we need ElasticSearch query. We used a boolean query in our implementation. A boolean query is a query that matches answers which match boolean combinations of other queries. The boolean query follows the Lucene BooleanQuery syntax [60]. This query uses one or more boolean clauses. Each clause in the query comes with a typed occurrence. The occurrence types are MUST, SHOULD, MUST NOT, FILTER. The score from each matching term mentioned in the 'must or should clause' will be added to provide the final score for each answer. For this research work, we have used MUST and SHOULD clauses. We create an ElasticSearch query from the mapped triple extracted from the user question by appending subject URI, predicate URI, and object URI. Since we have limited of documents in our data, our questions must be confined to data available. For this reason, we first determined valid triples from the documents and then posed questions. In different situations where web scraping is performed, one would not need to restrict the questions to ensure they contain specific triples.

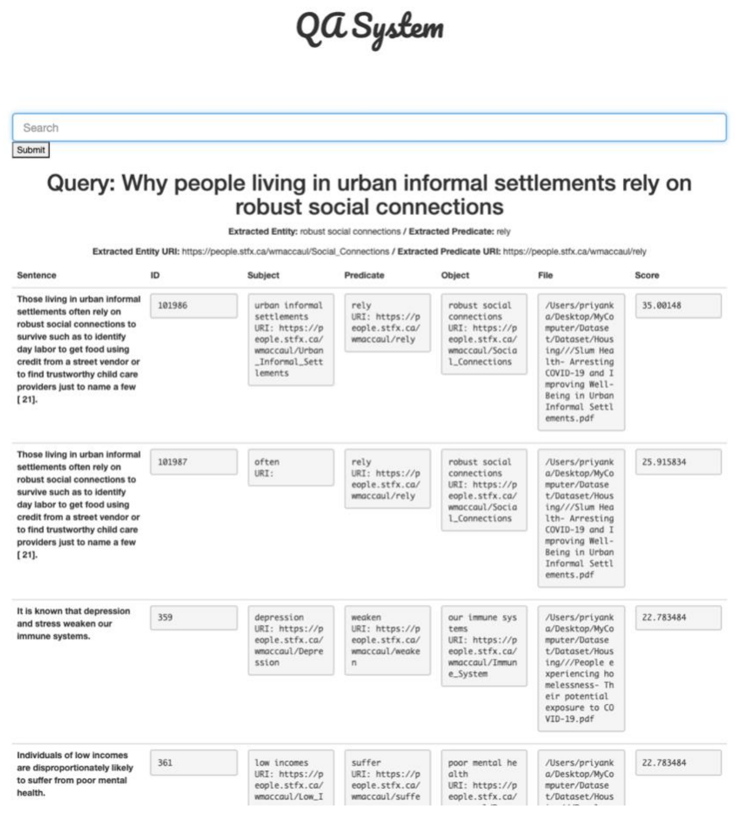
After generating ElasticSearch queries, we conduct an answer search method involving the NLP methods of semantic search and answer ranking. The word semantic refers to meaning in language or logic. The answer retrieval process is based on semantic search. The annotation and translation processes mentioned above allow a semantic search based on our ontology and term frequency (TF-IDF) weighing schemes. This is to find an answer based on user intent rather than matching the keywords in the search. The factors that guide semantic search are [61]: a) The intent of the user and b) The semantic meaning of search terms. As we have incorporated ontology linking with entities in our project, our search is based on URIs. Usage of URIs guarantees interoperability, which ensures the user's intent is carried over throughout the process. This semantic search is facilitated by using ElasticSearch. The system tries to give alternative answers for the query in our implementation rather than no answer based on user intent. TF-IDF weights also help search engines to have a good idea of what words statistically occur together and make semantic correlations for search and fetch results avoiding spams. The pattern in which this search happens using ElasticSearch query and answer is prioritized is as follows: a) Match all subject, predicate, and object in question with answer b) Match any two (Subject-predicate, object-predicate, subject-object) in the question with the answer c) Match at least one (subject/predicate/object) in the question with the answer

