DYNAMIC ONTOLOGY BASED Q&A SYSTEM FOR PANDEMIC SITUATIONS CASE STUDY COVID – 19 PANDEMICS

Subasinghe Arachchige Harsha Piyumi Subasinghe

(189397P)

Thesis submitted in partial fulfilment of the requirements for
the degree of Master of Science in Artificial Intelligence

Department of Computational Mathematics

University of Moratuwa
Sri Lanka

July 2022

Declaration

I declare that this dissertation does not incorporate, without acknowledgment, any material previously submitted for a degree or a Diploma in any University and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and summary to be made available to outside organization.

Name of the Student

S A H P Subasinghe Signature of Student

Date:

Supervised by

Dr. A T P Silva Signature of Supervisor(s)

Date:

Dedication

I dedicate my dissertation work to my family. A special feeling of gratitude to my loving parents, whose words of encouragement and push for tenacity ring in my ears.

Acknowledgement

I would first like to thank my supervisor, Dr. A T P Silva of the faculty of Information Technology at University of Moratuwa, for the patient guidance, encouragement, and advice she has provided. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly.

I would also like to gratefully acknowledge and thank my all colleagues and coworkers, at various stages of this work.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Abstract

In dynamic pandemic situations like covid-19, Many writeups, reviews, articles have been published every day. Rapidly updated data leads information overload, which make the public difficult to keep up with the latest data on pandemic situation. This paper focuses on introduce an efficient Q&A system for dynamic pandemic situation which help public to update with the real time data.

Several approaches including basic ontologies, expert knowledge base and linguistic knowledge have been used when model the knowledge base of Q&A systems. But these approaches are mainly based on experts' knowledge and mainly human interaction in knowledge acquisition, less handling of multimodal data, inefficient inferencing. Even though there are number of solutions which help public to update with the pandemic data, there are no fully automated real time updated systems. So, the intention is to introduce a fully automated multimodal data based real time updated system.

In order to archive this goal, fully automated dynamic ontology-based Q&A system was design, developed and evaluated for the pandemic situation like covid-19. Solution was design in such a way that users can enter question which is related to the covid-19 pandemic and retrieve a real time answer. Mainly the system is based on two modules as dynamic ontology module which use web scrapping for real time updated data extraction, process to map the changes in data and Q&A module which simplifies the questions into RDF triples based normal forms that effortlessly handled by database querying.

Evaluation of the system was conducted two ways by evaluation of the dynamic ontology module and evaluation of the question and answer module. In both evaluation processes time evaluation and precision has considered.

Keywords:

covid-19, dynamic ontology, Q&A, normal form, web scrapping

Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Prolegomena

Many reviews, news articles, web sites have been and continue to publish every day for a pandemic situation like COVID-19. Because of that, there is difficult for the public to update with the latest data. Developing Question Answering (QA) systems to automatically answering for natural language questions, has been a long-standing research problem since the beginning of AI [1]. QA systems are rapidly growing and evolving field of research, new ideas are being implemented continuously with success [2]. There are main characteristics in QA systems such as system domain, question type, and information source. System domain can be either closed domain (which accept questions only from a specific domain) or open domain (which do not have this limitation), Question type which describes the type of the questions which can be factoid/non-factoid, where factoid questions have simple answers with one/more words which gives the precise answer of the question and non-factoid which require complex answers, like descriptions, opinions, or explanations, which are mostly passage-level texts, Information source which describes the source of information the QA system uses to generate the answer this can be either document or knowledgebase. This system is based on closed domain, factoid question type, and knowledgebase as the information source.

Knowledge Modeling is a cross-disciplinary approach to capture and model, knowledge into a reusable format, for purpose of preserving, improving, sharing, substituting, aggregating, and reapplying it. In the computer world knowledge, modeling is used to simulate intelligence. Knowledge modeling is an important step in building knowledge-based applications [1]. According to knowledge base modeling and handling, fundamental theories of knowledge bases has categorized as 4 groups such as, expert knowledge bases, linguistic knowledge bases, expert knowledge bases, ontology most recently the cognitive knowledge base. Linguistic knowledge base is a collection of word association norms, frame semantics and common-sense knowledge.

The expert knowledge base is consisting of binary logic and fuzzy set. Ontology is based on the nature of being and cognitive knowledge base has implemented using the concept of human knowledge as fundamentals[1].

In dynamic ontology, based knowledge base structure is a collection of other three categories structure. dynamic ontology consists with a linguistic knowledge base, logical model, and object attribute relation, such as values, attribute, and relations of concepts. Knowledge acquisition in dynamic ontology is fully automated, and different from expert knowledge bases, linguistic knowledge bases as well basic ontologies. Because of this reason, the knowledge base using dynamic ontology is more suitable when dealing with rapidly expanding data and automate data extraction. This system is based on a dynamic ontology-based Q&A system and which access rapidly expanding multi-model data which extracted from newspaper articles, news, twitters, articles, documents, etc.

1.2 Aim and objectives

1.2.1 Aim

The aim of this research is to design and develop a dynamic ontology-based Q&A system for pandemic situation.

1.2.2 Objectives

- Critically review the literature related to Q&A systems and dynamic ontology.
- Critical review the literature related to semantic knowledge modeling using dynamic ontology.
- Design a dynamic ontology which integrate pandemic records from multiple data sources.
- Design Q&A engine that extract related facts, process data and to answer intelligent queries.
- Develop Q&A system based on the designed dynamic ontology.

- Evaluate the proposed dynamic ontology and developed system for its effectiveness.

1.3 Background and Motivation

Dynamic ontology-based knowledge acquisition is one of the fields which emerge during the last few years. With the Increase of rapidly expanding data, many systems tend to handle those data in an effective manner. Recently there are number of research has been conducted in this area; including IoT[2], information modeling [3]and many more.

With the rapid development in Question answering systems and the increase in interest of developing dynamic ontology-based systems; leads to need of a system which can effectively deal with rapidly expanding multi model data.

1.4 Problem in Brief

Even though there are number of research have been done in this field there are no fully automated real-time updated systems with the exponentially growing information of pandemic situations like covid-19. As a result of that, the public hardly updates with the recent information. Efficient Q&A system for dynamic pandemic situations is a current need.

1.5 Dynamic Ontology-based Q&A Solution for Pandemic Situations

In order to archive the objectives, define in the previous section a Q&A system for a pandemic situation like covid-19 based on dynamic ontology was proposed as a solution. This contains a number of modules to facilitate a number of features and functionalities.

1.6 Structure of the Thesis

Rest of the thesis is organized as follows. Chapter 2 provides a literature review on dynamic ontology-based systems and presents the research problem and identifies the research problem/gap and the possible technology to solve the problem. Chapter 3 is on the technologies including solving the problem. Chapter 4 present the approach with hypothesis, inputs, outputs, users, process, and features. Chapter 5 gives the top-level design of the solution. Chapter 6 presents the implementation of the design. Chapter 7 is about the evaluation of the system. Chapter 8 concludes the research findings with a note on further work.

1.7 Summary

In this chapter I have given introduction to the research. This chapter covered Prolegomena, Background and motivation, problem in brief, Objectives, Proposed Solution, and structure of the thesis. In chapter 2 I discuss the Evolution and State of art of dynamic ontology-based systems.

CHAPTER 2: EVOLUTION AND STATE OF ART OF Q&A SYSTEMS

2.1 Introduction

In chapter 1, introduced the overall project. This chapter presents our critical review of research on development in dynamic ontology-based systems. This chapter has structured under several headings, namely, early development in dynamic ontology-based systems, breakthroughs in dynamic ontology-based systems research, modern development in dynamic ontology-based systems, challenges in knowledge modeling, and problem definition.

Question answering systems based on knowledge base is a challenging and important process with a wide range of applications which use information retrieval and natural language processing. Knowledgebase is interlinked entities collection which enables analyze, storage, reuse the knowledge in machine interpretable way. Task of a knowledgebase is, represents relevant knowledge in specific application domain. Knowledge modeling is the knowledgebase creation process. It is a cross disciplinary approach which capture and model the knowledge into a reusable format for improving, substituting, preserving, sharing, aggregating the knowledge. In the computer world, it is used to simulate intelligence. Knowledge modeling is an important step in building knowledge-based applications. First-generation knowledge-based systems are based on a set of standard reasoning procedures which use declarative representations (Ex: rules, frames). Next-generation knowledge-based systems abstract from symbolic representation considering the design process and evolved to the paradigm of model-based system development [2]. According to knowledgebase modeling and manipulation fundamental theories, knowledge base technology can categorize into four groups such as, linguistic knowledge bases, expert knowledge bases, ontology and most recently the cognitive knowledge base.

2.2 Early Development in Q&A Systems

Early Development in Q&A Systems based on linguistic knowledge bases. Linguistic knowledge base is a combination of word association norms, common-sense knowledge and Frame semantics [1]. FrameNet, WordNet and ConceptNet are typical linguistic knowledge bases. FrameNet knowledge base structure is based on relation between frames, which represent the knowledge as frames and annotated corpus. Frames are based on generalizations over group of words. WordNet based on word association norms which given a lexical stimulus like noun, verb/adjective and responses often remain in specific semantic relations such as synonyms(similar), antonyms(opposite), hyponyms(subordinate)/hypernyms(superior) and meronyms (part)/holonyms(whole). ConceptNet based on semantic graphs, which describe human knowledge and how it express. Knowledge Representation of these knowledge bases uses frame elements, semantic network and semantic graphs and knowledge acquisitions is mostly manual with some automated processes such as Lexicosyntactic Pattern Extraction (LPSE) and acquiring data from Open Mind Common Sense (OMCS) corpus. Linguistic knowledge bases suitable for retrieval and extraction of information using natural language processing.

Simplify linguistic knowledge bases in order to simultaneously improve interoperability and accuracy has been discovered with database reduction and rule base simplification. In this system final knowledgebase resulted high interoperable and accurate with the covered examples [4].

Linguistic knowledge bases are depending on volatile experts' knowledge, difficulties with handling text coherence, issues with link arguments across sentences, expensive to expand and build, some of them have shallow coverage of knowledge than human knowledge.

