People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Akli Mohand Oulhadj, Bouira

Faculty of Sciences and Applied Sciences

Department of Computer Science

Thesis Submitted in Partial Fulfillment of the Requirement for the Master Degree

in Computer Science

Speciality: *Information System and Software Engineering*

Automatic text summarization for crisis management: Coronavirus(COVID-19) case

---

Supervised by

- AID Aicha

submitted by

- Karboua Sabrina

In front of the jury composed of

Mme. Chouiref Zahira	Professor, UAMOB	President
Mme.Mahfoud Zohra	Professor, UAMOB	Examiner
M. Bal Kamal	Professor, UAMOB	Examiner
Mme. Aid Aicha	Professor, UAMOB	Supervisor

2019/2020

Acknowledgments

This work would never see the light on this day without the support of many people.

First and foremost, I thank Allah for guiding me all the time. My life is so blessed because of his majesty.

I would like also to take this opportunity to express my gratitude to my respected supervisor Dr. Aid Aicha, Associate professor, Department of Computer Science , University of Bouira, for giving me the opportunity to work with her, I am thankful for her generous amount of guidance, support and care provided through the process of conducting this research. I am indebted to her for her valuable help, patience in every step of this work, she was and she will always be a source of inspiration and motivation for me.

Special thanks to the members of the committee for generously offering their time, guidance and good will throughout the preparation and review of this document.

I would not be where I am today without the tremendous amount of encouragement, support, pray, and love from my parents, sisters and my brother.

I would also like to thank my other amazing friends who, in one way or another, have helped me and was always there when I most needed it.

Dedication

I dedicate this work to my parents, sisters, and my brother, their unconditional love and support has made this work possible.

Sabrina Karboua

Abstract

In response to COVID-19 pandemic, plenty of scientific publications have been published to help the medical community to find answers to their important questions about the virus. The effort made by doctors, researchers and policy makers to shed light on the new virus is very important, but the huge number of these data implied a difficulty for information retrieval and made the process of gathering the most important information that the user looks for less valuable. Fortunately, Automatic Text Summarization is one of these solutions that can save the user's time and provide quickly what he is seeking for.

The objective of this thesis is to help the medical community by creating a simple and effective extractive query-based multi-document summarization system that from a large number of articles, extract the most relevant information corresponding to a specific query.

To achieve this work, the proposed solution is based on dividing the query based multi-document summarization process into two steps, the first step is single extractive query-document summarization, which depends on the use of semantic machine learning algorithm that uses a knowledge repository WordNet combined with information retrieval method BM25 OKAPI to generate a summary for each document, the set of the resulted summaries will be used in the second step which is an extractive generic multi-document summarization to create one single global summary using TextRank algorithm.

key words: Crisis, COVID-19, automatic text summarization, information retrieval, data mining, natural language processing, semantic similarity,Text-Rank, BM25 OKAPI.

Résumé

En réponse à la pandémie de la COVID-19, de nombreuses publications scientifiques ont été publiées pour aider la communauté médicale à trouver des réponses à leurs questions importantes sur le virus. L'effort fait par les médecins, les chercheurs et les décideurs politiques pour faire la lumière sur le nouveau virus est très important, mais le grand nombre de ces données impliquait une difficulté pour la récupération des informations et rendait le processus de collecte des informations les plus importantes que l'utilisateur recherche moins de valeur. Heureusement, le résumé automatique de texte est l'une de ces solutions qui peut faire gagner du temps à l'utilisateur et fournir rapidement ce qu'il recherche.

L'objectif de cette thèse est d'aider la communauté médicale en créant un système de résumé multi-document simple et efficace, basé sur des requêtes, permettant d'extraire à partir d'un grand nombre d'articles les informations les plus pertinentes correspondant à une requête spécifique.

Pour réaliser ce travail, la solution proposée est basée sur la division du processus de résumé multi-documents basé sur des requêtes en deux étapes, la première étape est le résumé extractif mono-document orienté guidé par une requête qui dépend de l'utilisation d'un algorithme d'apprentissage automatique sémantique qui utilise un référentiel de connaissances WordNet combiné avec la méthode d'extraction d'informations BM25 OKAPI pour générer un résumé pour chaque document. L'ensemble des résumés résultants sera utilisé dans la deuxième étape qui est un résumé multi-document générique et extractif pour créer un résumé global unique à l'aide de l'algorithme TextRank.

Mots clés: Crise, COVID-19, résumé automatique de texte, recherche d'informations, data mining, traitement du langage naturel, similarité simantique,Text-Rank, BM25 OKAPI.

ملخص

استجابةً لوباء COVID-19، تم نشر الكثير من المنشورات العلمية لمساعدة المجتمع الطبي في العثور على إجابات لأسئلتهم المهمة حول الفيروس. الجهد الذي يبذله الأطباء والباحثون لإلقاء الضوء على الفيروس الجديد مهم للغاية، لكن العدد الهائل من هذه البيانات جعل عملية جمع أهم المعلومات التي يبحثون عنها ذو قيمة أقل. لحسن الحظ، يعد التلخيص التلقائي للنص أحد هذه الحلول التي يمكن أن توفر وقت المستخدم وتوفر بسرعة ما يبحث عنه.

الهدف من هذه الأطروحة هو مساعدة المجتمع الطبي من خلال إنشاء نظام تلخيص متعدد المستندات بسيط وفعال، يستخرج من عدد كبير من المقالات المعلومات الأكثر صلة بسؤال معين.

لتحقيق هذا العمل، يعتمد الحل المقترح على تقسيم عملية تلخيص المستندات المتعددة القائمة على الاستعلام إلى خطوتين، الخطوة الأولى هي تلخيص استخلاصي لمستند واحد، والذي يعتمد على استخدام خوارزمية التعلم الآلي الدلالي التي تستخدم مستودعاً للمعرفة تم دمج WordNet مع طريقة استرجاع المعلومات BM25 OKAPI لإنشاء ملخص لكل مستند، وسيتم استخدام مجموعة الملخصات الناتجة في الخطوة الثانية وهي تلخيص عام استخلاصي متعدد المستندات لإنشاء ملخص واحد باستخدام خوارزمية Text-Rank.

الكلمات الرئيسية: أزمة COVID-19 ، ملخص تلقائي للنص ، بحث عن المعلومات ، استخراج البيانات ، معالجة لغة طبيعية ، تشابه دلالي ، Text-Rank، BM25 OKAPI.

Contents

List of Figures

List of Tables

List of abbreviations

ATC	Automatic Text Summarization.
BM25	Best Matching 25.
CZI	Chan Zuckerburg Initiative.
CSET	Center for Security and Emerging Technology.
ICT	Information and Communication Technology.
IDF	Inverse Document Frequency.
GIS	Geographic Information System.
LCH	Leacock CHordorow.
LCS	Least Common Sub-sumer.
MDS	Multi Text Summarization.
NLM	National Library of Medicine.
NLP	Natural Language Processing.
NLTK	Natural Language Toolkit.
OSTP	Office of Science and Technology Policy.
PMC	PubMed Central.
QCS	Query Cluster Summarization.
QFMDS	Query-Focused Multi Document Summarization.
ROUGE	Recall-Oriented Understudy for Gisting Evaluation.
SA	Situational Awareness.
SVM	Support Vector Machine.
TS	Text Summarization.

Introduction

On March 11, 2020, the World Health Organization declared COVID-19 as a pandemic pointing to over 118.00 cases of Coronavirus illness in over 110 countries around the world due to high rate spreads of the virus throughout the world.

With the ongoing outbreak of this virus, and in response to COVID-19, a lot of scientific literature and articles have been published, and rapidly increased at a rate of thousand per week to help medical researcher, doctors, clinicians, and policy makers to tackle the health emergency, better understand, mitigate and suppress its spread, developing vaccines and treatment. The challenge, however is that the growing number of papers with over 50.000 scientific publications makes extremely difficult and nearly impossible to go through this tremendous data and get useful insight about the new COVID-19 and gather as much information they are looking for as possible without making effort to read long documents, which in nowadays known as information overloaded problem, where the person faces problems in understanding or making decisions when faced with excessive amounts of information. Therefore, it's important to identify situational information and take benefits from them to combat this virus.