The final output from our system would be the best-ranked answer from this method. ElasticSearch uses TF-IDF scoring model for ranking the answers, i.e., addition of TF-IDF scores of all the terms in the answers retrieved that matches the question terms (e.g., if answer has 2 terms that match the question term with scores 0.25 and 0.3, the rank would be 0.55). The ranking is done by adding the scores of the less frequent words in the answer retrieved. For example, in the sentence "people suffers from pneumonia", the words people and from are more frequent and the words suffers, and pneumonia are less frequent. Therefore, the subject of the search would be suffers, and pneumonia and the scores of these words are added for ranking. The answers retrieved are sorted in descending order of their ranks.

3.5. User interface (UI)

We used Flask, which provides useful tools and features to create web applications as it is a lightweight python web framework [62][63]. For this purpose, flask uses Jinja templates to dynamically build HTML pages which uses familiar python concepts like variables, loops, lists, etc. These templates are simple files containing static data as well as placeholders for dynamic data. On the UI, the user can enter a question in the search box and click on submit; the system will provide the top 10 related answers on the screen. Figure 4 demonstrates the page layout and design of our QA system.

There is a search bar in the UI through which user can query the system. The query the user searched for appears on the screen with the retrieved answers with search bar for next question. The system provides the metadata of the answers. The field ID provide us the ID of the triple from all the triples extracted in rule-based triple extraction process. The sentence field contain the sentence from which the triple is extracted. The subject, predicate, and object fields provide the information on triples with their mapped URIs. The file field provides the information about the location at which the file is saved; user can go to that location to access the file if needed. The score field gives the rank of the answers retrieved, which are sorted ranks (highest rank first).From the example shown in Figure 4, the rank of the first answer is highest as the URIs we are searching and the URIs retrieved matches. The URIs in the answers from the second position are partially matched with the URIs of the question. As the URIs of the triples from position 3, on, are same and the search and similarity measures use URIs, the score of the answers from position 3, on, are same. The screenshots for all the 20 questions answered are available at https://drive.google.com/drive/folders/1mXaCG-Pmx9m_NWNp-CKUCGxqy4U4X_qm?usp=sharing.



QA System

Search

Submit

Query: Why people living in urban informal settlements rely on robust social connections

Extracted Entity: robust social connections / **Extracted Predicate:** rely

Extracted Entity URI: https://people.stfx.ca/wmaccaul/Social_Connections / **Extracted Predicate URI:** https://people.stfx.ca/wmaccaul/rely

Sentence	ID	Subject	Predicate	Object	File	Score
Those living in urban informal settlements often rely on robust social connections to survive such as to identify day labor to get food using credit from a street vendor or to find trustworthy child care providers just to name a few [21].	101986	urban informal settlements URI: https://people.stfx.ca/wmaccaul/Urban_Informal_Settlements	rely URI: https://people.stfx.ca/wmaccaul/rely	robust social connections URI: https://people.stfx.ca/wmaccaul/Social_Connections	/Users/priyanka/Desktop/MyComputer/Dataset/Dataset/Housing///Slum Health- Arresting COVID-19 and Improving Well-Being in Urban Informal Settlements.pdf	35.00148
Those living in urban informal settlements often rely on robust social connections to survive such as to identify day labor to get food using credit from a street vendor or to find trustworthy child care providers just to name a few [21].	101987	often URI:	rely URI: https://people.stfx.ca/wmaccaul/rely	robust social connections URI: https://people.stfx.ca/wmaccaul/Social_Connections	/Users/priyanka/Desktop/MyComputer/Dataset/Dataset/Housing///Slum Health- Arresting COVID-19 and Improving Well-Being in Urban Informal Settlements.pdf	25.915834
It is known that depression and stress weaken our immune systems.	359	depression URI: https://people.stfx.ca/wmaccaul/Depression	weaken URI: https://people.stfx.ca/wmaccaul/weaken	our immune systems URI: https://people.stfx.ca/wmaccaul/Immune_System	/Users/priyanka/Desktop/MyComputer/Dataset/Dataset/Housing///People experiencing homelessness- Their potential exposure to COVID-19.pdf	22.783484
Individuals of low incomes are disproportionately likely to suffer from poor mental health.	361	low incomes URI: https://people.stfx.ca/wmaccaul/Low_I	suffer URI: https://people.stfx.ca/wmaccaul/suffe	poor mental health URI: https://people.stfx.ca/	/Users/priyanka/Desktop/MyComputer/Dataset/Dataset/Hous	22.783484