Expert Knowledge Base use of binary Logic and fuzzy set. knowledge representation of expert knowledge bases is based on IF THEN rules. Knowledge Acquisition commonly use a manual process though domain experts using questionnaire or interviews and can be data driven. There are limitations with these knowledge bases

such as, difficulty of handling experts' knowledge, rules and large rule base as well as inference efficiency problems [1].

Expert Knowledge Base for radar system maintenance describe a methodology to recognize object state and update the user regarding the object in each state. A diagnostic and functional analysis of the object using of an artificial neural network using vector base analysis has conducted for this purpose. Even though the accuracy is high with the system, diagnostic information which transferred to set of servicing information, is based on experts' knowledge [5].

Mainly, in logical rule-based systems, knowledge acquires from domain experts by interviews, communicating their knowledge by questionnaires manually. Knowledge can be acquired automatically with rules. Such as rule-based information retrieval by computer (RUBRIC) which constructs rules from thesauri, shows automatically constructed rules are more effective than hand-made rules in terms of precision [6].

Semi-automated systems such as KnowRob, automatically obtain information by different knowledge sources, with the aid of aligning imported knowledge sources and human for correcting mistakes [7]. The knowledge acquisition process greatly depends on domain expert knowledge. Expert knowledge sometimes can be certain and uncertain, precise and imprecise or complete and incomplete. Getting an expert's knowledge is more difficult since most of expert's knowledge is hidden with their skills [1].

Fuzzy rule-based systems use fuzzy sets use for representing knowledge. In fuzzy rule-based system, consequent could be partially true, if rule antecedent is true, which differs from rule-based systems. Knowledge acquisition in fuzzy systems can be done by data-driven (which identify parameters with structure of fuzzy models, by datasets using different methods like C4. 5 classification tree and feature space mapping, fuzzy rule learning algorithm, fuzzy scheduler, swam intelligence approach, differential evolution learning, and genetic algorithm) or human experts (which knowledge retrieved from experts through interviews or open questions) [8].

As well early developments of question and answering systems were based on linguistic approaches. This integrates the natural language processing(NLP)

techniques and knowledge information are based on frames, logical, templates with these systems data handling was difficult since knowledge bases designed for handle prestored data, scalability is quite complex because of foe every new concept , rule should be introduced [9].

2.3 Breakthrough in Q&A Systems

Breakthrough of question and answering systems are based on statistical approach, which statistical learning is the key process and involves ontological knowledge acquisition. This approach can dela with large amounts of data as well their heterogeneity. Statistical techniques such as support vector machine (SVM) classifiers, Maximum entropy, Bayesian classifiers has used. This approach is suitable for complex non factoid, shallow and factoid question types. Mostly use for handling large, trained data and uses supervised approach [9].

Ontologies are based on classes, relations and instances. Knowledge Representation of ontologies use taxonomy of concepts with attributes, relations and values. Knowledge Acquisition of an ontology mostly acquired manually by domain experts or knowledge engineers. In ontology there are limitations such as, difficulties of take experts' knowledge and lack of generalized and sufficiently validate methodologies [1] [22] [23] [24].

Knowledge acquisition in application ontologies is based on eliciting reasoning mechanisms used by experts in order to do a task or solve a problem. In order to elicit knowledge from technical reports, documents, expert interviews which merge with inferential modeling technique that supports the knowledge engineer for identifying various knowledge types, Semi-automated techniques used.

With the rapidly expanding data ontological systems also evolve. Most of ontologies depends on user inputs. ontologies have evolved with the usage of external background knowledge sources rather than depending on user inputs [8].

Information system based on ontology (OBIS), with web based information analyzing and fully functional, the system can use for real world applications with larger data but the issues with handling individuals on ontology classes with SPARQL quires [10].

2.4 Modern Development in Q&A Systems

Latest developments of question and answering systems are based on pattern matching approach. Simplicity of this approach suitable for small and medium size websites, which unable to offer complex solutions. This approach has 2 types of namely surface pattern based and template pattern based. Mostly this is semi-automatic answer retrieval. These approach suitable for acronym, factoid, definition question handling and less semantic understanding uses [9]. Knowledge handling of these approach mainly uses automate data handling such as dynamic ontologies.

Early development of dynamic ontology generates by the combination of automatic and semi-automatic methods which was not useful with the expansion of dynamic data. Building cooperative learning based dynamic ontology modeling, using set of corporate and interact to evaluate agents has introduced. In this system, candidate extraction using NLP based dynamo module and concept identifying using WordNet1.2 has used. But this approach should evaluate on more realistic corpus sizes and dynamic corpuses [7].

Real-time propagation from the changes in the data source structure is a feature of dynamic ontology. A dynamic ontology consists of a systematic propagation method, triggered by changes in the data source structure. Most of the time, propagation uses a delta script that contains the difference between the previous and the current data structure [11].

Research to develop a data-driven dynamic ontology model is motivated by the lack of a global standard and a common understanding of the community's knowledge repository using a novel delta script as a crucial tool in the propagation process. The propagation model is lightweight most of the time. This concept should apply to multiple sources of knowledge [12].

2.5 Challenges in Q&A Systems

Literature Review has identified major achievements and issues by considering most cited research in the literature. Findings and limitations have summarized in table 2.1.

Table 2: 1 Summary of benefits and challenges

RESEARCH	BENIFITS	ISSUES IDENTIFIED
Linguistic Approach [9]	Deep Semantic understanding Most reliable as answers are extracted from self- maintained knowledge base.	Quite complex as new rules must be introduced in the knowledge base for every new concept. Domain specific manually developed test collections.
Statistical Approach[9]	Complex non-factoid along with factoids Shallow Semantic understanding Reliable as use of supervised approach	Most suitable for handling large data once properly trained.
Pattern Approach[9]	Factoids, definition, acronym, birth date.	Less semantic understanding Rely on knowledge resource validity. Less scalability as new patterns should be learned with each new concept. Most suitable for small and medium websites, Semantic web.
Linguistic Knowledge Base deep learning[1]	Effectively deals with non-sequential properties in human language	Volatile expert knowledge. Expensive and Difficult to expand and build, while maintaining the richness of its annotations.

Linguistic knowledge based on Hownet knowledgebase natural language processing [17] [18]	Can provide high speed and accuracy on searching answers	Gap towards the actual QA systems Do not provide solution in open field circumstance Need to improve information extraction technique of complicated entity as well with the dynamic data
Expert Knowledge Base [5] [6]	Can build the knowledgebase from experts' knowledge, using interviews and questionnaires. It could also be data driven.	Difficulty to capture experts' knowledge. Brittleness of rules. Difficulty maintains large rule-base. Inference efficiency problem.
Expert knowledge base semi-automated	information acquires from different knowledge sources automatically	Need of human for aligning imported knowledge sources and correcting mistakes
Ontology [7][8]	Data-driven by extracting data from database schemas, dictionaries and web documents	Difficulty to capture expert knowledge. Lack of generalized and sufficiently validated development methodology
Dynamic Ontology [13]	Not depend on expert knowledge and update with dynamic data	

Dynamic Ontology with Vector Space Model	Ability to describe news, reason about news, and deal with the whole process of news analysis. duplicate news removing	Vector Space Model (VSM) used for representing long text-based documents
Dynamic Ontology with Relatedness measurement	Good performance	further analysis could be done to experimental results such as factors influencing each proportion of the searched news with different relatedness degree
Dynamic ontology with semantic search	Obtaining more fresh results	Can use for news like short text based

Based on various categories compared, the dynamic ontology-based knowledge base structure contains a collection of the structure of other three categories. It has a linguistic knowledge base, a logical model, an object attribute relation which similar as relations of concepts in an ontology, values, and attribute. In dynamic ontology-based systems Knowledge acquisition is fully automated, which differ from linguistic knowledge base, expert knowledge base, and basic ontologies. Because of this reason, the knowledge base using dynamic ontology is more suitable when dealing with rapidly expanding data which leads to information overload and dynamic data. Fully automated techniques for Knowledge acquisition

Fully automated techniques for Knowledge acquisition, Solutions for embedding Multimodal Data in Knowledge Base are rare, Handling noise in collected data is challenging, there should be a solution for an inefficient inferencing mechanism.

2.6 Problem Definition

How to develop efficient question and answering system for dynamic pandemic situation.

2.7 Summary

In this chapter I have given a critical review of research by highlighting applications and the issues in the knowledge modeling. I have also identified various technologies used for knowledge modeling using. More importantly I have defined my research problem as in chapter 3 I discuss about the technology adopted in the process.

CHAPTER 3: TECHNOLOGY ADOPTED

3.1 Introduction

In chapter 2 comprehensive critical review of literature in question and answering systems has presented and define the research problem as need of efficient question and answering system for a dynamic pandemic situation. To solve this problem, we have recognized the need for research in to expand with the multi model dynamic data. This chapter is structured under the heading technology adopted and consists of detailed explanation about the theoretical foundation about the technologies used in the proposed solution. Which basically categorize in to two sections Dynamic Ontology and Question and Answering systems.

3.2 Question and Answering Systems

There are several kinds of Question and Answering Systems are available for pandemic situations. Those systems have their own knowledge modeling methods and answer retrieval technologies [9]. According to the selected scope in pandemic situations, developers tend to select different techniques in order to build question and answering systems. Even though most of the question and answering systems used almost same functionalities, there are differences with the knowledge acquisition and answer retrieval methods of those systems. When consider about the pandemic situation always the data are rapidly expanding. When it comes to a pandemic situation like covid-19 the variants, guidelines, vaccines, treatments, rules always changed. This leads to information overload, making it difficult for handle knowledge acquisition in a question answering systems. Because of that, searching relevant information with rapidly expanding data is quite a challenging task.

In a process of question answering system, broadly there are three processes such as, question analysis/preprocess, knowledge analysis and answer analysis/processing [14].

3.2.1 Preprocess

Consists of question classification and query development. Question classification can mainly influence on the performance of a question and answering system. Query development can be done by using different logics. Query development process is required since a question can have information which is not use for answer retrieval. Because of that query should output only useful information for answer retrieval. This query can use search over document for find the relevant answer.