Creating a simple yet effective scientific retrieval system that takes the user's query(question in natural language) and analyzes a very large corpus of scientific papers to provide reliable information to the COVID-19 related questions is crucial in the current time-critical race to find cure for the virus. One of these solutions that works to address the problem presented above and can help enormously in providing useful information to their day to day questions is Automatic Text Summarization.

Automatic Documents Summarization(ADS) is defined as the process of creating a summary that briefly presents the important and relevant information existing within the original document(s). A summary can be generated from a single document or multiple documents. Summary can also be generic summary or tailored to present the user's specific query, which known as query-specific summary, it can also be extractive or abstractive summary; for extractive summary, the most important sentences are selected from the document source without any modification, the abstractive summary consists on producing new sentences. In this thesis, we are interested in proposing an extractive query-based multi-document summarization system because it takes into consideration the user's request.

The main objective of our work is to propose an effective method for creating query-based multi-document summarization system that produces from one or more documents a summary that contains the most important information relevant to the user's request. This system is based on COVID-19 open research dataset(CORD-19) proposed by KAGGLE, which is a resource of over 50.000 articles and will use Natural Language Processing, information retrieval and extraction methods, which are two fields that have focused on mining and analysis of large amounts of data and extraction of high quality information relevant to user's information needs.

The proposed query based multi-document summarization system basically works in two steps, extractive query-based single-document summarization and extractive generic multi-document summarization. In the first step, and for each document, the similarity score is calculated between the query and the sentences within the document using two algorithms, the first one is a semantic similarity algorithm that uses a knowledge repository named WordNet constructed by humans to enhance the machine understanding of human language and measure the semantic similarity between sentences, improved an information retrieval framework called BM25 OKAPI, the summaries results from this step will be used as input to the next step which will use the TextRank algorithm to generate a global summary.

The structure of the thesis

The structure of this thesis can be segmented into four chapters:

0. **Chapter 1:** presents the definition of crisis management as well as their steps to confront the crisis, then explains the role of information and communication technology during the crisis.

0. **Chapter 2:** presents the background and the basic concepts of Automatic Text Summarization, definition and its characteristics and the role behind using it. After that, it details its various type and finally explains the methods used to create an automatic text summarization.

0. **Chapter 3:** presents some related works and the dataset used, explains the approach used to achieve this work and detailed the proposed solution.

0. **Chapter 4:** in this chapter, the programming language and development environment is presented, the evaluation of the system is analyzed and discussed and finally the application web Django is displayed.

Finally, a general conclusion of this work is provided, and some perspectives that may perform our system will be addressed.

Chapter 1

Crisis management

1.1 Introduction

Crisis is a sudden adverse or unfortunate extreme event which causes great damage to human beings as well as plants, animals and infrastructure. Disasters occur rapidly, instantaneously and indiscriminately. These extreme events require to elaborate a strategy for rapid crisis response.

During crisis management, a massive amount of information are gathered and a lot of reports are generated. The ability to decipher through such a massive amount of data to extract useful information is a significant undertaking and requires an automatic mechanism like text summarization in order to provide a useful insight into time-critical situations to act on emergency responses in a timely manner.

This chapter is organized like that: first we define what does mean a crisis, describe the crisis management and its different phases. Then, we present the context of situation awareness in crisis management and give a brief overview of the role of information and communication technology for disaster management.

1.2 Definition of Crisis

In general, a crisis (or an emergency situation) is a sudden and unforeseen event that threatens the safety of a population, property or the environment and requires immediate interventions[1].

Crisis is also defined as: "a complex and dynamic phenomenon, which constitutes a threat to the survival of an organization and its members, which leaves little time to react, and which leads to an adjustment of the system". This definition is associated with crisis that threaten the survival of a company. In the case of crisis that threaten life of people, the following definition is associated: a humanitarian crisis is "any situation where there is an exceptional and large-scale threat to the life, health or basic subsistence of individuals and a community"[2].

Another definition states that "a crisis is a process which, under the effect of a triggering event, awakens dysfunctions, this dysfunction generates the moment of rupture determined by chaos, the moment of uncertainties, truncated and contradictory data and information. This gives rise to impacts that could be human and/or material"[3].

1.3 Crisis management cycle

Crisis management includes all the activities, programs and measures that can be taken before, during and after a disaster in order to avoid a disaster, reduce its impact or recover from its losses. The key four steps of activities undertaken in the context of disaster risk management are[4]:

0. **Mitigation:** pre-disaster actions taken to identify and reduce risks, protect people and structure, reduce the costs of response and recovery (for example: building dikes, establishing construction rules, risk mapping, etc).

0. **Preparedness:** because it is not possible to mitigate completely against every hazard that poses a risk, preparedness measures can help to reduce the impact of the remaining hazards and allows emergency managers and the public to be able to respond adequately by taking certain actions before an emergency event occurs includes plans or other preparations made to save lives and facilitate response and recovery operations (for example: alert systems, training exercises for rescuers, evacuation and rescue plans, etc).

0. **Response:** response begins when an emergency event is imminent or immediately after an event occurs to reduce human and material losses, response encompasses all activities taken to save lives and reduce damage from the event and includes: pro-

viding emergency assistance to victims (rescue operations plan), restoring critical infrastructure and ensuring continuity of critical services (law enforcement and public works).

0. **Recovery:** the recovery phase includes activities that return the affected geographic area back to normal state and allow the affected population to return to their normal social and economic activities.

1.4 The response cycle

Disaster response is one of the most important phases of disaster management and aims to provide immediate life support. In order to improve disaster response, it is important to increase knowledge on disaster management.

Irrespective of the nature and scale of the crisis and the organizations involved, crisis response activities can be viewed as consisting of four interrelated phases:

0. **Damage evaluation:** in this phase, disaster-related losses are identified on both incident level and regional scales, and their magnitudes are assessed. Severely impacted areas, disruptions to critical infrastructure, situations where secondary hazards may develop if initial damage is not mitigated and other problems of high urgency are identified, and estimates of the time needed to restore disrupted systems are developed.

0. **Needs evaluation:** in this phase, incidents requiring some level of response are identified. For example: building collapses where victims are trapped may require search, rescue and medical resources, release of hazardous materials may require large-scale evacuation, etc. Operationally, these incidents are assigned a measure of urgency/ priority, typically based on immediate threats to life safety.

0. **Prioritization of response measure:** in this phase, incidents requiring response are matched with available resources. If the total demand is greater than the systems capacity to respond as is invariably the case in large-scale disasters decision-makers must establish priorities for response.

0. **Organizational response**: in this phase, emergency resources are deployed and organizational decisions are disseminated to crisis-workers and the population at large. Ideally, response activities take place in accordance with pre-disaster planning [5].

1.5 Situational awareness for crisis response

During mass emergencies, affected populations construct an understanding of the situation based on incomplete information. Often, potential victims, members of formal response agencies, and/or concerned outsiders gather available information before they decide what action to take regarding an emergency. This process of gathering information, or situational assessment, leads to a state of situational awareness (SA).

Situational awareness is a state of knowing what is happening in your immediate environment and understanding what that information means for a particular situation including perception of the elements in the environment and how those elements relate to each other [6].

Several definitions and models of situation awareness have been proposed in the literature. The most known is the model defined by Endsley (1995).

Endsley model indicates that SA as a complex process of three hierarchically organized phases:

- **Perception of elements in current situation:** involves information gathering processes that would describe the current state of the components of the geographic area affected by the crisis.
- **Comprehension current situation:** involves integration and interpretation of this set of information collected to produce an understandable view of the current post-crisis situation.
- **Projection of future status:** the projection predicts the possible future state of these same elements that make up the environment in order to support rapid decision-making [1].

These three-levels lead to knowing what is happening in a given environment and understanding what a given information means in a given situation, including the perception of the elements constituting this environment and how these elements are interconnected.

McGuinness and Foy, extended the SA model proposed by Endsley by adding a fourth level called resolution. This level provides awareness of the best path to take and the best actions to take to resolve problems related to an emergency. They claim that perception is the attempt to answer the question «What are the current facts?»; Understanding asks «What is really going on?»; The screening asks «What is most likely to happen if ...?» and the resolution asks «What exactly do i have to do?»[1].