Figure 4: User Interface

The domain SDoH is very vast, and the user will have many other questions regarding the impacts of the domain on COVID-19. Our QA system can answer the questions only for the questions from the dataset we have used for this implementation. As we used some portion of the CORD-19 dataset for this implementation, there are some more steps that need to be done for the system to answer a question out of its range (from outside the dataset). They are: a) Gather data with information related to domain and query. b) Extract triples from data using the rule-based triple extraction model (may need more triple patterns). c) Though we have built an ontology related to Homelessness and Gender, users may have questions involving other SDoH factors. For this, the user needs to create ontology classes/relations/instances/axioms according to the requirements. d) Index the triples extracted in step b and store them in ElasticSearch database. The user can then go to the UI and submit the question.

4. EVALUATION

This section describes the evaluation metrics used in measuring the performance of the system, which are the same ones used in IR [64][65]. The implementation is evaluated on the system's ability to correctly retrieve and rank answers for a given question. As our domain for this implementation is novel, there are no benchmark values for these metrics. For a given set of relevant answers also referred to as 'ground truth positive' and a set of answers retrieved by the system also referred to as 'predicted', precision is calculated as per Equations (5).

$$\text{Precision} = \frac{|\{\text{relevant answers}\} \cap \{\text{retrieved answers}\}|}{|\{\text{retrieved answers}\}|} \tag{5}$$

Precision takes all retrieved answers into account [64] giving the fraction of retrieved answers that are relevant. It can also be evaluated considering only the topmost results (Top 5 or top 10 answers for one question) returned by the system using precision@k (p@k), where k is the kth ranked answer retrieved by the system for a question q.

Recall is another metric which is often called sensitivity [64]. It is the probability that a relevant answer is retrieved by the query. Recall is calculated as per Equation (6).

$$\mathrm{Recall} = \frac{|\{\mathrm{relevant\ answers}\} \cap \{\mathrm{retrieved\ answers}\}|}{|\{\mathrm{relevant\ answers}\}|} \qquad (6)$$

F-measure is a single measure that trades off precision and recall, it is defined as the weighted harmonic mean of precision and recall. This measure gives the accuracy of the model and is calculated as Equation (7).

$$\mathrm{F-measure} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \qquad (7)$$

The above-mentioned measures (precision, recall and F-measure) do not consider the order in which the returned answers are presented, rather they consider only if the relevant answers are retrieved (or not). Whereas the metric precision@k (p@k) is used to determine relevancy of the system by considering the topmost answers retrieved.

The Average Precision (AP) is a metric that gives a good measurement of quality, i.e., the ability of the model to appropriately sort the results of a query. AP is calculated as Equation (8).

$$\mathrm{AP} = \frac{1}{\mathrm{GTP}} \sum_{j=1}^{N} \mathrm{p@j} \times \mathrm{rel@j} \qquad (8)$$

where N is the total number of positions i.e., the total number of results returned.