3.2.2 knowledge analysis

Documents/data in search domain should also have processed. Successful search can be done using that. In many Questions Answering Systems tokenization, lemmatization, stemming indexing NLP techniques has used to organize data. With that productive fast search can be done.

3.2.3 Answer processing

This is the last stage and answer can be retrieved as a set of answers or a passage, which received from document preprocess module. In this system query processing and formulation has been use with SPARQL Protocol and RDF Query Language (SPARQL). SPARQL is the standard language for querying graph data represented as RDF triples [15]. The SPARQL provides meaningful query language to retrieving information from knowledge bases employing those formalisms. With light-weight logics SPARQL help for efficient reasoning as well as query answering provide new possibilities for using ontologies in data access [16].

3.3 Knowledge acquisition in question answering system

Covid 19 is a new pandemic and new corpus with novel terms has introduced, because of that knowledge acquisition is a difficult process to handle. Less or no expertise knowledge for a pandemic like covid 19. Many reviews, news articles, web sites have been and continue to publish every day for a pandemic situation like COVID-19. As well as there is a concern with handling rapidly changing multi model data in a

question answering system. Dynamic ontology systems can be used to solve this issue. There are number of dynamic ontology systems are available with various features. Those systems have their own base ontology and data propagation methods. According to the requirement and the solution, developers has selected most suitable techniques for develop dynamic ontologies. Ontologies can be categorized into reference or application formats. Reference ontologies are domain level ontologies as well it describes group of related concepts. Application ontologies are mostly specific and use when modeling across multiple domains. When new disease emergence corresponding ontologies should be developed. To overcome these aims data-driven dynamic ontology is the suitable approach. Dynamic ontology model has two sub processes such as base ontology creation and knowledge propagation. In the base knowledge creation phase, community knowledge should capture from data into an ontology representation, rather than just transforming a specific data format into an ontology as found in existing studies. In knowledge propagation phase, the dynamic knowledge sources become the trigger of propagation [13].

3.4 Summary

In this chapter I have given a review of technology adopted in the proposed system. In chapter 4 I discuss about the approach to the proposed system.

CHAPTER 4: APPROACH

4.1 Introduction

In chapter 3 we have presented the technology adopted in the research. This chapter present approach of the research on development in ontology-based Q&A systems. This chapter has structured under several headings, namely, hypothesis, input, output, process, features, and potential users for the system.

4.2 Hypothesis

Availability of a dynamic ontology-based Q&A system will improve the easiness of use, reliability, and flexibility for public to keep up with the latest data on the pandemic situations. Inspiration behind this hypothesis is coming from the summery of issues identified in Q&A system.

4.3 Input

Multi model data sources such as statistical data, news, articles, tweets, research outcomes, related to pandemic will be the *Input* for the system. Data has been collected through

- https://www.bbc.com/news/coronavirus, https://www.cdc.gov/media/archives.html
- https://www.news-medical.net/condition/Coronavirus-Disease-COVID-19
- https://interestingengineering.com/s/search?q=covid&sort=new
- https://www.nejm.org/coronavirus
- https://twitter.com/hashtag/COVID19?src=hashtag_click
- https://twitter.com/search?q=%23covid,
- https://twitter.com/search?q=%23CORONAVIRUS, https://twitter.com/who

4.4 Output

Output of the system will be answer, related to the question ask by the user. Answers consists with corresponding descriptions, navigate url for more details and the domain which the answer retrieval from.

Table 4: 1 Answer formats

QUESTION	ANSWER
what is the efficiency of Pfizer?	The details related to Pfizer vaccine efficiency are going to be displayed with domain, details, url for the news item ✓ factcheck, Pfizer documents do not reveal dangers of Covid 19_vaccine, https://factcheck.afp.com/doc.afp.com.329Y6J4 ✓ pfizer, Pfizer BioNTech COVID 19 Vaccine Demonstrates Strong Immune Response, High Efficacy and Favorable Safety in Children 6 Months to Under 5 Years of Age Following Third Dose, https://www.pfizer.com/news/press-release/press-release-detail/pfizer-biontech-covid-19-vaccine-demonstrates-strong-immune
What are the latest news?	The latest news (today news) related to covid 19 will displayed with domain, details, url for the news item
Omicron latest news?	The latest news (today news) related to omicron will displayed with domain, details, url for the news item

4.5 Process

Facilitate the dynamic data sources as inputs and produce the required answer for the given question can be consider as the process of the system. In order to handle this, there are two major modules, such as dynamic ontology module and Q&A module has been used.

There are two major processes to be handle in dynamic ontology module, such as base ontology development and dynamic ontology propagation. To do that below sub processes has used in the system. Base ontology creation consists with nine sub processes namely, define the purpose, competency questions derivation, term extraction, analysis, knowledge synthesis, reuse and standardization, design the representational model, ontology development and evaluation. Dynamic ontology propagation has 4 sub modules such as, get real time changes of data, differencing, populate the ontology. populate the ontology has three phases such as candidate instances identification, classifier construction and instances classification. Candidate instances Identification has 3 sub processes as summarize, morpho-lexical analysis, named entity recognition and construction of a classifier has 3 sub tasks classes properties and relationships selection, triggers selection and rules generation and classification of instances consists of 2 sub tasks as association of instances and Instantiation.

Q&A module has two types of answer retrieval patterns such as direct retrieval of answers by analyzing the content of individuals and convert questions into normal form and by mapping questions into RDF Tripels (subject-predicate-object) and answer retrieval by query processing. Generate RDF triples has 3 sub processes such as preprocessing, global transformation, local transformation.

The designing of the system is discussed in the design chapter in a detailed manner.

4.6 Features

The overall features of the system include the following.

- Handle multi model dynamic data in a pandemic situation
- Handle the automated data retrieval from dynamic data.

- Easiness of use
- Reliability
- Low data cost
- Flexibility

4.7 Users

This system is focus on the users who are engage with searching covid 19 related data such as public, researchers etc.

4.8 Summary

This chapter summarize about the hypothesis, input, output, process, features, users of the system. In chapter 5 I discuss about the design of the proposed system.

CHAPTER 5: DESIGN

5.1 Introduction

In chapter 4 we have presented, the approach to dynamic ontology-based Q&A system for pandemic situation. This chapter present the top-level architecture of dynamic ontology-based Q&A system for pandemic situation, comprising 2 modules namely, dynamic ontology module and question answering module. The top-level architecture for design of the system shows in figure 5.1.

5.2 Top Level Design

System was designed with main two major modules.

1. Dynamic Ontology Module
2. Question Answering Module

5.2.1 Dynamic Ontology Module

In emergence of a new disease, knowledge is changed frequently. However, in cases of fast evolving and novel pandemic, domain experts' domain knowledge is not available. There for original knowledge bases need to be developed in the emergence of a new disease as well as there is a method for update the changed knowledge frequently. To achieve the aim, the dynamic ontology model is the most suitable approach. This module is for the construction of dynamic ontology. There are two sub-modules namely, base ontology creation and dynamic ontology creation and propagation.

Following sections will discuss design details about the sub processes.

- base ontology development
- dynamic ontology development and propagation

5.2.1.1 Base ontology development

Base ontology development is the first step in the dynamic ontology concept. There are several steps to achieve this. In based ontology creation first step was to identify the purpose of an ontology as handle the dynamic data related to pandemic, with the purpose derivation of competency questions using existing news, articles, tweets etc (ex: What is the success rate of pfizer vaccine, what is the outcome of vaccine) has done. Using COVID-19 datasets and by calculating the word frequencies and considering the words with higher frequency using of covid 19 related news, articles and tweets term extraction has done (ex: patient, vaccine, efficacy, Pfizer, age limit). Then analysis **(**analysis the covid 19 vaccines based on their type. ex: Moderna and Pfizer are mRNA vaccines) and knowledge synthesis by arranging the knowledge defining the relationships (ex: Moderna vaccine is a mRNA vaccine). Reuse and standardization using CIDO: Ontology of Coronavirus Infectious Disease has been reused in this system. CIDO is open source, community driven, biomedical ontology in coronavirus infectious disease, which has developed in order to representation of various coronavirus infectious diseases, provide standardized human and computer interpretable annotation and including their transmission, etiology, diagnosis, pathogenesis, prevention and treatment. Design of representational model was the next step and using that, model domain knowledge using classes, properties, and their relationships. Finally, development of the ontology has done by structuring and modelling the domain knowledge produced in the previous steps.

5.2.1.2 Dynamic ontology development and propagation

This section elaborates the process of dynamic ontology development and propagation. Real time changes in data sources identified by web scraping.

Propagation

Retrieving the real time changes of the news, twitters, raw data from the Web using web scrapping technique is the main idea of this. Web scrapping used to extract the

website data automatically with the use of html tags. Automation part of the web scrapping was handled by task schedulers which has scheduled with 12 hours periods in each day. Realtime changes detected by data sources such as news sites and publications (Ex: https://www.bbc.com/news/coronavirus, https://www.cdc.gov/media/archives.html, https://twitter.com/hashtag/COVID19?src=hashtag_click, https://www.news-medical.net/condition/Coronavirus-Disease-COVID-19 etc.). By inspecting the structure of html in the given site list by deciphering the data encoded in urls, for news sites and publications extract the relevant data from web using requests and beautiful soup object. The soup object can be used to extract data such as headings, paragraphs from the website which is scraping. The data which dynamically loaded to the page such as twitters have scrapped with the use of selenium, web drivers and beautiful soup module. After the web scrapping useful data such as news/article/tweet header, link to the item and date have been saved to .csv file with a timestamp.

Next step was differencing the recently saved .csv file and the previous version of .csv file (which saved to the system before 12 hours), the changes identified in data as the basic operations of added, removed, changed. The added and changed data saved as a .csv file for the use of ontology population.

Population

Ontology Population is related to identify non taxonomic relationships instances and ontology class property instances. In this system domain specific process used for automatic population of ontology. NLP and information extraction (IE techniques has used to identify and classify ontology instances. Ontology population process consists of three processes, such as candidate instances identification, classifier construction and instances classification.

Candidate instances Identification mainly focused on non-taxonomic relationships instances identification and ontology class properties identification from inputted document annotating. Full text in the document/web page summarized and then identified the grammatical category of every term in summarized sentences, after that identified names which refer places, persons and organizations etc.