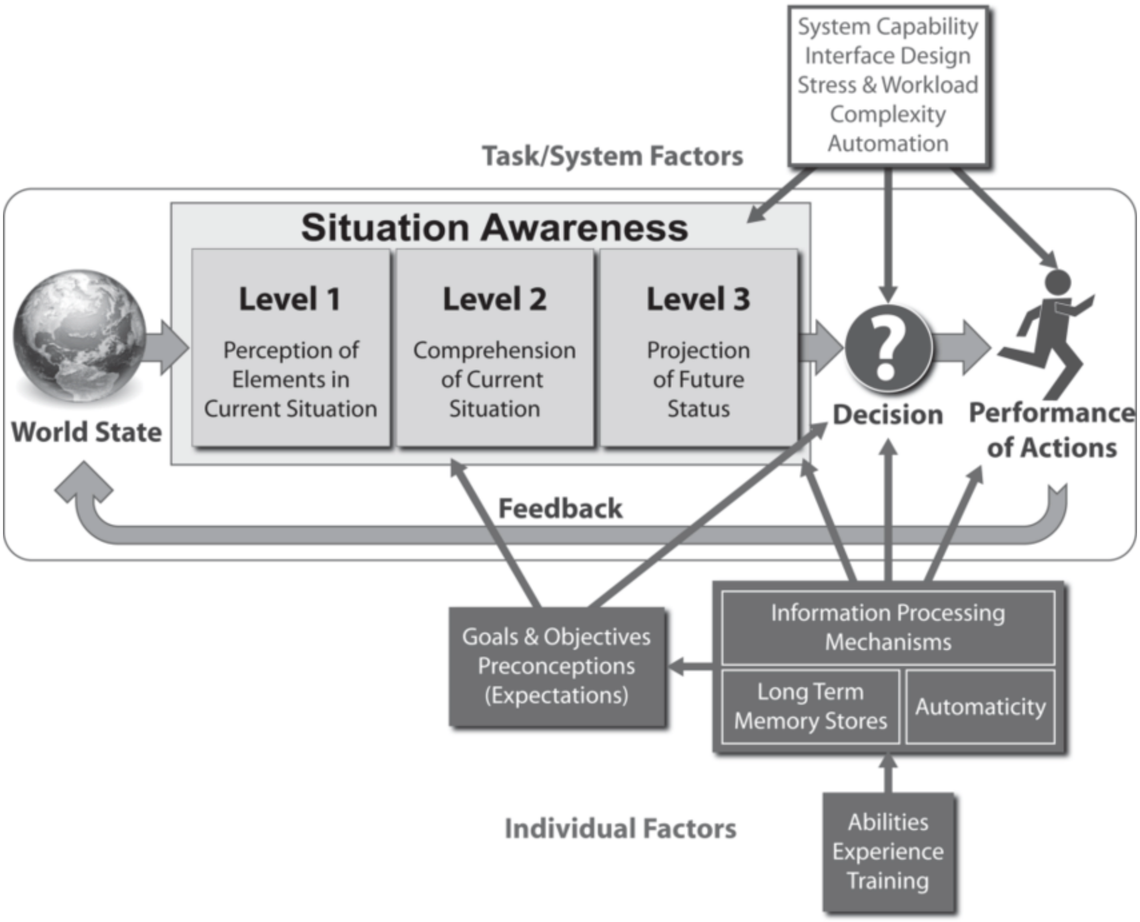


Figure 1.1: Situational awareness model proposed by Endsley(1995).

1.6 Information and communication technology for crisis management

Disaster management involves intensive information and communication activities before, during and after disasters. The revolutionary potential of ICT lies in their ability to instantly and continuously facilitate rapid communication and flow of information, capital, ideas, people and products. Because of this potential, information and communication technologies (ICT) are used more and more in the management of major disasters, with the aim of transmitting information and helping at the cost of decision. Their use has enabled the improvement of coordination in critical time, inter and intra-organizational collaboration and plays the crucial role of mediator of situational information between the multiple actors involved[1]. These technologies are used to:

a) Effectively alert using multiple communication channels.

b) Integrate situational information from heterogeneous sources.

c) Coordinate the various intervention operations.

d) Encourage social, institutional and public interventions.

e) Assess the damage caused by the crisis.

Indeed, information and communication technology in the form of Internet, GIS, remote sensing, satellite-based communication links, can help a great deal in planning and implementation of disaster risk reduction measures. These technologies have been playing a major role in designing early warning systems, catalyzing the process of preparedness, handling the situations and mitigating the losses. ICT tools are also being widely used to build knowledge warehouses using Internet and data warehousing techniques. These knowledge warehouses can facilitate planning and policy decisions for preparedness, response, recovery and mitigation at all levels. Similarly, GIS-based systems improve the quality of analysis of hazard vulnerability and capacity assessments, guide development planning and assist planners in the selection of mitigation measures. Communication systems have become indispensable for providing emergency communication and timely relief and response measures, also with the pervasiveness of social media networks, blogs, content-generation tools and photo and video sharing applications, users have become active entities rather than passive participants.

All of these models and solutions developed to assist situation-awareness in responding to a crisis demonstrate the importance of the valuable resource, which is situational information.

This resource is characterized by its heterogeneity, coming from various sources and in different formats. It may be:

a) Emergency action plans.

b) Continuous situational reports.

c) Damage analysis reports.

d) Geographic data and maps of the affected area.

e) Information on the condition of roads/bridges/airports and other infrastructure such as electricity, fuel, hospitals, schools, etc.

f) Logistical information on food/water/medicine deliveries.

g) Financial data to manage donations.

h) Satellite images of the affected area after the crisis and other multimedia data like video.

In addition to the uniqueness of the content, disaster management data also have different temporal/spatial characteristics and can be classified into three categories different types: spatial data, temporal data, and spatio-temporal data.

The impact of increased situational information has resulted in information overload. Making sense of the vast amount and types of this information is becoming harder and harder. This is particularly apparent in the aftermath of disasters, where time is of the essence, in making key decisions based on best available information at the same time that the information itself is continually evolving. This can often be a matter of survival for those affected by the crisis. To solve this problem, the application of well-studied information technologies to this unique domain are required. The data analysis technologies that are generally used for disaster-related situations are[7]:

- **Information Extraction (IE):** disaster management data must be extracted from the heterogeneous sources and stored in a common structured (e.g., relational) format that allows further processing.
- **Information Retrieval (IR):** users should be able to search and locate disaster-related information relevant to their needs, which are expressed using appropriate queries. It is the task that we are interested in and we will working on.
- **Information filtering (IF):** as disaster-related data arrives from the data producers (e.g., media, local government agencies), it should be filtered and directed to the right data consumers. The goal is to avoid information overload.
- **Data mining (DM):** collected current and historic data must be mined to extract interesting patterns and trends. For instance, classify locations as safe/unsafe.
- **Decision Support:** analysis of the data assists in decision-making. For instance, suggest an appropriate location as ice distribution center .

1.7 Conclusion

In this chapter, we have presented the basic concepts of crisis and disaster management, as well as their phases (mitigation, preparation, response, and recovery). We also detailed the response phase and its process. Then we have talked about situation-awareness and how it can help to carry out relief efforts. We have also explained the role of information and communication technologies during crisis management.

In the next chapter, we will introduce Automatic Text Summarization (ATS) and its different techniques.

Chapter 2

Automatic text summarization

2.1 Introduction

Due to great amount of information we are provided with and the huge number of articles which are published everyday, investigating and monitoring of all the topics are not possible in a short time, therefore the need of producing summaries have become more and more widespread.

A summary can be defined as a text that is produced from one or more text(s), that contains a significant portion of information in the original text(s). Text summarization is the process of extraction the most important information from a source to produce an abridged version for a particular user and task. When this is done by means of computer i.e automatically, we call this Automatic Text Summarization.

2.2 Definition of text summarization

The literature provides various definitions of text summarization:

DEFINITION 1.1: According to Radev et al, «a summary is a text produced from one or more texts, which contains important and significant information in the original text(s), and this does not represent more than half of the original text(s) and generally less than that»[8].

DEFINITION 1.2: Radev et al add that : Text Summarization (TS) is the process of identifying salient concepts in text narrative, conceptualizing the relationships that exist among them and generating concise representations of the input text that preserve the gist of its content [9].

According to Horacio Saggion and Guy Lapalme , in terms of function, a summary is:

DEFINITION 1.3: a condensed version of a source document having a recognizable genre and a very specific purpose: to give the reader an exact and concise idea of the contents of the source[9].