For a query, AP@k, the average precision till the kth position in the ranked list, is calculated as in Equation (9).

$$\mathrm{AP@k} = \frac{1}{\mathrm{GTP}} \sum_{j=1}^{k} \mathrm{p@j} \times \mathrm{rel@j} \qquad (9)$$

where, rel@k is the indicator function which equals to 1 if answer at rank k is relevant, 0 otherwise and GTP is the number of ground truth positives. The result of this metric will be '1' if the answers are sorted correctly and all the relevant answers appear at the top of the ranked list. Remark: in general, AP@N = AP where N is the last position.

The metrics explained above (Precision, Recall, F-measure, and Average Precision), are based on single question. The metrics: Mean Average Precision, Mean Average Precision@k and system Precision@k, involve all the queries.

The Mean Average Precision, mAP, for a set of queries $\{q_i \mid 1 \leq i \leq Q\}$ is the mean of the AP over all queries q_i; it is given in Equation (10).

$$\mathrm{mAP} = \frac{1}{Q}\sum_{i=1}^{Q} \mathrm{AP}(q_i) \tag{10}$$

The mAP@k, mean average precision at the kth position over all queries q_i, is calculated as in Equation (11).

$$\mathrm{mAP@k} = \frac{1}{Q}\sum_{i=1}^{Q} \mathrm{AP@k}(q_i) \tag{11}$$

We use mAP@k to evaluate our project as it is the most popular metric to evaluate the performance of any IR systems. mAP@k is a metric which gives a single-figure measure of quality across all queries at the kth position [66].

The system precision@k, sp@k, is the averaged precision over all the answers retrieved of all queries till the kth position. This metric is calculated as Equation (12).

$$\mathrm{sp@k} = \frac{\sum_{i=1}^{Q}\sum_{j=1}^{k} \mathrm{p@j}(q_i)}{|\{\text{total retrieved answers}\}|} \tag{12}$$

This metric gives the overall performance of the system till the kth position.

4.1. Results

We calculated p@k for all the positions; Figure 5 shows how the ranking quality of the application is gradually decreasing as the rank of the answer is increasing. Therefore, we decided to show the top 10 answers on the UI for any question as the graph is gradually falling off after position 10.

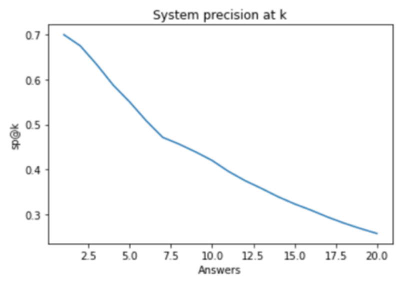


Figure 5: System Precision@k

We used the system precision calculated above along with the recall (which is manually calculated) to obtain F-measure, which gives the accuracy of overall system for top 5 answers retrieved for 20 questions, top 10 answers retrieved for 20 questions, top 15 answers retrieved for 20 questions and top 20 answers retrieved for 20 questions. The Table 3 shows the evaluated results. From the obtained output of F-measure, it is clear that the system's accuracy is increasing till the top 10 answers and falling off from answers 11-20.

Table 3: System Precision, Recall and F-measure

Top n answers	System precision	Recall	F-measure
5	0.55	0.495	0.5298
10	0.419	0.773	0.56644
15	0.3233	0.895	0.495937
20	0.2574	0.947	0.42261526

We evaluated the quality of the system using the mAP@k metric. We conducted these experiments for mAP at each kth position using 'Information Retrieval (IR) Effectiveness Evaluation Library for Python' [68]. This library takes two files as input; one is ground truth (relevant) file, and another is predicted (retrieved) file. We manually analysed the answers and determined which were correct for predicted file and then labeled the data with relevant answer and non-relevant answer to get the ground truth file (for the questions under consideration). The results obtained for mAP@k are shown the graph in Figures 6.