By using base ontology and the output a classifier is the main purpose of construction of a classifier. Class properties selection and relationships selection, triggers selection and rules generation are the main three tasks in this process. Main step of this process is generation of rules with if <condition> then <conclusion> form.

Classification of instances outputs a populated ontology using classifier and annotated summarized data generated from above steps. Association of instances and instantiation are the main tasks of this process. In association of instances process, instances have linked to their corresponded class properties and to non-taxonomic relationships, while instances process effectively populate the ontology with data populated in association of instance task.

5.2.2 Question Answering Module

This module mainly focusses on WH questions. There are two types of answer retrieval processes such as direct retrieval of answers by analyzing the content of ontology individuals and answer retrieval by RDF Tripels (subject-predicate-object) formulation from the question.

In direct retrieval of answers using ontology individuals the questions such as what is the latest situation? retrieve all the latest data for today and the question such as what is omicron latest? retrieve all the latest data which related to omicron.

In answer retrieval by RDF Tripels formulation from the question first create the dependency tree using Stanford dependency parser. Then simplification of the dependency tree to normal form using tree preprocessing, global transformation and local transformation.

includes 2 processes namely, question preprocess and query formulation and answer retrieval. In question preprocess, convert entered question for normal structure, which based on the RDF format has been done. There are 4 sub processes in question preprocess namely, Normal form extraction using dependency tree simplification by preprocessing, global transformation, and local transformation.

5.3 System Architecture

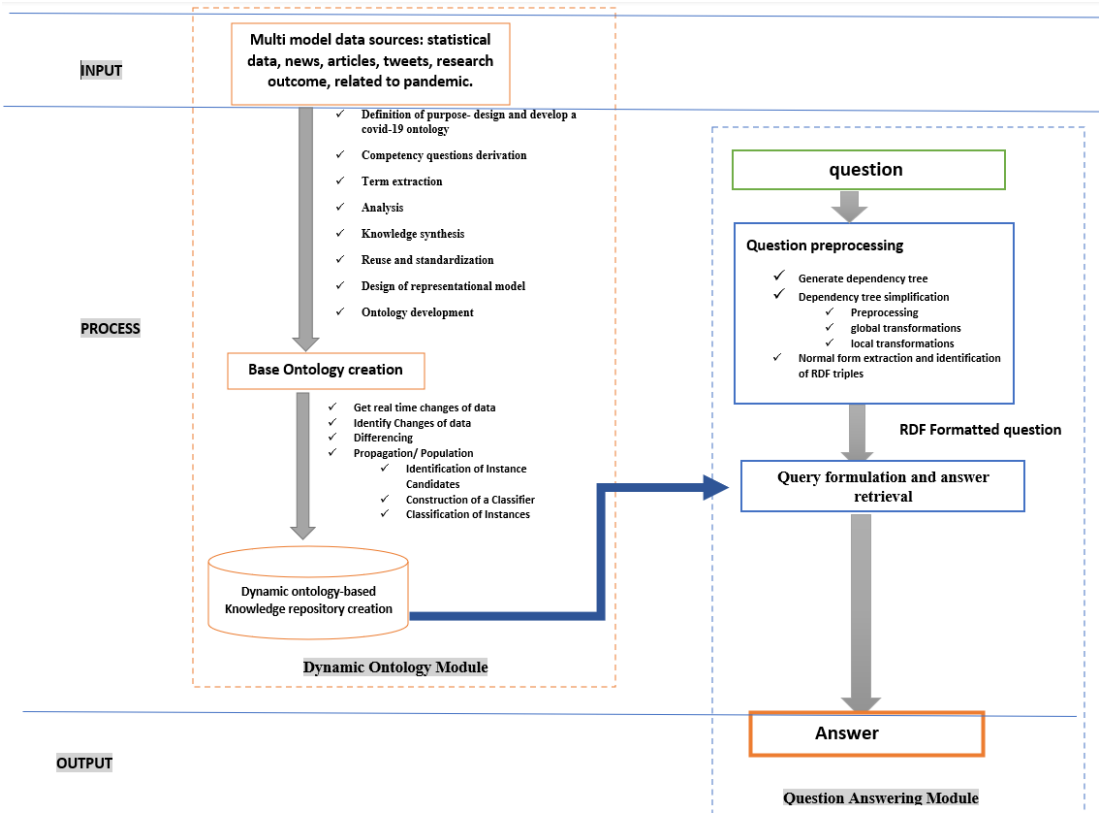


Figure 5.1:System Architecture

5.4 Summary

This chapter summarize the top-level design and model architecture of the system. Starting from the top-level architecture and the configuration of the system; Followed by the list of major modules in the design and in-depth design details of each module. In chapter 6 I discuss about the implementation of the proposed system.

CHAPTER 6: IMPLEMENTATION

6.1 Introduction

In chapter 5 we have presented, the design to dynamic ontology-based Q&A system for pandemic situation. This section will discuss about the implementation of the dynamic ontology-based Q&A system. This chapter has structured under several headings, namely, Dynamic Ontology Implementation, Q&A module implementation. Testing evaluation will be discussed in the separate section.

6.2 Dynamic Ontology Implementation

6.2.1 Base ontology implementation

Base ontology implementation is based on several steps. First step was to identify the purpose of an ontology as, handle the dynamic data related to pandemic. Derivation of competency questions by analyzing existing news, articles, tweets etc (ex: What is the success rate of pfizer vaccine, what is the outcome of vaccine) has done.

Using COVID-19 datasets and by calculating the word frequencies and considering the words with higher frequency using of covid 19 related news, articles and tweets term extraction has done (ex: patient, vaccine, efficacy, Pfizer, age limit). In this step calculating the word frequencies has done by using existing news, articles. Using the urls first generate a python BeautifulSoup object and then search for all paragraphs in the html document by considering paragraph tags in the object. By iterating over paragraphs, removing square brackets, extra spaces, special characters and digits, generated a formatted article text, then by use of nltk sentence tokenize, tokenized the formatted article text, after that by removing English stop words, word frequencies have been calculated. Largest frequency words considered as terms of the covid domain.

Then analyzed (analysis the covid 19 vaccines based on their type. ex: Moderna and Pfizer are mRNA vaccines) and knowledge synthesis by arranging the knowledge defining the relationships (ex: Moderna vaccine is a mRNA vaccine).

Reuse and standardization using CIDO: Ontology of Coronavirus Infectious Disease has been reused in this system. CIDO is open source, community driven, biomedical ontology in coronavirus infectious disease, which has developed in order to representation of various corona virus transmissible diseases, transmission, etiology, diagnosis, treatment, prevention, and pathogenesis. Design of representational model was the next step and using that, model domain knowledge using classes, properties, and their relationships. Finally, development of the ontology has done by structuring and modelling the domain knowledge produced in the previous steps. Base ontology consists with 34 classes, 10 object properties related to COVID-19 vaccine types, vaccines series, age limits, severe diseases, side effects, variants.

6.2.2 Implementation of dynamic behavior

6.2.2.1 Propagation

Retrieving the real time changes of the news, twitters, raw data from the Web using web scrapping technique has implemented in this step. Web scrapping used to extract the website data automatically with the use of html tags. Automation part of the web scrapping was handled by task schedulers which has scheduled with 12 hours periods in each day. Created two tasks, with windows task scheduler using the python script. In this step task created by adding 2 new tasks with an action by python script and setting the trigger daily at 6AM and 6PM.

Realtime changes detected by data sources such as news sites and publications (Ex: https://www.bbc.com/news/coronavirus,https://www.cdc.gov/media/archives.html, https://twitter.com/hashtag/COVID19?src=hashtag_click,https://www.news-medical.net/condition/Coronavirus-Disease-COVID-19 etc.). By inspecting the structure of html in the given site list by deciphering the data encoded in urls, for news sites and publications extract the relevant data from web using requests and beautiful soup object. Request is python HTTP library use for http request send and beautiful soup is a HTML parser use to pass the Document Object Module (DOM) and extract the data which use in the system. The soup object has used to extract data such as

headings, paragraphs from the website which is scraping. But in twitter sites the data loaded to the page dynamically. Dynamic web pages take more time to load than a static web page and in dynamic web pages information is changed frequently. In order to scrape dynamic twitter sites Selenium test automation framework has used with chrome driver and beautiful soup object has created with returned driver page source. After the web scrapping useful data such as news/article/tweet header, link to the item and date have been saved to .csv file with a timestamp.

Next step was differencing the recently saved .csv file and the previous version of .csv file (which saved to the system before 12 hours), in python csv-diff, the changes of data identified as added, removed, changed. The added and changed data saved as a .csv file for the use of ontology population. csv-diff use to get human readable summary from differences between files. This is not same as a standard differencing tool which compares the records line by line and order of records consideration. CSV-Diff recognize common lines using key field(s) and then compare the fields contents in each line. The **key** option means, column should be treated as the unique key, in order to identify which records have changed. The csv-diff process maintains a excellent level of control for what to diff, as well it can voluntary ignore certain changes types (ex: adds, deletes, position changes etc). For parent-child formatted data csv-diff is suites. Small changes of the tree organization in upper level of tree can cause to a big impact in the position of descendant records. With the use of matching records using key, CSV-Diff has avoided this issue, when detect changes in sibling order.

6.2.2.2 Ontology Population

Ontology population has done by Identification of non-taxonomic relationship instances and ontology class properties. In this system domain specific process used for ontology population. Mainly NLP and information extraction (IE) techniques has used to identify and classification of ontology instances. Ontology population consists with three processes namely candidate instances identification, classifier construction, instances classification.

Candidate instances identification

Purpose of this process is, by annotating the input document, identify non taxonomic relationship instances and ontology classes properties. Mainly NLP techniques has used in this process. There are 3 sub processes such as summarize the full text in the document/web page, morpho lexical analysis and named entity recognition.