2.3 Characteristics of a summary

A summary must have some characteristics: conciseness, coverage, fidelity, cohesion and consistency. We expose these characteristics as they were defined in[10]:

- **Conciseness:** it's related to the reduction rate which is the ratio between the length of the source text and length of the summary. In general, the reduction rate is proportional to the restrictive nature of criteria used to generate the summary. For example, an indicative summary produce fairly short summary that includes just the main idea of the text. However, an informative summary contains more details about the subject and will in most cases lead to longer summaries.
- **Loyalty:** it is also an important criteria for characterizing a summary. It represents the relationship of objective similarity between the summary and the source text. It is a sort of measure of the overall quality of the summary. The notion of fidelity includes coverage as a component. As a general rule, a summary with correct coverage will be fairly faithful to the source text.
- **Coverage:** is a kind of ratio between the number of themes or elements present in the source text and those present in the summary. The nature of the elements depends on the type of summary considered: for an indicative summary, only the themes addressed will be retained. In the case of an informative summary, coverage is more difficult to determine.

- **Cohesion and consistency:** The last two criteria defining a summary are closely linked to the notion of text itself. Cohesion can be seen as the result of the application of mechanisms aimed at maintaining a referential and argumentative(use of connectors) unity. Coherence, on the other hand, derives more from the correct application of rhetorical.

2.4 The need for automatic summarization

With the huge amount of information generated every day in form of web pages, articles, etc. It is difficult to find the information desired by the manuals ways. The purpose of automatic text summarization is to extract the main points from the original text without having to read the entire document while selecting the most salient sentences, minimize information redundancy and make the summary readable and understandable, which allows [9]:

- Summaries reduce reading time.
- When researching documents, summaries make the selection process easier.
- Automatic summarization improves the effectiveness of indexing.
- Automatic summarization algorithms are less biased than human summarizers.
- Personalized summaries are useful in question-answering systems as they provide personalized information.

2.5 Steps of automatic text summarization

There are three main steps to summarize a document: identification, interpretation and generation of summary[11].

- **Identification:** in this step the most important informations in the text are identified. There are different techniques for identifying subjects, methods based on the position of sentences are the most useful methods for identifying subjects.

- **Interpretation:** the important informations that are identified in the first step are combined more cohesively. During this step, some modifications of the original sentences may prove necessary.

- **Generation:** the result of the second step is a summary which may not be consistent to the reader. Consequently, the aim of this step is to reformulate the extracted summary into a new coherent and understandable text .

2.6 Concept of salient sentences and scoring sentence

A principal concern in extractive document summarization is the selection of the most important content that can represents a document for inclusion in summary output, those sentences are known as **salient sentences**. The focus of these research areas are addressed by the following question: how can a system determine which sentences are representative of the content of a given text? In general, three approaches are followed: (i) Word scoring – assigning scores to the most important words; and (ii) Sentence scoring – verifying sentences features such as its position in the document, similarity to the title, etc.; and (iii) Graph scoring – analyzing the relationship between sentences. Based on those sentences, we can evaluate the pertinence of a summary. To do that, different ranking algorithms have been used to decide which sentences are more important and tend to be selected as summary sentences by assigning a pertinence score for each sentence. The score for each sentence is simply the linear combination of the weights given for each feature. The highest scoring sentences are selected as candidates to be part of the summary in the last stage. The following section presents the main methods in each of the aforementioned approaches:[12].

1- Word scoring: The initial methods in sentence scoring were based on words. Each word receives a score and the weight of each sentence is the sum of all scores of its constituent words. The approaches in the literature are: word frequency, TF/IDF, upper case, proper noun, word co-occurrence and lexical similarity

2- Sentence scoring: This approach analyzes the features of the sentence itself and was used for the first time in 1968 (Edmundson, 1969) analyzing he presence of cue words in sentences. The main approaches that follow this idea are: cue-phrases, sentence inclusion of numerical data, sentence length, sentence position, sentence centrality, Sentence resemblance to the title.

3- Graph scoring: In graph-based methods the score is generated by the relationship among the sentences. When a sentence refers to another it generates a link with an associated weight between them. The weights are used to generate the score of sentence using: text rank, Bushy path of the node and Aggregate similarity,

2.7 Various types of summaries

Automatic summaries and their methods can be categorized according to different criteria. We will quote the most known and used[13]:

2.7.1 Summary based on input

In terms of input, a summary may be based on:

A) **single document:** early attempts in summarization were mainly based on a single document summary, in which systems produced a summary from a single document.

B) **Multiple documents:** the system generates a summary for a set of documents that share a similar subject. This type is more difficult than the single document one. Indeed, the system should remove any redundancies between documents and also reconcile the content into a coherent summary .

2.7.2 Summary based on details

A) **Indicative summary:** an indicative summary system gives a global perspective of the subjects covered by the text, it presents only the most important idea of the text. This type of summary helps the user to decide on whether or not to read the document, indicating the themes addressed and developed in the source document, without considering the details. Summary's length depends on the compression rate.

B) **Informative summary:** the informative summary system is considered to be an abridged version and covers all aspects of the main text preserving the general organization of the source document. Its main objective is to inform the reader about the main quantitative and qualitative information of the text. The length of the informative summaries is approximately 20-30% of the original text .

2.7.3 Summary based on output

A) **Extractive summary:** an extractive summary is generated by selecting the relevant and important sentences as they appear in the source document and concatenating them without any modification. It is a simple and robust way to produce a summary. However, it come with the risk of producing incoherent text because the selected sentences may not share a semantic relation between them.

B) **Abstractive summary:** in this type of summary, natural language generation techniques are used. We try to understand the original document by identifying the key concepts and then produce new sentences that are grammatically correct, concise, consistent and which should give a result close to a human summary .

C) **Hybrid summary:**This type of summary merges between both the extractive and abstractive approaches to generate the output text.

2.7.4 Summary based on content

A) **Generic summary:** aims to summarize the source text without noticing its domain and context.

B) **Query-based summary:** the summary is generated by selecting a sentence which corresponds to the user's query. Sentences that are relevant to the query receive a higher chance of being retrieved for the final summary. Query-based summary systems, however, do not provide overall view of document's concepts because they focus on the user's query.

C) **Domain specific summary:** it provides a summary according to the specific domain, for example: summarizing news articles, web pages and biomedical documents. This type of summary requires domain-specific knowledge to select sentences for summaries.

2.7.5 Summary based on language

A) **Mono-lingual summary:** the source document and the summary are written in the same language.

B) **Multi-lingual summary:** the system can process multiple languages, and produce summaries in the same language as the the source document.

C) **Cross-lingual summary:** the language of source document and the summary must be different .

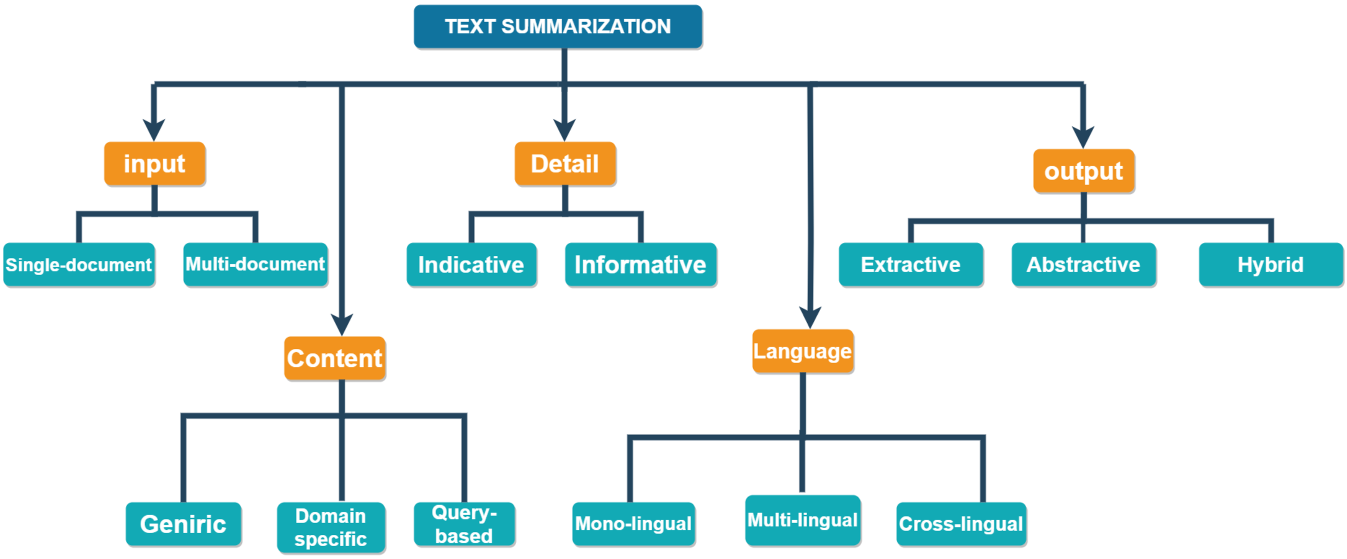


Figure 2.1: Text summarization classification

2.8 Update summarization

With the ever increasing popularity of news search engines, displaying the information in a more practical and pleasant way is becoming a challenging and important issue. One possible solution is to summarize multiple news so as to propose only one short text.