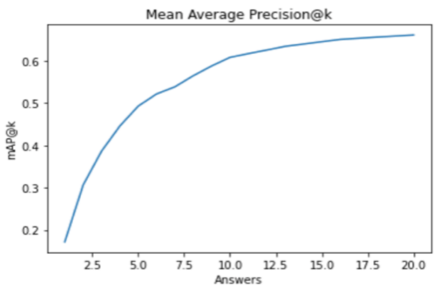


Figure 6: Mean Average Precision@k

Figure 6 shows that mAP@k is increasing indicating the quality of the system is increasing for each question as the number of relevant answers are being retrieved by the system are increasing. Figure 6 shows that mAP@10 i.e., the performance of the system for 20 questions and top 10 answers each is 0.60.

5. Conclusions and Future Works

In this research, we developed a prototype QA system so that people can get information from a large corpus for questions involving SDoH without reading all the documents. The QA system developed is the integration of semantic web technologies and IR systems. Innovative aspects for this work include using ElasticSearch, which is the most widely used IR systems, instead of SPARQL for querying the database, and creating a domain specific ontology which supports this QA system. To meet the needs of an interdisciplinary audience, we present an overview of various methods involved in the implementation of QA systems such as: how to collect or create domain-specific ontologies, how to collect the domain-specific data, how to create RDF triples using a rule-based approach, how to perform search over RDF triples using ElasticSearch, how to index an RDF dataset, how to rank the data, what data should be ranked and how to evaluate an IR-based QA system using some popular metrics like Mean Average Precision and System Precision. Our system performance can be improved if the future works discussed below are integrated with the existing system. Also, this QA system can be easily adopted to any domain by creating domain specific patterns, collecting domain specific documents and domain specific knowledge graphs or ontologies.

From our literature review on the SDoH domain, we determined that social determinants of health have a major impact on a crisis like COVID. Our goal through this application is to show that we can leverage the existing research and literature on COVID to help decision-makers manage health emergencies so that everyone has an equal opportunity to stay healthy. In future, the ontology we created can be enhanced by adding other factors (other than homelessness and gender) to cover the entire domain of SDoH. We will further investigate how to improve the overall performance of the system by implementing more semantic similarity metrics like cosine similarity for linking and query expansion. Query expansion using ontology parent-child and sibling relations will improve results. An appropriate axiom schema to further enhance the semantics of our search must be developed and integrated into our system. This requires interactions with clinicians, epidemiologists, and other public health professionals to further leverage queries on the existing research and literature on COVID-19. Data relevant to queries to cover all SDoH factors is also needed. Other future work includes data cleaning applied to triple extraction process and getting this application evaluated by clinicians.

REFERENCES

[1] Dimitrakis, E., Sgontzos, K. and Tzitzikas, Y., 2020. A survey on question answering systems over linked data and documents. Journal of Intelligent Information Systems, 55(2), pp.233-259.

[2] Wang, W., Auer, J., Parasuraman, R., Zubarev, I., Brandyberry, D. and Harper, M., 2000. A question answering system developed as a project in a natural language processing course. In ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems.

[3] Kolomiyets, O. and Moens, M.F., 2011. A survey on question answering technology from an information retrieval perspective. Information Sciences, 181(24), pp.5412-5434.

[4] Athenikos, S.J. and Han, H., 2010. Biomedical question answering: A survey. Computer methods and programs in biomedicine, 99(1), pp.1-24.

[5] Gupta, P. and Gupta, V., 2012. A survey of text question answering techniques. International Journal of Computer Applications, 53(4).

[6] Kalyanpur, A., Boguraev, B.K., Patwardhan, S., Murdock, J.W., Lally, A., Welty, C., Prager, J.M., Coppola, B., Fokoue-Nkoutche, A., Zhang, L. and Pan, Y., 2012. Structured data and inference in DeepQA. IBM Journal of Research and Development, 56(3.4), pp.10-1.

[7] Soares, M.A.C. and Parreiras, F.S., 2020. A literature review on question answering techniques, paradigms and systems. Journal of King Saud University-Computer and Information Sciences, 32(6), pp.635-646.