In this summarize process, the **added** and **changed** changes in data identified in the differencing task take into consideration. By iterating the changes rows in .csv file and using the row item in story (link to the news/document/tweeter item) column, summarize the full text in the document/web page using word frequency and sentence frequency calculations and derivation. Using the urls first generate a python BeautifulSoup object and then search for all paragraphs in the html document by considering paragraph tags in the object. By iterating over paragraphs, removing square brackets, extra spaces, special characters and digits, generated a formatted article text, then by use of nltk sentence tokenize, tokenized the formatted article text, after that by removing English stop words, word frequencies have been calculated. Then sentences which length is less than 30 take into consideration and calculate the sentence frequencies using above calculated word frequencies. Finally sentence with the largest sentence frequency selected and use for the instance candidate identification. Then identified the grammatical category of every term in a sentence by morpho lexical analyzing. Then by named entity recognition, identified persons, organizations and places etc. Finally output the annotated summary.

Classifier construction

Classifier construction process have 3 sub processes such as triggers selection, classes properties and relationships selection, and rules generation. In this process as inputs provide corpus and ontology finally output the classifier.

Using the base ontology and corpus the classes, properties and their relationships identified by comparing the semantic similarity. And using the ontology classes and properties selection of triggers/synonyms has done by tokenizing and using synonyms and hypernym. Rules generation have the form of **if <condition> then <conclusion>.** Condition consists with five predicated which join by logical conjunctions.

- Noun_phrase(b) - **b** noun phrase (ex: Noun_phrase(Pfizer))
- Instance (I, b) - **I** instance which going to be classified noun phrase **b (ex:** Instance (Pfizer_news_instance**,** Pfizer))
- Trigger (b, c) - **c** trigger of a noun phrase **b** (Ex: Trigger (Pfizer, Vaccine))
- Relationship synonym (b, R) - **b** synonym of **R** non-taxonomic relationship (Ex: Relationship synonym (efficacy, has_efficient)
- Non taxonomic relationship (R, d) -**R** non-taxononlmic relationship with **b** class (Ex: Non taxonomic relationship (has_efficient, Pfizer)

Conclusions consist with two predicates which joins by logical conjunction.

- Is_a(I, d)- I instance should classified as instance of d class (ex: is_a(Pfizer_news_instance**,** Pfizer)
- Non taxonomic relationship association (I" 12, R, b) - I instances associated by a R non-taxonomic relationship of b class (ex: Non taxonomic relationship association (Pfizer_news_instance, symptomatic, has_efficient, vaccine))

Noun_phrase(Pfizer)

Instance (Pfizer_news_instance**,** Pfizer))

Trigger (Pfizer, Vaccine)

Relationship synonym (efficacy, has_efficient)

Non taxonomic relationship (has_efficient, Pfizer)

Is_a(Pfizer_news_instance, Vaccine)

Non_taxonomic_relationship_ association (Pfizer_news_instance, symptomatic, has_efficient, vaccine)

Figure 6: 1 Classification rule example for non-taxonomic 'has_efficient' relationship.

Instance classification

Instance classification process consist with two sub tasks namely association of instances and instantiation. In this process annotated content, ontology and the classifier provide as input and output is the populated ontology.

With instance association, instances links to their corresponded classes, properties or non-taxonomic relationships and in instantiation populate the ontology efficiently with the output of instance association process.

Annotated sentences noun phrases can be followed or proceed by a trigger/verb. When that trigger matches the trigger predicate of a classification rule type Trigger (b, c) (c trigger of a noun phrase b), instances are classified according to the Is_a (I, d) (I instance should classified as instance of d class) predicates of its conclusion. Related non-taxonomic relationship of this class(b) is also instantiated as, non-taxonomic relationship association (I" 12, R, b) (I instances associated by a R non-taxonomic relationship of b class) predicates of its conclusion. As well whenever a trigger matches the trigger predicate of classification rule type Relationship synonym (b, R) (b synonym of R non-taxonomic relationship), instance is classified according to the is_a(l, b) predicate of its conclusion.

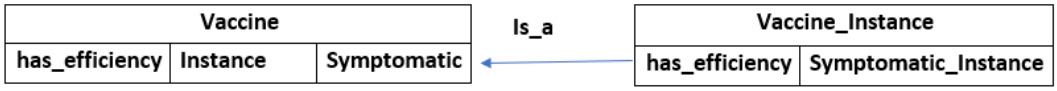


Figure 6: 2An example of a populated class

6.3 Q&A module implementation

There are two types of answer retrieval, such as direct retrieval of answers by query processing after analyzing the content of individuals and by query processing with mapped questions into RDF triples.

6.3.1 Direct answers retrieval by query processing

In this method answer retrieval is done by analyzing the ontology individuals with the use of owlready2. Owlready2 is a python package for ontology related programming. Question which contains word “latest” will retrieve all the news for today, and specific latest news also can be retrieved (ex: latest news for omicron).

6.3.2 Answers retrieval by query processing for RDF triples mapped questions

In this process, dependency tree created with Stanford dependency parser for the question. Then simplification is done using tree preprocessing, global transformation and local transformation for dependency tree in order to create the normal form. Finally, the RDF triples retrieved from the normal form and by query processing answers has been retrieved.

Convert the user entered question to RDF format based normal form was the main purpose. Stanford dependency tree represent the grammatical relationships of the words in a sentence.

Leaves of the dependency trees can be a value of

- Resources - represent a person/location/date etc.
- List – resources with ordered collection (without duplicates)
- Missing (denote by?) – unknown value which need to find

Internal nodes of the dependency tree can be an operator of

- Triple (subject, predicate, object), each triple has one missing(?)
- Union/ interaction ($\cup$/$\cap$): which input is a 2 list and output the intersection or union
- Sort (*sort(l,a)*): which sort the list(l) in increasing order with predicate(a)
- First and last: take a list of elements and output the first/last element of the list

Normal form creation consists of sub tasks such as, simplification of the dependency tree and normal form construction. Simplification of the dependency tree process contains three sub processes namely preprocessing, global transformation, and local transformation.

6.3.2.1 Simplification of the dependency tree

Preprocessing

First performed multiword expressions recognition for the purpose of merge every node which belongs to the same expression. Mainly merged the neighbor nodes which have same name entity. Next identification of question words and removed has been done using list of thirty question words and checking the two first words of question. Lastly applied lemmatization for nouns and nounification added on verbs.

Global transformation

Mainly two types of transformations which modify the tree, such as transformation of *amod* dependency and transformation of conjunction dependencies (ex: *conj_or, conj_and* etc). These transformations balance the tree by adding a new node to the tree.

Local transformation

All remaining edges locally analyzed. Based on their dependency tags merge, remove, replace rules applied for each of them. Merge applied for remaining *amod* dependencies or *nn* with two end points at the edge of the tree. Remove rule applied for all sub trees at the edge of the tree, and mainly applied for det dependencies. (Ex: vaccine det→ the becomes vaccine). Replace dependency tag by a triple production tag. Eight types of triple production tags have been introduced namely R0, R1, R2, R3, R4, R5, R_{spl}, R_{conj}.Based on the question word in the question information added to relevant nosed (ex: if the question word is *why*, word *reason/cause* added in root child). This transformation supported currently for thirty question words.

6.3.2.2 Normal form construction

This is a recursive function which normalize the tree into normal form. Trees display as T and node as N. Firstly if node(N) is a leaf normalize of the node is a value node.

After that rule R0, R1, R2, R3, R4, R5 use to generate different types of triples. Table shows Normalize node with R_i of child(T), when T is only child of N and $R_i \in$ {R0, R1, R2, ..., R5}.

Table 6: 1 Normalization rules for R0,R1,R2,R3,R4, R5

Rule R	$\texttt{Normalize}(N \xrightarrow{R} T)$
R_0	$\texttt{Normalize}(T)$
R_1	$\underline{T}$
R_2	$\texttt{Triple}(\underline{T}, \underline{N}, ?)$ if T is a leaf
	$\texttt{Normalize}(T)$ otherwise
R_3	$\texttt{Triple}(?, \underline{N}, \texttt{Normalize}(T))$
R_4	$\texttt{Triple}(?, \texttt{Normalize}(T), \underline{N})$
R_5	$\texttt{Triple}(\texttt{Normalize}(T), \underline{N}, ?)$

If root N has more child, all are linked by a rule in {R0, R1, R2, ..., R5}. Rule R_{spl} and R_{conj} are result of global transformation. Rule R_{spl} occurs if root node $_{has}$ ordinal one child or superlative. Based on the ordinal or superlative normalize the output for relevant nodes. After applying all these steps normal form generated.

6.3.2.3 Query formulation and answer retrieval

Finally retrieve the RDF format from normal form using json formatters in python. Then SPARQL query formulation based on RDF triples using rdflib and retrieve answer was done.

6.4 Summary

This chapter summarize about the implementation process of the system. In addition to that several diagrams were discussed in order to understand the workflows for different scenarios. In chapter 7 I discuss about the evaluation of the proposed system.

CHAPTER 7: EVALUATION

7.1 Introduction

This chapter contains the evaluation process which used to evaluate (and test) the Q&A system which is discuss during this documentation. Since the main outcome of the research is a software framework for the dynamic ontology-based Q&A system; unique approach was used to evaluate the system. Evaluation process consists of two major sub-processes.

One will be evaluation of the dynamic ontology module and Second process is to evaluate the question-and-answer module. In both evaluation processes time evaluation and precision has done.

7.2 Evaluation of dynamic ontology module

In evaluation of the dynamic ontology module, using 50 rows which has collected in propagation has used and calculated the average time each row takes to populate the ontology. Time consumed for 50 rows was, 653 seconds and average time taken per row there for 13.06 seconds.

Precision calculation means the ratio between the true positives and all the occurrences that classified as positive. When consider about the 50 rows which has collected in propagation number of ontology individuals should generate is 50, but the number of ontology individuals generated was 35, there for precision for 50 rows calculated as 70%.

7.3 Evaluation of question-and-answer module

In evaluation of question-and-answer module calculate the average time that takes to discover triples and retrieve the answer and precision calculated for RDF triple-based answer retrieval. First by considering same question with different patterns (20 patterns) got time and the precisions for generated RDF triples and retrieved answer.