This is, intuitively, a reasonable solution though producing summaries from large collection of documents is a very complicated task. However, as the number of documents increases, facts that are considered as important and have to appear in the summary also become more numerous. In this case, a choice must then be made to drop important facts in order to satisfy size constraints. One way to tackle this problem is to remove facts that the user is already aware of. This variant of text summarization is called update summarization. More formally, update summarization is the task of producing summaries while minimizing redundancy with previously read documents. Indeed, segments have to be selected according to their salience but also to their ability to capture novelty.

The most intuitive way to go about update summarization would be to be identify temporal references within documents(dates, elapsed times, temporal expressions, etc.) and to construct a time-line of the events. It is a complex task as temporal references depend on surrounding elements in the discourse but also require an understanding of the ontological and logical foundations of temporal reference construction. Assuming the time-line is constructed, update summaries could be produced by assembling sentences containing the most recent events[14].

2.9 Text summarization methods

The four main approaches for summarization are the statistical, the graph-based, machine learning and swarm intelligence approaches.

- **Statistical based Approaches:** were the first introduced in ATS. These techniques are independent of any language,then it can summarize text in any language. They concentrate on statistics got from non-linguistic features of the document,like term frequency, position and length, cue phrases, title words, resemblance of sentence to the title, and sentence position. A statistical approach does not require any training dataset and doesn't need much language dependent tools; just some basic NLP tasks such as sentence segmentation, tokenizaion, stop word elimination, and stemming. Mostly, it is easy to be implemented and does not need a lot of processing power. So, for each sentence in the document, a score is computed and highly scored sentences are chosen for generating the summary, mostly these features are combined together in hope to increase sentence relevance[15].

- **Machine learning based approaches:** learn from the data. They can be supervised, unsupervised or semi-supervised. In supervised approach, there is a collection of documents and their respective human-generated summaries such that useful features of sentences can be learnt from them. Supervised or trainable summarizers classify each sentence of the test document either into «summary» or «non-summary» class with the help of a training set of documents. Large amount of labeled or annotated data is needed for learning purpose. Support Vector machine (SVM), Naïve Bayes classification, Mathematical Regression, Decision trees, Neural networks (Multilayer Perceptron), are some of the supervised learning algorithms. On the other hand, unsupervised systems do not require any training data. They generate the summary by accessing only the target documents. They try to discover hidden structure in the unlabeled data. Thus, they are suitable for any newly observed data without any advanced modifications. Such systems apply heuristic rules to extract highly relevant sentences and generate a summary. Semi-supervised learning techniques require labeled and unlabeled data both to generate an appropriate function or classifier[16].

- **Semantic based approaches:** semantic approaches mainly focus on the semantic analysis and correlations among sentences. They identify the relationship between words and sentences by use of thesaurus, ontologies, part of speech tagging, grammatical analysis and selection of meaningful sentences to generate summary. Various techniques like lexical chain, cluster and graph-based methods have being developed in this approach. Semantic-based methods are useful since they consider the meaning of each sentence and word that make a coherent and meaningful summary but using these techniques are time-consuming and require more efforts than the other techniques[11]. In the following paragraph, graph based method will be explained because it will be used in our proposed solution.

Graph based method: In the field of natural language processing, graphs are used to display the structure of the text and the connection between sentences. Sentences are represented as a node, and the relation between sentences are depicted by edges. The graph-based text summarization method is a technique to extract a significant, appropriate, and informative text in a compressed version. In order to use this technique, a pre-processing phase should be done on the input text to remove stop words, tokenize the sentences, and so on. Then sentences are ranked to identify the important sentences. Afterward, the relation between sentences is computed to recognize the relevant sentences. At the end, sentences are extracted for the summary based on the ranked and relevant sentences [17].

- **Swarm intelligence approaches:** in a computational context, a swarm is a group of simple agents that have a collective behavior to perform a complex task by acting as a community.Swarm Intelligence algorithms have recently sprung up as a family of nature-inspired, population methods that are capable of producing low cost, fast, and robust solutions to several complex problems.They can, therefore, be defined as a relatively novel branch of Artificial Intelligence (AI) that is employed to model the collective behaving of societal swarms in nature, for example the social behavior of ants, termites, bees, and other social beings have motivated researchers to discover their lifestyle, they noticed that they are moderately innocent with restricted capacities on their own, they are interacting together with certain behavioral patterns to cooperatively accomplish tasks necessary for their survival. The social connections among swarm individuals can be either direct or indirect Some swarm-based algorithms such as particle swarm optimization, cuckoo optimization algorithm, and bacterial foraging optimization have been introduced in text summarization [18].

Approach	Method	Advantages	Disadvantages
Statistical	- Cue phrase - Title words - Sentence location - Word frequency	- Simple to perform	- Don't consider the meaning of word on the text. - Duplication in summary sentences.
machine learning	- Naive bases method - Artificial neural network. - Fuzzy logic	- Comprehensive summary is produced.	- Depend on the training dataset. - Complexity in computation
semantic based	- Lexical chain - Clustering - Graph based method	-Reduce redundancy in multi-documents summarization.	- Limit each sentence to be put only in one cluster
Swarm intelligence	- Particle swarm optimization. - Cuckoo optimization algorithm. - Bacterial foraging optimization method.	- Better performance compared with other. method	- Complexity in computation.

Table 2.1: Comparison between different Automatic Text Summarization approaches that exist according to their advantages and disadvantages.

2.10 Evaluation techniques of text summarization

Evaluation methods can be classified into two types: intrinsic and extrinsic method:

2.10.1 Intrinsic evaluation

Its purpose is to evaluate the performance of automated summary content by comparing it with an ideal summary,it have assessed mainly the coherence and informativeness of summaries. There are two metrics for the intrinsic metric.

First measure involves **quality evaluation**, which tries to verify that the summary is syntactically correct, has no grammatical errors, logically articulated, has structural consistency

and does not contain redundant information.
Second measure is **content evaluation** which are divided into two groups: **co-selection** measure and **content-based** metric [19].

The co-selection measure can count as a match only exactly the same sentences. This ignores the fact that two sentences can contain the same information even if they are written differently. Furthermore, summaries written by two different annotators do not in general share identical sentences. It consists of a precision, recall, and F-measure.

- **Precision:** it reflects how relevant the selected data is. It is computed by the intersection of summarized extracted, and ideal sentences, divided by all extracted sentences.
- **Recall:** is computed by the intersection between the relevant and retrieved sentences, divided by all the relevant sentences.
- **F-measure:** is an average of precision and recall criteria, and determines the score of the final set of the selected sentences in a produced summary.

On the other hand,Whereas co-selection measures cannot do this, content-based similarity measures can. The most used content-based metrics are listed as follow:

- **Cosine similarity:** it measures the angle between two vectors that represent the sentences. Assuming that X and Y are the automatic text summary and the reference summary respectively.
- **Rouge (Recall-Oriented Understudy for Gisting Evaluation):** evaluate the quality of the summary by comparing it with a human-generated summary; it counts the number of units in common between a particular summary and a collection of reference summaries established manually to obtain more precise measures of quality summaries.