[8] Chan, H.Y. and Tsai, M.H., 2019. Question-answering dialogue system for emergency operations. International Journal of Disaster Risk Reduction, 41, p.101313.

[9] Covid-19 - what we know so far about...social determinants of health, https://www.publichealthontario.ca/-/media/documents/ncov/covid-wwksf/2020/05/what-we-know-social-determinants-health.pdf?la=en, [Online; accessed 19-June-2021].

[10] Five big lessons experts say canada should learn from covid-19, https://www.ctvnews.ca/health/coronavirus/five-big-lessons-experts-say-canada-should-learn-from-covid-19-1.5282125, [Online; accessed 19-June-2021].

[11] Singu, S., Acharya, A., Challagundla, K. and Byrareddy, S.N., 2020. Impact of social determinants of health on the emerging COVID-19 pandemic in the United States. Frontiers in public health, 8, p.406.

[12] Abacha, A.B. and Zweigenbaum, P., 2015. MEANS: A medical question-answering system combining NLP techniques and semantic Web technologies. Information processing & management, 51(5), pp.570-594.

[13] Introduction to natural language processing (nlp), https://www.kdnuggets.com/2019/10/introduction-natural-language-processing.html, [Online; accessed 23-May-2021].

[14] Wikipedia contributors. Question answering. Wikipedia, The Free Encyclopedia. August 14, 2021, 06:46 UTC. Available at: https://en.wikipedia.org/w/index.php?title=Question_answering&oldid=1038707361. Accessed August 26, 2021.

[15] Prager, J., 2021. Question answering. In The Oxford Handbook of Computational Linguistics 2nd edition.
[16] Srinivasa-Desikan, B., 2018. Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras. Packt Publishing Ltd.
[17] Wikipedia contributors. Hidden Markov model. Wikipedia, The Free Encyclopedia. August 5, 2021, 04:02 UTC. Available at: https://en.wikipedia.org/w/index.php?title=Hidden_Markov_model&oldid=1037203864. Accessed August 26, 2021.
[18] Different ways of doing relation extraction from text, https://medium.com/@andreasherman/different-ways-of-doing-relation-extraction-from-text-7362b4c3169e, [Online; accessed 03-July-2021].
[19] Wikipedia contributors. Ontology. Wikipedia, The Free Encyclopedia. August 21, 2021, 13:25 UTC. Available at: https://en.wikipedia.org/w/index.php?title=Ontology&oldid=1039903261. Accessed August 26, 2021.
[20] Rubin, D.L., Noy, N.F. and Musen, M.A., 2007. Protege: a tool for managing and using terminology in radiology applications. Journal of digital imaging, 20(1), pp.34-46.
[21] Introduction to ontology, https://www.ontotext.com/knowledgehub/fundamentals/what-are-ontologies/#:~:text=An%20ontology%20is%20a%20formal,relationships%20that%20hold%20between%20them.&text=As%20a%20result%2C%20ontologies%20do,new%20knowledge%20about%20the%20domain, [Online; accessed 18-June-2021].
[22] What is an rdf triplestore, https://www.ontotext.com/knowledgehub/fundamentals/what-is-rdf-triplestore/, [Online; accessed 10-July-2021].
[23] What's a uri and why does it matter?, http://www.ltg.ed.ac.uk/~ht/WhatAreURIs/#:~:text=URI%20stands%20for%20Uniform%20Resource,the%20World%20Wide%20Web%20consortium., [Online; accessed 10-July-2021].
[24] Rdf 1.1 primer, https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/, [Online; accessed 10-July-2021].
[25] Introduction to semantic web, https://graphdb.ontotext.com/documentation/standard/introduction-to-semantic-web.html, [Online; accessed 18-June-2021].
[26] Olsson, J., 2019. Using Elasticsearch for full-text searches on unstructured data.
[27] Kadilierakis, G., Fafalios, P., Papadakos, P. and Tzitzikas, Y., 2020, May. Keyword search over RDF using document-centric information retrieval systems. In European Semantic Web Conference (pp. 121-137). Springer, Cham.
[28] Elasticsearch as an ir tool, https://colab.research.google.com/github/fastforwardlabs/ff14_blog/blob/master/_notebooks/2020-06-30-Evaluating_the_Retriever_&_End_to_End_System.ipynb#scrollTo=v1MjzT9zlTrf,[Online; accessed 10-June-2021].
[29] Berglund, P., 2014. Shard Selection in Distributed Collaborative Search Engines A design, implementation and evaluation of shard selection in ElasticSearch.
[30] Wu, G., He, Y. and Hu, X., 2018. Entity linking: an issue to extract corresponding entity with knowledge base. IEEE Access, 6, pp.6220-6231.
[31] The levenshtein-algorithm, http://www.levenshtein.net/, [Online; accessed 12-June-2021].
[32] Article on charngram - pytorch, https://pytorchnlp.readthedocs.io/en/latest/_modules/torchnlp/word_to_vector/char_n_gram.html, [Online; accessed 16-June-2021].
[33] Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T., 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, pp.135-146.
[34] Green Jr, B.F., Wolf, A.K., Chomsky, C. and Laughery, K., 1961, May. Baseball: an automatic question-answerer. In Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference (pp. 219-224).
[35] Woods, W.A. and WA, W., 1977. Lunar rocks in natural English: Explorations in natural language question answering.
[36] Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J. and Schlaefer, N., 2010. Building Watson: An overview of the DeepQA project. AI magazine, 31(3), pp.59-79.
[37] Song, S., Zhang, X. and Qin, G., 2017. Multi-domain ontology mapping based on semantics. Cluster Computing, 20(4), pp.3379-3391.