Ex:

- what is the success rate of pfizer vaccine?
- what is the pfizer vaccine success rate?
- pfizer vaccine success rate is?
- how about the success rate of pfizer vaccine?
- pfizer vaccine success rate is what?
- show me the pfizer vaccine success rate?
- show me the success rate of pfizer vaccine?
- give me the success rate of pfizer vaccine?
- give me the pfizer vaccine success rate?
- what is the success rate of pfizer? Etc.

In this process time consumed for 20 questions was 204.74 seconds and there for average time taken per question is 10.237 seconds, and number of RDF triples should generate (Tp + Fp) is 20 and number of RDF triples generated that are correct (Tp) was 14 and the Precision for RDF triples generated is 70%. And number of answers should retrieve is 20 and Number of RDF triples generated that retrieve the answer was 12, there for precision for answer retrieval is 60%.

Then considered different questions related to covid 19 vaccines and got average time that takes to discover triples and retrieve the answer and the precisions for generated RDF triples and retrieved answer. Time consumed for 50 questions was 508.5 seconds, there for average time taken per question is 10.17 seconds and number of RDF triples should generate was 50 and number of RDF triples generated that are correct was 34 there for precision for RDF triples generated in different question related to vaccine was 68%. And number of answers should retrieve was 50, but number of RDF triples generated that retrieve the answer was 30, precision for answer retrieval is 60%.

7.4 Summary

This chapter summarize about the evaluation of the system. In chapter 8 I discuss about the conclusion and further works of the proposed system.

CHAPTER 8: CONCLUSION AND FURTHER WORK

8.1 Introduction

This chapter explains the small conclusion of the proposed system. Also, then it will explain the future work of the research. As in every research we have many problems while we were doing the research. This chapter explains limitations of the research, future works of the research and the conclusion of the research. Content of this chapter includes to what extend the main objective(s) were archived by the designing and implementation phase and verify using the evaluation process.

8.2 Conclusion

It was hypothesized that the 'Availability of a dynamic ontology-based Q&A system will improve the easiness of use, reliability, and flexibility for public to keep up with the latest data on the pandemic situations'. In order to prove that hypothesis dynamic ontology-based question answering system has developed.

This system proposes an approach for ontology population, using the automatic generation of a classifier, and using automate detection of the multi modeled latest data which continuously updated.

Finally, the evaluation process and results were presented in the chapter 7. According to the evaluation results it is clear that using this solution, the objective was archived. Dynamic ontology-based Q&A system can be used as an efficient Q&A system for dynamic pandemic situation.

8.3 Limitations and Further Work

Regarding the limitations, one of the major limitations is ontology structure population, current system populates only latest data to the system. But the ontology structure can be changed with evolving data, terms in a new domain such as pandemic.

Few things were identified as potential further works, to make the system more useful and powerful.

One will be adapting the system with ontology structure population, in order to increase the reliability of the system with the evolving data.

Another will be focusses on improve the different parts of the Question-and-answer algorithm (multiword expressions recognition, better analysis of grammatical dependencies), in order to easiness of use the system.

8.4 Summary

This chapter summarize about the conclusion, limitations and future works of the developed dynamic ontology-based question and answering system. The limitations were identified, and potential solutions also discussed. Few addition works were identified as further works in order to make the system more useful and powerful

REFERENCES

[1] A. T. Bimba *et al.*, "Towards knowledge modeling and manipulation technologies: A survey," *Int. J. Inf. Manag.*, vol. 36, no. 6, pp. 857–871, Dec. 2016, doi: 10.1016/j.ijinfomgt.2016.05.022.

[2] H. Rahman and Md. I. Hussain, "A light-weight dynamic ontology for Internet of Things using machine learning technique," *ICT Express*, vol. 7, no. 3, pp. 355–360, Sep. 2021, doi: 10.1016/j.icte.2020.12.002.

[3] D. Ai, H. Zuo, and G. Liu, "Dynamic ontology-based user modeling in personalized information retrieval system," p. 5, 2010.

[4] J. M. Alonso, L. Magdalena, and S. Guillaume, "A Simplification Process of Linguistic Knowledge Bases," p. 7, 2005.

[5] S. Duer, "Expert Knowledge Base to Support Maintenance of a Radar System," *Def. Sci. J.*, vol. 60, no. 5, pp. 531–540, Jul. 2010, doi: 10.14429/dsj.60.84.

[6] Minkoo Kim, Fenghua Lu, and V. V. Raghavan, "Automatic construction of rule-based trees for conceptual retrieval," in *Proceedings Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000*, A Curuna, Spain, 2000, pp. 153–161. doi: 10.1109/SPIRE.2000.878191.

[7] M. Tenorth and M. Beetz, "KnowRob: A knowledge processing infrastructure for cognition-enabled robots," *Int. J. Robot. Res.*, vol. 32, no. 5, pp. 566–590, Apr. 2013, doi: 10.1177/0278364913481635.

[8] K. Lakel and F. Bendella, "Dynamic Evaluation of Ontologies," *Procedia Comput. Sci.*, vol. 73, pp. 16–23, 2015, doi: 10.1016/j.procs.2015.12.043.

[9] S. K. Dwivedi and V. Singh, "Research and Reviews in Question Answering System," *Procedia Technol.*, vol. 10, pp. 417–424, 2013, doi: 10.1016/j.protcy.2013.12.378.

[10] M. Zviedris, A. Romane, G. Barzdins, and K. Cerans, "Ontology-Based Information System," in *Semantic Technology*, vol. 8388, W. Kim, Y. Ding, and H.-G. Kim, Eds. Cham: Springer International Publishing, 2014, pp. 33–47. doi: 10.1007/978-3-319-06826-8_3.

[11] J. Murdock, C. Buckner, and C. Allen, “TWO METHODS FOR EVALUATING DYNAMIC ONTOLOGIES,” p. 15.

[12] D. H. Fudholi, W. Rahayu, E. Pardede, and Hendrik, “A Data-Driven Approach toward Building Dynamic Ontology,” in *Information and Communicatiaon Technology*, vol. 7804, K. Mustofa, E. J. Neuhold, A. M. Tjoa, E. Weippl, and I. You, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 223–232. doi: 10.1007/978-3-642-36818-9_23.

[13] D. H. Fudholi, W. Rahayu, and E. Pardede, “A data-driven dynamic ontology,” *J. Inf. Sci.*, vol. 41, no. 3, pp. 383–398, Jun. 2015, doi: 10.1177/0165551515576478.

[14] Research Scholar, Department of Computer Science & Engineering, Mewar University, Chittorgarh, Rajasthan, India, V. Mishra*, Dr. N. Khilwani, and Technical Architect, Edifecs RoundGlass, Noida, India., “QUASE: AN Ontology-Based Domain Specific Natural Language Question Answering System,” *Int. J. Recent Technol. Eng. IJRTE*, vol. 8, no. 4, pp. 261–268, Nov. 2019, doi: 10.35940/ijrte.D6773.118419.

[15] A. Rodriguez Diaz, A. Benito-Santos, A. Dorn, Y. Abgaz, E. Wandl-Vogt, and R. Theron, “Intuitive Ontology-Based SPARQL Queries for RDF Data Exploration,” *IEEE Access*, vol. 7, pp. 156272–156286, 2019, doi: 10.1109/ACCESS.2019.2948115.

[16] J. Schoenfisch and H. Stuckenschmidt, “Analyzing real-world SPARQL queries and ontology-based data access in the context of probabilistic data,” *Int. J. Approx. Reason.*, vol. 90, pp. 374–388, Nov. 2017, doi: 10.1016/j.ijar.2017.08.005.

[17] K.-M. Kouamé and H. Mcheick, “An Ontological Approach for Early Detection of Suspected COVID-19 among COPD Patients,” *Appl. Syst. Innov.*, vol. 4, no. 1, p. 21, Mar. 2021, doi: 10.3390/asi4010021.

[18] P. Qian, X. Qiu, and X. Huang, “Analyzing Linguistic Knowledge in Sequential Model of Sentence,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, 2016, pp. 826–835. doi: 10.18653/v1/D16-1079.

[19] Z. Xie, Z. Zeng, G. Zhou, and T. He, “Knowledge Base Question Answering Based on Deep Learning Models,” in *Natural Language Understanding and Intelligent Applications*, vol. 10102, C.-Y. Lin, N. Xue, D. Zhao, X. Huang, and

Y. Feng, Eds. Cham: Springer International Publishing, 2016, pp. 300–311. doi: 10.1007/978-3-319-50496-4_25.

[20] X. Wen, X. Ma, J. Li, J. Z. Pan, and J. Xie, “Toward Ontology Representation and Reasoning for News,” p. 2.

[21] H. Beheshti, F. Poorahangaryan, and S. A. Edalatpanah, “NewsSE: An Ontology-based Search Engine for News,” *Comput. Sci. Inf. Technol.*, vol. 5, no. 2, pp. 37–49, Mar. 2017, doi: 10.13189/csit.2017.050201.

[22] Lin Li, Xia Hu, Chao Xu, and Yi-Ming Zhou, “Relatedness measurement for news items,” in *2008 International Conference on Machine Learning and Cybernetics*, Kunming, China, Jul. 2008, pp. 2580–2584. doi: 10.1109/ICMLC.2008.4620843.

[23] B. Hu, J. Wang, and Y. Zhou, “Ontology Design for Online News Analysis,” in *2009 WRI Global Congress on Intelligent Systems*, Xiamen, China, 2009, pp. 202–206. doi: 10.1109/GCIS.2009.78.

[24] A. Albarghothi, W. Saber, and K. Shaalan, “Automatic Construction of E-Government Services Ontology from Arabic Webpages,” *Procedia Comput. Sci.*, vol. 142, pp. 104–113, 2018, doi: 10.1016/j.procs.2018.10.465.

APPENDICES

Appendices A: Dynamic ontology module

A.1 Introduction

Dynamic ontology module consists with base ontology creation and dynamic ontology propagation. Following sub sections will provide detail about some of the process descriptions.

A.2 Base Ontology

Figure A: 1 Ontology structure

A.3 Dynamic ontology Propagation

Web Scrapping

Retrieving the real time changes of the news, twitters, raw data from the Web using web scrapping technique has implemented in this step.