2.10.2 Extrinsic evaluation

The extrinsic evaluation metric is known as task-oriented (task-based) metrics. These metrics do not analyze sentences in the summary. They try to measure the prospect of using summaries for a certain task. Various approaches to task-based summarization evaluation can be found in literature. question-answering, classifying texts and information retrieval are examples of the extrinsic method. Their objective is to evaluate a summary performance based on a special task [19].

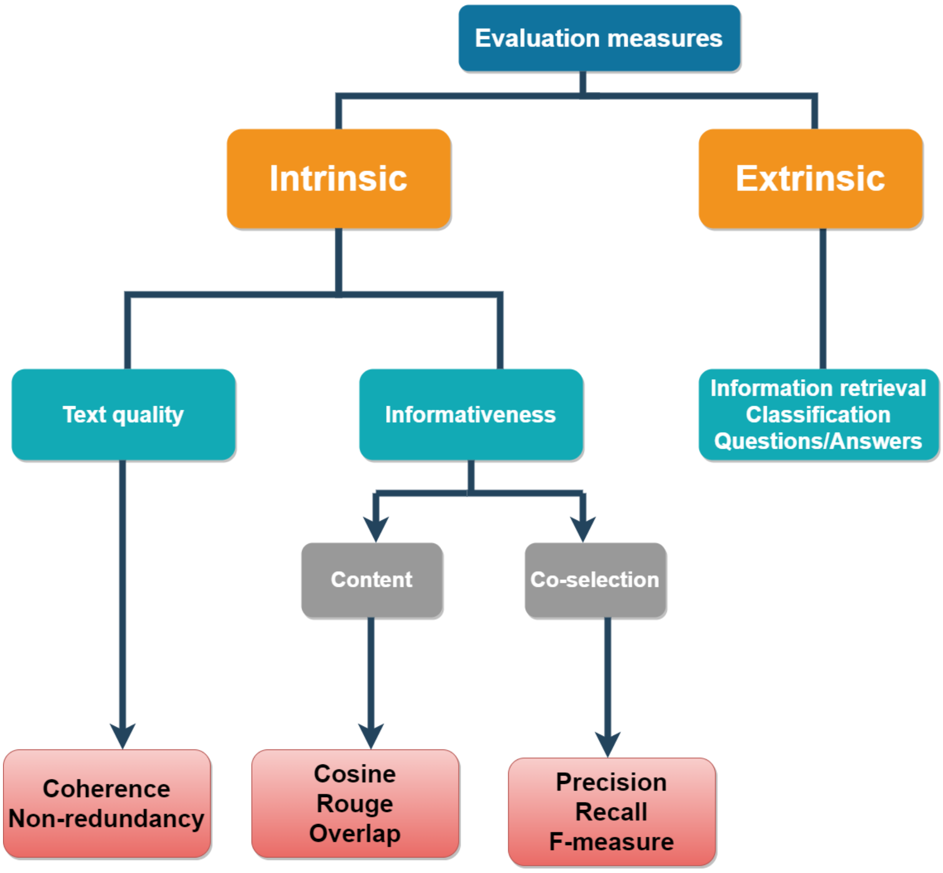


Figure 2.2: Evaluation techniques of Automatic Text Summarization

2.11 Conclusion

In this chapter, we have explained what does mean an automatic text summarization and what are its characteristics, we have given an overview about different classifications for a summary based on their input, output, content, purpose and language. We have also talked about the update summarization and the concept of salient sentences, then a various techniques like statistical, machine learning, semantic-based and swarm intelligence based methods were described. Finally, some evaluation method were introduced, which could be used to examine and compare the results of different approaches.

Chapter 3

Approach and solution

3.1 Introduction

The ongoing of COVID-19 pandemic involved a rapid increase in the volume of corona virus literature. The large amount of this data makes it difficult for members of medical community to find what they need, yet before we can take benefits from these massive amount of data, we often have to face a challenge of how to grab the essential information and knowledges quickly. Thus, creating a data mining tool that can analyze a very large corpus of scientific papers and return a specific text contains answers to open questions that may help the medical community to combat this pandemic is very essential. Fortunately, automatic query based multi-documents summarization can be one of those solutions that can resolve this problem.

Query-focused multi-document summarization (QFMDS) methods have been proposed as one such technique that organizes and presents information to users in an effective way. The goal of query-focused multi document summarization is to create a short summary from a set of documents that answers a specific query.

This chapter cites some related works and presents our problem. Then, describes briefly the dataset used. After that, the proposed solution to resolve this problem is presented in detail by explaining the main approaches used, and dive deeper into the system's architecture on a conceptual level. Finally its implementation will be covered.

3.2 COVID-19 crisis

The COVID-19 pandemic, also known as the Coronavirus pandemic, is an ongoing global pandemic caused by newly discovered Coronavirus (severe acute respiratory syndrome) and spreads primarily through droplets of saliva or discharge from the nose when an infected person coughs or sneezes.

This pandemic is the defining global health crisis of our time and the greatest challenge we have faced since World War Two. The virus has spread to every continent except Antarctica. Since 31 December 2019 and as of 08 July 2020, 11 801 805 cases of COVID-19 have been reported, including 543 902 deaths [20].

Most people infected with the COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. Older people, and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease and cancer are more likely to develop serious illness. Most common symptoms of this pandemic are fever, dry cough, tiredness. Less common symptoms are: aches and pains, sore throat, diarrhea, conjunctivitis, headache, loss of taste or smell, a rash on skin, or discoloration of fingers or toes and Serious symptoms are difficulty breathing or shortness of breath, chest pain or pressure, loss of speech or movement. On average symptoms takes 5 to 6 days from when someone is infected with the virus to show, however it can take up to 14 days [21].

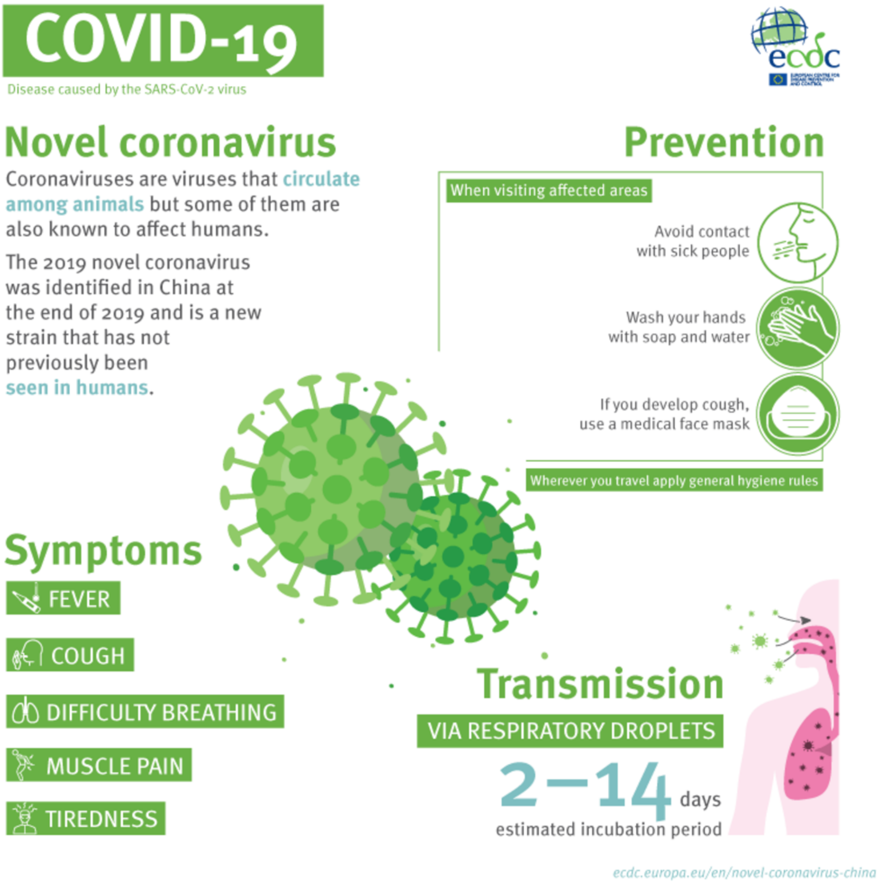

COVID-19
Disease caused by the SARS-CoV-2 virus
ecdc
Novel coronavirus
Coronaviruses are viruses that circulate among animals but some of them are also known to affect humans.
The 2019 novel coronavirus was identified in China at the end of 2019 and is a new strain that has not previously been seen in humans.
Prevention
When visiting affected areas
Avoid contact with sick people
Wash your hands with soap and water
If you develop cough, use a medical face mask
Wherever you travel apply general hygiene rules
Symptoms
FEVER
COUGH
DIFFICULTY BREATHING
MUSCLE PAIN
TIREDNESS
Transmission
VIA RESPIRATORY DROPLETS
2–14 days
estimated incubation period
ecdc.europa.eu/en/novel-coronavirus-china

Figure 3.1: COVID-19 disease caused by SARS-CoV-2 virus

COVID-19 is considered much more than a health crisis, it's also an unprecedented socio-economic crisis, by stressing every one of the countries it touches, it has the potential to create devastating social, economic and political effects that will leave deep scars [22], this is why every country needs to act immediately to prepare, respond and recover this crisis.