[38] Anam, S., Kim, Y.S., Kang, B.H. and Liu, Q., 2016, February. Adapting a knowledge-based schema matching system for ontology mapping. In Proceedings of the Australasian Computer Science Week Multiconference (pp. 1-10).

[39] Pham, M., Alse, S., Knoblock, C.A. and Szekely, P., 2016, October. Semantic labeling: a domain-independent approach. In International Semantic Web Conference (pp. 446-462). Springer, Cham.

[40] Ramnandan, S.K., Mittal, A., Knoblock, C.A. and Szekely, P., 2015, May. Assigning semantic labels to data sources. In European Semantic Web Conference (pp. 403-417). Springer, Cham.

[41] Cheng, G. and Qu, Y., 2009. Searching linked objects with falcons: Approach, implementation and evaluation. International Journal on Semantic Web and Information Systems (IJSWIS), 5(3), pp.49-70.

[42] Liu, X. and Fang, H., 2010, April. A study of entity search in semantic search workshop. In Proc. of the 3rd Intl. Semantic Search Workshop.

[43] Delbru, R., Campinas, S. and Tummarello, G., 2012. Searching web data: An entity retrieval and high-performance indexing model. Journal of Web Semantics, 10, pp.33-58.

[44] Ilievski, F., Beek, W., van Erp, M., Rietveld, L. and Schlobach, S., 2016, May. LOTUS: Adaptive text search for big linked data. In European Semantic Web Conference (pp. 470-485). Springer, Cham.

[45] Johnson, T., 2013. Indexing linked bibliographic data with JSON-LD, BibJSON and Elasticsearch. Code4lib Journal, (19).

[46] Fernandez, M., Zhang, Z., Lopez, V., Uren, V. and Motta, E., 2011, June. Ontology augmentation: combining semantic web and text resources. In Proceedings of the sixth international conference on Knowledge capture (pp. 9-16).

[47] Dutta, B. and DeBellis, M., 2020. CODO: an ontology for collection and analysis of COVID-19 data. arXiv preprint arXiv:2009.01210.