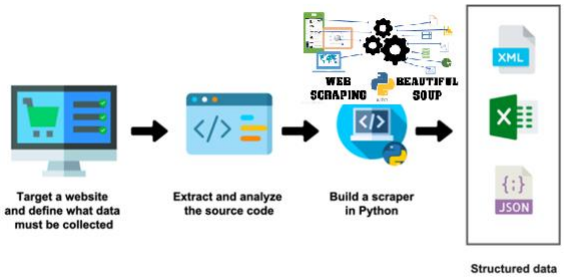


Figure A: 2 Web Scrapping using python beautiful soup

Differencing

First scrapped data of new articles, document and twitter save as .csv file with a time stamp

now = datetime.datetime.now().strftime('%Y-%m-%d %H')

V1 = str(now) + ".csv"

id,Header,Time,Story

1,Three people who used to work in aviation up until the Covid pandemic discuss whether they would return to the industry.,Posted at 23:13 16 Jul,https://www.bbc.com/news/business-61830479

2,NHS Borders says it is doing "everything possible" to ensure routine operations can go ahead.,Posted at 15:40 16 Jul,https://www.bbc.com/news/uk-scotland-south-scotland-62134021

3,Some 200 staff are off sick or self-isolating while bosses are "working hard" to maintain services.,Posted at 14:43 16 Jul,https://www.bbc.com/news/uk-england-cambridgeshire-62132810

Figure A: 3 format of the outputted .csv file at now

lastTwelveHourDateTime = datetime.datetime.now() - datetime.timedelta(hours=12)

V2 = lastTwelveHourDateTime.strftime('%Y-%m-%d %H') + ".csv"

id,Header,Time,Story

1,It is estimated that one in 17 people in Wales have the virus^ according to latest official survey.,Posted at 15:48 17 Jul,https://www.bbc.com/news/uk-wales-62178744

2,Manx Care says "good progress" is being made and dedicated services should be set up in September.,Posted at 15:30 17 Jul,https://www.bbc.com/news/world-europe-isle-of-man-62171615

3,More people than originally planned will be offered the jab in the UK ahead of the coming winter.,Posted at 15:30 17 Jul,https://www.bbc.com/news/health-62183714

Figure A: 4 Format of the outputted .csv file before 12 hours

{'added': [

{'id': '1', 'Header': 'Three people who used to work in aviation up until the Covid pandemic discuss whether they would return to the industry.', 'Time': 'Posted at 23:13 12 Jul', 'Story': 'https://www.bbc.com/news/business-61830479'},

{'id': '2', 'Header': 'NHS Borders says it is doing "everything possible" to ensure routine operations can go ahead.', 'Time': 'Posted at 15:40 12 Jul', 'Story': 'https://www.bbc.com/news/uk-scotland-south-scotland-62134021'},

{'id': '3', 'Header': 'Some 200 staff are off sick or self-isolating while bosses are "working hard" to maintain services.', 'Time': 'Posted at 14:43 12 Jul', 'Story': 'https://www.bbc.com/news/uk-england-cambridgeshire-62132810'}

],

'removed': [

{'id': '1', 'Header': 'It is estimated that one in 17 people in Wales have the virus^ according to latest official survey.', 'Time': 'Posted at 15:48 15 Jul', 'Story': 'https://www.bbc.com/news/uk-wales-62178744'},

{'id': '2', 'Header': 'Manx Care says "good progress" is being made and dedicated services should be set up in September.', 'Time': 'Posted at 15:30 15 Jul', 'Story': 'https://www.bbc.com/news/world-europe-isle-of-man-62171615'},

{'id': '3', 'Header': 'More people than originally planned will be offered the jab in the UK ahead of the coming winter.', 'Time': 'Posted at 15:30 15 Jul', 'Story': 'https://www.bbc.com/news/health-62183714'}

],

'changed': [],

'columns_added': [],

'columns_removed': []}

Figure A: 5 identified changed data

A.4 Dynamic ontology Population

Ontology population process consists of three processes, such as candidate instances identification, classifier construction and instances classification.

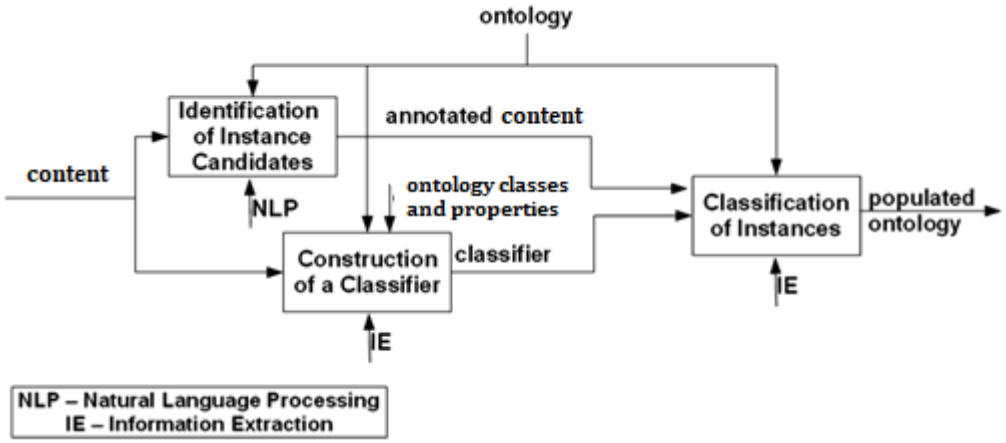


Figure A: 6 Automatic ontology population process

Full text in the document/web page summarized and then identified the grammatical category of every term in summarized sentences, after that identified names which refer places, persons and organizations etc in identification of instance candidates process.

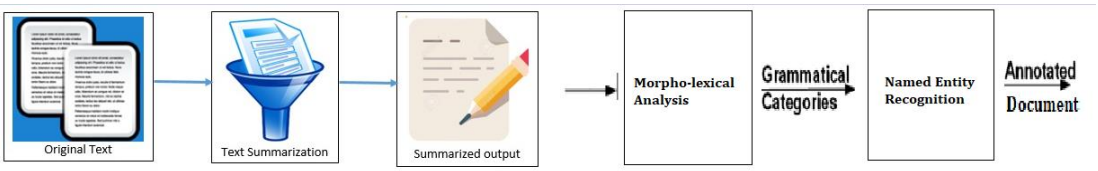


Figure A: 7 Identification of instance candidate process.

By using base ontology and the output a classifier is the main purpose of construction of a classifier. Class properties selection and relationships selection, triggers selection and rules generation are the main three tasks in this process.

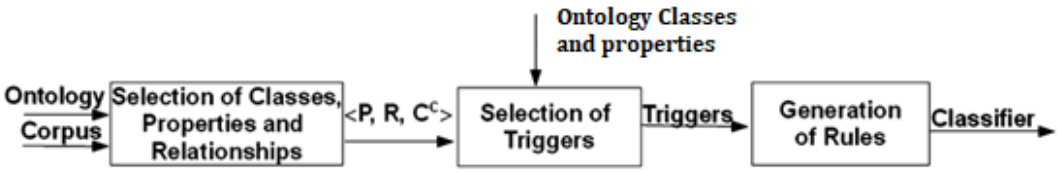


Figure A: 8 Classifier construction process

Classification of instances outputs a populated ontology using classifier and annotated summarized data generated from above steps. Association of instances and instantiation are the main tasks of this instances classification phase.

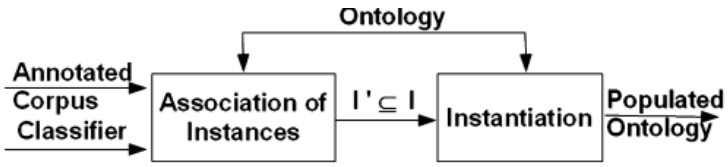


Figure A: 9 instances classification phase

Appendices B: Question and answer module

B.1 Introduction

This module mainly focusses on WH questions. There are two types of answer retrieval processes such as direct retrieval of answers by analyzing the content of ontology individuals and answer retrieval by RDF Tripels (subject-predicate-object) formulation from the question. Following sub sections will provide detail about some of the process descriptions.

B.2 Algorithm

Overall algorithm used for Simplification of question to normal form is

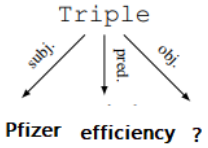
Input -: Question

Output -: RDF triples, with one hole (? :) by triple

1. Dependency tree computation
2. Name entity recognition to merge useful nodes and attach description on them (location, date etc)
3. Simplifications (merge, delete) on dependency tree as possible. Generate a new tree which uses a restricted set of dependencies (not 50 dependencies possible, as in Stanford dependency)
4. question words/ type of question identification
5. create the triples with use of question type
6. Add the triples involved by other parts of the tree
7. Output the conjunction of all the triples

B.3 Simplification of the dependency tree

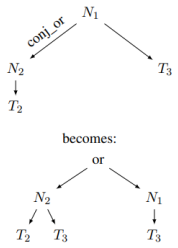
Normal Form



Linear representation: (Pfizer, efficiency, ?)

Figure B: 1 Possible normal form for What is the efficiency of Pfizer?