In response to this pandemic, a lot of scholarly articles have been published recently and made freely available. In the first 100 days of 2020, over 5,000 research articles were published related to SARS-CoV-2 or COVID-19, together with articles about similar viruses researched before 2020, the body of research exceeds 50,000 articles. This results in a considerable burden for those seeking information about various facets of the virus and finding answers to various questions regarding COVID-19, including researchers, clinicians, and policy-makers. Thus, developing text and data mining tools that can help the medical community develop answers to the COVID-19 related questions from the latest academic resources is crucial, especially for the medical community in the current time-critical race to treat patients and to find a cure for the virus.

3.3 Examples of Automatic Summarizers

Here some of the most common academic summarization systems are presented with a brief description about each one. The table below presents a summary of the main public automatic summarizers and highlights their features.

System	Description
MEAD	- Multi document multi lingual system.
Open Text Summarizer	- Multi document – Multi lingual system.
QCS	- Query, Cluster and Summarize Multi-document.
FastSum	- Query-Based Multi-Document Summarization.
MultiSum	- Query-Based Multi-Document Summarization.

Table 3.1: Examples of automatic text summarization system

3.3.1 MEAD

MEAD is a multi-document extractive summarizer that scores sentences according to a linear combination of features including centroid, position and first sentence overlap. These scores are then refined to consider cross-sentence dependencies, chronological order and user supplied parameters. Initially, documents are segmented into clusters with a distinctive theme covering each cluster. Then, all input documents are represented with TF-IDF vectors. Other features are also factored in at subsequent stages to help assign a score to each sentence [23].

3.3.2 Open Text Summarizer

The Open Text Summarizer is an open-source tool that analyzes texts in various languages and tries to present the most important parts of the text and present them in a summary. It works by first removing stop words from the text and stemming all terms. Then, a weight is assigned to each word based on its frequency and sentences with highest weighted terms are chosen for the summary. It has a downloadable version in addition to an on-line one. In addition, it ships with several Linux distributions such as Ubuntu and Fedora.

3.3.3 QCS

Given a query, the Query, Cluster and Summarize (QCS) system separates the retrieved documents into topic clusters and creates a summary for each cluster. LSA is used for documents retrieval, spherical k-means for clustering and a HMM-based module for extractive summarization. The system has an on-line demo with limited access to only the DUC collection dataset and MEDLINE documents [24].

3.3.4 FastSum

FastSum is a fast query-based multi-document summarizer called based solely on word-frequency features of clusters, documents and topics. Summary sentences are ranked by a regression SVM. The summarizer does not use any expensive NLP techniques such as parsing, tagging of names or even part of speech information. It only involves sentence splitting, filtering candidate sentences and computing the word frequencies in the documents of a cluster, topic description and the topic title [25].

3.3.5 MultiSum

Open-domain query based multi-document summarization system which combines existing techniques in a novel way such as Multi-Layered Architecture, Sentence Ordering Model, Heuristic Sentence Filtering and paragraph Clustering. The system is capable of automatically identifying query-related on-line documents and compiling a report from the most useful sources, whilst presenting the result in such a way as to make it easy for the researcher to look up the information in its original context[26].

3.4 Overview of the problems addressed and the proposed solution

The core aim of any MDS system is outputting a coherent and relevant summary from multiple resource of information. The query based MDS systems mentioned above in the previous section have used statistical and heuristic metrics to find the most informative sentences in a document. In our proposed solution, we will use another information retrieval ranking model called **Okapi BM25**.

Okapi BM25 is a probabilistic information retrieval ranking method that ranks documents based on relevance score which is the probability of a document being relevant to the input query. The main advantage which makes it popular is its efficiency. The BM25 score is calculated based on two main components: TF and IDF. However, there are some techniques for document length normalization and satisfying the concavity constraint of the term frequency. Based on these heuristic techniques, BM25 often achieves better performance compared to TF-IDF. Even if Okapi BM25 does a good job but it has a one convenient consists on the fact that finding the most informative sentences in a document rely on the presence of words query in the document, thus the system is always limited to the words explicitly mentioned within the text and can't detect the implicit relationships between words in a document, with such system, there are high possibility of loss of coherence and ambiguity in sentences. Without the ability to find the similarity and relatedness between terms like "sick" and "ill", the system would treat these terms as two unrelated entities and this my affect the judgment of their importance in a document.

The ability to detect such implicit relationship between terms in a document and resolve the problem of machine understanding requires an external knowledge .One of the main goal of using external repositories is to be able to apply some reasoning on a text document by measuring semantic distance between units. Semantic distance is a generic measure used to define how close or distant two units of text are in term of their meanings, the units can be words, group of words, sentences or paragraphs. To resolve the problem of machine understanding, we propose providing a system with knowledge repositories constructed by humans, the model proposed doesn't require training once the required features from knowledge repositories are constructed and build. Among the closed repositories is **WordNet** which has been used to enrich the understanding of text documents for different types of applications.

Generally, our solution consists on two steps, in the first step we try to use the best things of both approaches and combine them to overcome the weaknesses they have. By combining the classic search algorithm BM25 with the semantic similarity, we benefit from strength of each method to get the most accurate and relevant results, and in the second step we will use **Text-Rank** algorithm to generate a single summary.

3.5 WordNet

WordNet is the product of the research project that was conducted at Princeton University to create a model of a native speaker lexical knowledge and store it in a machine-readable dictionary [27]. It is an on-line database including nouns, verbs, adjectives and adverbs grouped into sets of cognitive synonyms (synsets). Synset represents a specific meaning of a word. It includes the word, its explanation and its synonyms. WordNet words are usually represented in a specific format with each word tagged with its Part-of-Speech(POS). Four letters are used to represent the four available POS types in WordNet: n for nouns, v for verbs, a for adjectives and r for adverbs.

Words senses and synsets are connected via a variety of relations. The relations connecting words senses are called semantic relation while those connect synsets are Lexical Relations [27]. For example, nouns have the following semantic relations:
Hyponym/Hypernym (IS-A , HAS A)
Meronym/Holonym (Member-of, Has-member,Part-of, Has-Part)

The latest on-line version of WordNet contains 155,287 words and 117,659 synsets. The majority of the words are nouns with a count of 117,798. The number of verbs is 11529, while adjectives and adverbs are 21,479 and 4,481 respectively.

part-of-speech	unique string	synset	Total WordSense pair
Noun	117798	117798	146312
Verb	11529	11529	11529
Adjective	21479	21479	21479
Adverb	4481	3621	5580
Total	155287	155287	155287

Table 3.2: WordNet database Statistics

3.6 Semantic similarity between sentences

Semantic similarity is an active research area which is increased explosively, it tries to calculate how close are words, concepts, sentences and documents. Similarity among two words is a measure of the likeliness of their meaning, computed based on the properties of concepts and their relationships in taxonomy or ontology.

Given two sentences, the semantic similarity between two sentences is the measurement that determines how similar the meaning of two sentences is. The higher the score, the more similar the meaning of the two sentences. Several measures have been defined to quantify the semantic similarity between any two senses [28]:

Path-based Similarity

Return a score denoting how similar two word senses(C1,C2) are, based on length of the shortest Path connecting (distance between C1 and C2) the senses in the is-a (hypernym/hypnoym) taxonomy. The distance can then be used to find the semantic similarity between any two synsets C1 and C2 by applying the formula:

$$sim_{path} = \frac{1}{distance(C1,C2)}$$

The score is in the range 0 to 1, except in those cases where a path cannot be found, in this case None is returned.