[48] Homelessness and clinical data recording, https://bioportal.bioontology.org/ontologies/HCDR/?p=summary, [Online; accessed 19-June-2021].

[49] Gsso, https://bioportal.bioontology.org/ontologies/GSSO, [Online; accessed 19-June-2021].

[50] Wang, L.L., Lo, K., Chandrasekhar, Y., Reas, R., Yang, J., Eide, D., Funk, K., Kinney, R., Liu, Z., Merrill, W. and Mooney, P., 2020. Cord-19: The covid-19 open research dataset. ArXiv.

[51] Pubmed library - national library of medicine, https://pubmed.ncbi.nlm.nih.gov/?term=covid-19+homeless+canada&filter=simsearch3.fft, [Online; accessed 21-June-2021].

[52] Chiticariu, L., Li, Y. and Reiss, F., 2013, October. Rule-based information extraction is dead! long live rule-based information extraction systems!. In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 827-832).

[53] Honnibal, M. and Montani, I., 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 7(1), pp.411-420.

[54] spacy 101: Everything you need to know, https://spacy.io/usage/spacy-101, [Online; accessed 16-June-2021].

[55] Introduction to libraries of nlp in python - nltk vs. spacy, https://medium.com/@akankshamalhotra24/introduction-to-libraries-of-nlp-in-python-nltk-vs-spacy-42d7b2f128f2, [Online; accessed 16-June-2021].

[56] Social determinants of health, https://www.healthypeople.gov/2020/topics-objectives/topic/social-determinants-of-health, [Online; accessed 30-April-2021].

[57] Braveman, P. and Gottlieb, L., 2014. The social determinants of health: it's time to consider the causes of the causes. Public health reports, 129(1_suppl2), pp.19-31.

[58] Nchhstp social determinants of health, https://www.cdc.gov/nchhstp/socialdeterminants/faq.html, [Online; accessed 30-April-2021].

[59] Artiga, S., Orgera, K. and Pham, O., 2020. Disparities in health and health care: Five key questions and answers. Kaiser Family Foundation.

[60] Boolean query, https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-bool-query.html, [Online; accessed 18-June-2021].

[61] Semantic search, https://blog.alexa.com/semantic-search/, [Online; accessed 18-June-2021].

[62] Building modern user interfaces with ask, https://ckraczkowsky.medium.com/building-modern-user-interfaces-with-flask-23016d453792, [Online; accessed 22-June-2021].

[63] How to make a web application using ask in python3, https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3, [Online; accessed 22-June-2021].

[64] Wikipedia contributors. Evaluation measures (information retrieval). Wikipedia, The Free Encyclopedia. July 30, 2021, 08:39 UTC. Availableat: https://en.wikipedia.org/w/index.php?title=Evaluation_measures_(information_retrieval) &oldid=1036232908. Accessed August 26, 2021.
[65] Breaking down mean average precision (map), https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52, [Online; accessed 21-June-2021].
[66] Evaluation in information retrieval, https://nlp.stanford.edu/IR-book/pdf/08eval.pdf, [Online; accessed 21-June-2021].
[67] Improving entity resolution with soft tf-idf algorithm, https://medium.com/enigma-engineering/improving-entity-resolution-with-soft-tf-idf-algorithm-42e323565e60, [Online; accessed 16-June-2021].
[68] Information retrieval (ir) effectiveness evaluation library for python, https://pypi.org/project/ir-evaluation-py/, [Online; accessed 22-June-2021]

AUTHORS

Priyanka Addagudi is a Graduate Research Assistant at St Francis Xavier University, Canada. Her interest areas and research work include building information retrieval-based QA system which involves semantic search engines, knowledge graphs, NLP, modern deep learning architectures, and statistical language modeling.

Wendy MacCaull is a Senior Research Professor in Computer Science at St Francis Xavier University, Canada. Her research interests include knowledge representation and reasoning, ontologies, non-classical logics, and workflow systems with applications to healthcare.