Global transformations



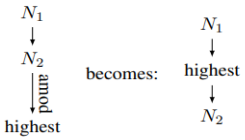


Figure B: 2 Remove conj_or dependencies

Figure B: 3 : Remove amod dependencies

B.4 Process of question answering system

In a process of question answering system, broadly there are three processes such as, question analysis/preprocess, knowledge analysis and answer analysis/processing

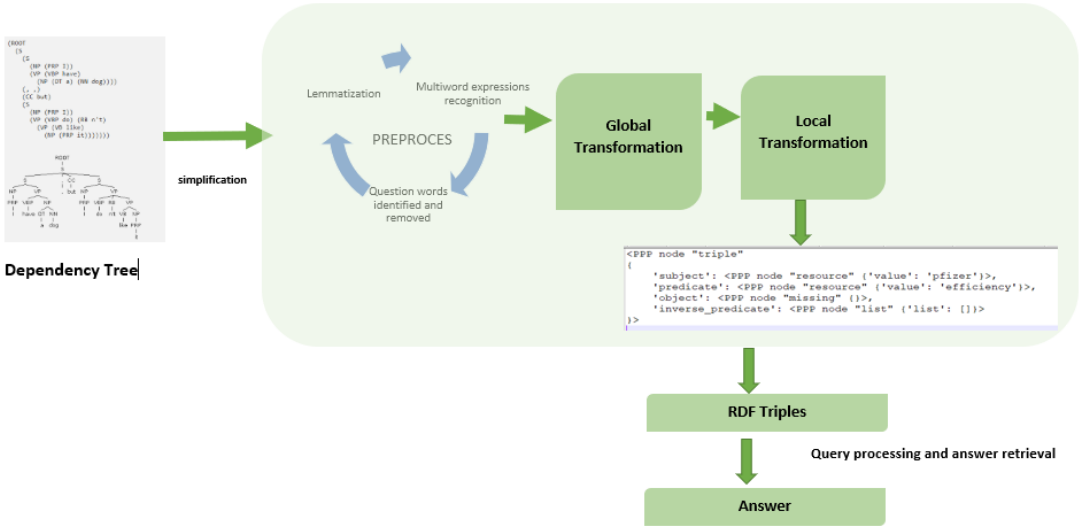


Figure B: 4 Block diagram of Question answering system

Appendices C: Sample Codes

C.1 Introduction

Following sub section will discuss how to use the software framework when implementing dynamic ontology-based Q&A system for pandemic situations. Please note that this will not cover each and every functionality, methods and features of the framework. Complete documentation on system will be available on-line.

C.2 Dynamic ontology module - Web scrapping

In order to detect real time changes in data sources such as news sites and publications, web scraping has used with python requests and beautiful soup module to extract data from html files.

```
urls = ['http://www.bbc.com/news/coronavirus',
        'https://www.news-medical.net/condition/Coronavirus-Disease-COVID-19',
        "https://www.cdc.gov/media/archives.html",
        "https://www.who.int/westernpacific/emergencies/covid-19/news-covid-19"]
for url in urls:
    page = requests.get(url)
    soup = BeautifulSoup(page.text, 'html.parser')
    domain = tldextract.extract(url).domain
    getContent(soup, time, domain,f, frame, count)
```

Web scrapping in dynamic web sites like twitter has used with python selenium, web driver and beautiful soup module to extract data from html files.

```
urls = ["https://twitter.com/hashtag/COVID19?src=hashtag_click",
        "https://twitter.com/search?q=%23covid",
        "https://twitter.com/search?q=%23CORONAVIRUS",
        "https://twitter.com/who"]
for url in urls:
    option = webdriver.ChromeOptions()
    option.add_argument('--headless')
    option.add_argument('--no-sandbox')
    option.add_argument('--disable-dev-sh-usage')
    driver = webdriver.Chrome(executable_path='C:\\chromedriver\\chromedriver', options=option)
    time.sleep(2)
    driver.get(url)  # Getting page HTML through request
    time.sleep(10)
    driver.execute_script("window.scrollTo(0,document.body.scrollHeight)")
    time.sleep(3)
    soup = BeautifulSoup(driver.page_source, 'html.parser')  # Parsing content using beautifulsoup
```

C.3 Dynamic ontology module - Differencing

In order to detect real time changes in data this framework uses web scraping technique. Here the web scrapping gets the difference of the current version and previous version of .csv format data, using ‘csv-diff’ python CLI library for diffing CSV and JSON files.

```
diff = compare(load_csv(open(path1), key="Header"),
               load_csv(open(path2), key="Header"))
```

C.4 Summarize the html documents

```
req = Request(url, headers={'User-Agent': 'Mozilla/5.0'})
    webpage = urlopen(req).read()
    parsed_article = bs.BeautifulSoup(webpage, 'html.parser')
    paragraphs = parsed_article.find_all('p')

    for p in paragraphs:
        article_text += p.text
        # Removing Square Brackets and Extra Spaces
        article_text = re.sub(r'[[0-9]*]', ' ', article_text)
        formatted_article_text = re.sub('[^a-zA-Z]', ' ', article_text)
        formatted_article_text = re.sub(r's+', ' ', formatted_article_text)
        sentence_list = nltk.sent_tokenize(article_text)
        stopwords = nltk.corpus.stopwords.words('english')

        for word in nltk.word_tokenize(formatted_article_text.lower()):
            if word not in stopwords:
                if word in word_frequencies.keys():
                    word_frequencies[word] += 1
```

```
    maximum_frequncy = max(word_frequencies.values())
    for word in word_frequencies.keys():
        word_frequencies[word] = (word_frequencies[word] / maximum_frequncy)

    sentence_scores = {}
    for sent in sentence_list:
        for word in nltk.word_tokenize(sent.lower()):
            if word in word_frequencies.keys():
                if len(sent.split(' ')) < 30:
                    if sent not in sentence_scores.keys():
                        sentence_scores[sent] = word_frequencies[word]
                    else:
                        sentence_scores[sent] += word_frequencies[word]
summary_sentences = heapq.nlargest(1, sentence_scores, key=sentence_scores.get)
```

C.5 Ontology Population

Populate the ontology with generated individuals using owlready2.

```
onto_path.append("C:\\ONTOLOGY")
Ontopath = 'C:\\ONTOLOGY\\CovidNewsOnto.owl'
onto = get_ontology(Ontopath)
onto.load()
instpath = "C:\\ONTOLOGY\\CovidNewsOnto." + istanceOf
for cls in onto.classes():
    if(str(cls) == instpath):
        detail = re.sub(r"[^a-zA-Z0-9]+", ' ', summery[0])
        dom = tldextract.extract(url).domain
        fulldetail = dom, ",", detail, ",", url, ",", str(datetime.datetime.now().date())
        cls(fulldetail)
        break
onto.save(file=Ontopath)
```

C.6 Stanford dependency tree simplification

Stanford dependencies provides a representation of grammatical relations between words in a sentence. To define the StanfordCoreNLP following code segment was used.

```
class stanfordParse:
    def __init__(self, host='http://localhost', port=6000):
        self.nlp = StanfordCoreNLP(host, port=port,
                                   timeout=30000)  # , quiet=False, logging_level=logging.DEBUG)
        self.props = {
            'annotators': 'tokenize,ssplit,pos,lemma,ner,parse,depparse,dcoref,relation',
            'pipelineLanguage': 'en',
            'outputFormat': 'json'
        }

    def parse(self, sentence):
        return self.nlp.parse(sentence)


    def dependency_parse(self, sentence):
        return self.nlp.dependency_parse(sentence)
```

C.7 Generate normal form of the dependency tree

Dependency tree parse to the **computeTree** for Compute the dependence tree. Take the result produced by StanfordNLP (if vaccine is this result, then stanfordResult = vaccine ['sentences'][0]) Apply quotation and NER merging Return the root of the tree (word 'ROOT-0'). Replace/push the spaces in the nodes of the tree.

```
handler = QuotationHandler()
nonAmbiguousSentence = handler.pull(sentence)
result = stanfordnlp.parse(nonAmbiguousSentence)
tree = computeTree(result)
handler.push(tree)
NamedEntityMerging(tree).merge()
PrepositionMerging(tree).merge()
qw = simplify(tree)
return normalFormProduction(tree, qw)
```

Once the tree computed, it sends to the NamedEntityMerging in order to merge child parent and merge sibling. MergeChildParent merge all nodes n1, n2 such that: n1 is parent of n2, n1 and n2 have a same named entity tag and not merge if the 2 words are linked by a conjunction.

```
def _mergeChildParent(cls, tree):

    for child in tree.child:
        cls._mergeChildParent(child)
    if tree.namedEntityTag == 'undef':
        return
    sameTagChild = set()
    for child in tree.child:
        if child.namedEntityTag == tree.namedEntityTag and not child.dependency.startswith('conj'):
            sameTagChild.add(child)
    for child in sameTagChild:
        tree.merge(child, True)
```

mergeSibling merge all nodes n1, n2 such that: n1 and n2 have a same parent, n1 and n2 have a same namedEntityTag (except conjunction) and n1 and n2 have a same dependency.

```
for child in tree.child:
    cls._mergeSibling(child)
tagToNodes = {}
for child in tree.child:
    if child.namedEntityTag == 'undef' or child.dependency.startswith('conj'):
        continue
    try:
        tagToNodes[child.namedEntityTag+child.dependency].add(child)
    except KeyError:
        tagToNodes[child.namedEntityTag+child.dependency] = set([child])
for sameTag in tagToNodes.values():
    x = sameTag.pop()
    for other in sameTag:
        x.merge(other, True)
```

```
        for child in tree.child:

            cls._mergeNode(child)

            if child.getWords() in cls.prepositionSet:

                tree.merge(child, True)
```

and mergeEdge replace a -prep_x→b by 'a x' -prep→ b if a is a verb, a -prep→ b otherwise and replace a -agent-> b by 'a by' -agent-> b

```
for child in tree.child:
    cls._mergeEdge(child)
    if child.dependency.startswith('prep'): # prep_x or prepc_x
        preposition = cls.getPreposition(child.dependency) # type of the prep (of, in, ...)
        if tree.isVerb():
            tree.appendWord(preposition)
        child.dependency = 'prep'
    if child.dependency == 'agent':
        assert tree.isVerb()
        tree.appendWord('by')
```

Once the merge is done, simplify the tree by identifying and removing question word, collapse dependencies of tree t. Then collapse the tree according to dependenciesMap1, remove conjunction connectors, remove remaining amod connectors, change the tree depending on the qw, propagate types from bottom to top, propagate types from top to bottom applied. Once the simplify is done to the tree, normalize/ map the tree to a normal form by replacing the dependency tag with a triple production tag.

```
qw = identifyQuestionWord(t)            # identify and remove question word
collapseMap(t, dependenciesMap1, qw)      # collapse the tree according to dependenciesMap1
conjConnectorsUp(t)                  # remove conjonction connectors
connectorUp(t)                       # remove remaining amod connectors
questionWordDependencyTree(t, qw)        # change the tree depending on the qw
collapseMap(t, dependenciesMap2, qw)      # propagate types from bottom to top
collapseMap(t, dependenciesMap2, qw, False) # propagate types from top to bottom
return qw
```