Leacock Chordorow(LCH) similarity

Leacock Chordorow defines the similarity measure which is an extended version of Path-based similarity as it incorporates the depth of the taxonomy. Therefore, it is the negative log of the shortest path (min-path) between two concepts (synset-1 and synset-2) divided by twice the total depth of the taxonomy (D). The LCH similarity scores are between 0 and 3.689.

$$sim_{lhc} = -log\frac{minpath(C1,C2)}{2*D}$$

WUP similarity

Wu and Palmer extend this similarity by incorporating the depth of Least Common Subsumer(LCS). LCS is the most specific concept that two concepts share as ancestor (closest common ancestor of C1 and C2 from the root node), in this measure, the similarity is twice the depth of two concepts LCS divided by the sum of the depths of the individual concepts:

$$sim_{wup} = \frac{2*depth(lcs(C1,C2))}{depth(C1)+depth(C2)}$$

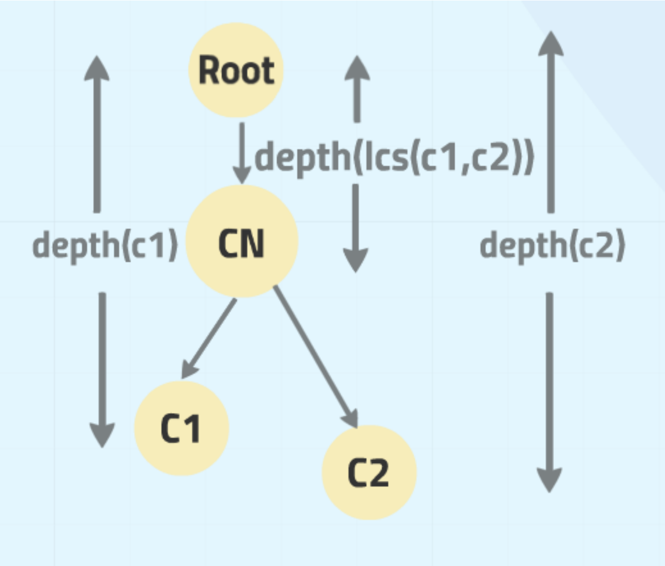


Figure 3.2: WUP similarity measure

Wu-Palmer similarity calculation gives a similarity score from 0 to 1.

In order to show the difference between the above algorithms that measure the semantic similarity between two text units , we apply these three algorithms to a set of words to compare their semantic similarity:

After the output of each similarity measures, we can see that the WUP similarity gives the best results compared to LCH and path_similarity, this is why we will opt for **WUP_similarity** to calculate the semantic similarity in our proposed solution.

3.7 BM25(Best Matching) OKAPI Algorithm

DEFINITION

(word 1,word 2)	(disease,sickness)	(disease,cancer)	(building,skyscraper)	(car,vehicle)
path_sim	0.5	0.25	0.5	0.2
LCH_sim	0.79	0.60	0.79	0.54
WUP_sim	0.95	0.86	0.93	0.8

Table 3.3: Semantic similarity score between words using three different algorithms

In information retrieval, Okapi BM25 stands for "Best Match 25", is a ranking function used by search engines to estimate the relevance of documents to a given search query. It is based on the probabilistic retrieval framework developed in the 1970s and 1980s by Stephen E.Robertson, Karen Spärck Jones and others [29].

More specifically, BM25 is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of their proximity within the document [29].

Given a query Q, containing keywords q1,...,qn, the BM25 score of a document D is:

$$Score(Q,D) = \sum_{i}^{n} IDF(q_i) \frac{f(q_i,D) * (k1+1)}{f(q_i,D) + k1 * (1 - b + b * \frac{|D|}{avd_{dl}})}$$

Where:

- q_i**:** is the i^{th} query term.
- $f(q_i$,D): is q_i's term frequency in the document D(the number of times term qi occurs in document D).
- $|D|$: is the length of the document D(number of words in document D).
- d_{avg} : is the average document length in the text collection from which documents are drawn.
- b and k1: are hyper-parameters for BM25 :
  - k1 : This parameter controls non-linear term frequency normalization (saturation). It controls how quickly an increase in term frequency results in term-frequency saturation. The default value is 1,2. Lower values result in quicker saturation and higher values in slower saturation [30].

  - b: This parameter controls how much effect field-length normalization should have. A value of 0.0 disables normalization completely, and a value of 1.0 normalizes fully. The default is 0.75 [30].

- $IDF(q_i)$: is the IDF (inverse document frequency) weight of the query term q_i. It is usually computed as:

$$IDF(q_i) = log\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}$$

  where :

  - N: is the total number of documents in the collection.
  - $n(q_i)$: is the number of documents containing term (q_i).

Consider we have a query Q with n words such as q1,q2,q3,... , qn, for each query word, term frequency and inverse document frequency will be calculated using the above formula and the score will be generated using BM25. Once we get the scores for each document, we can arrange the documents in an increasing order of BM25 score.

3.8 CORD-19 dataset description

On March 16, 2020, the Allen Institute for AI (AI2), in collaboration with partners at The White House Office of Science and Technology Policy (OSTP), the National Library of Medicine (NLM), the Chan Zuckerburg Initiative (CZI), Microsoft Research, and Kaggle, coordinated by Georgetown University's Center for Security and Emerging Technology (CSET), released the first version of CORD-19 In response to the COVID-19[31].

The picture 3.3 below shows that publications increased during and following the SARS and MERS epidemics(2002/2003), but the number of papers published in the early months of 2020 exploded in response to the COVID-19 epidemic.

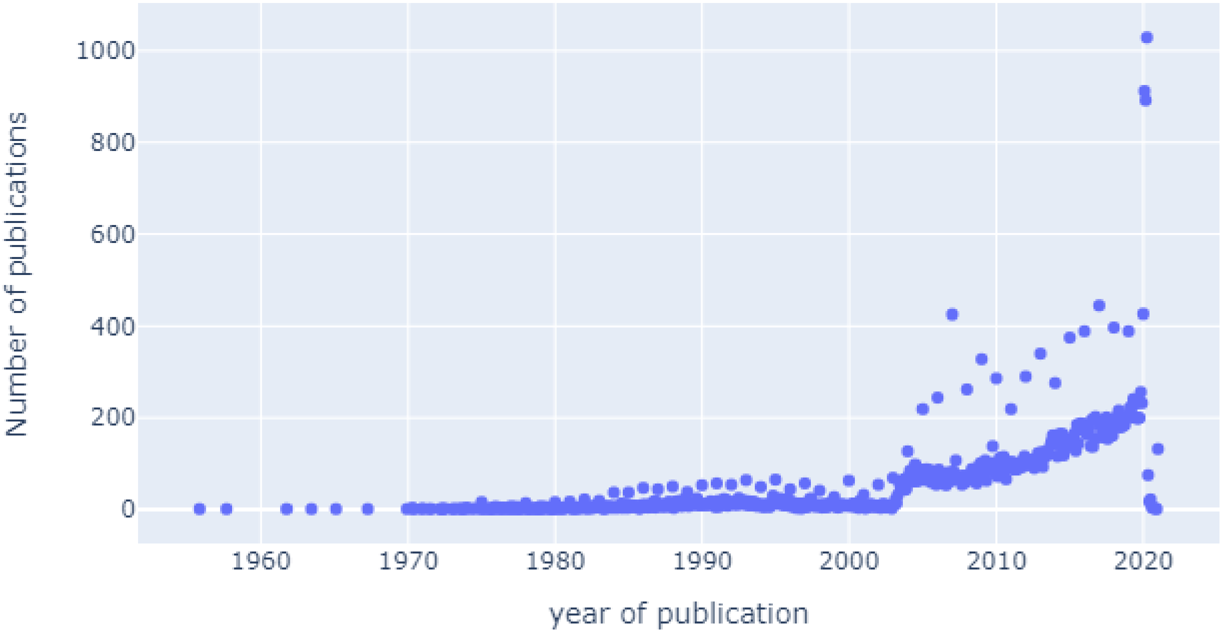


Figure 3.3: Number of publications over time

Most of articles can be found at the PMC (PubMed Central) and Elsevier. The other sources are the "Chan Zuckerberg Initiative","WHO", "bioRxiv" and the "medRxiv", as shown in the picture 3.4 below:

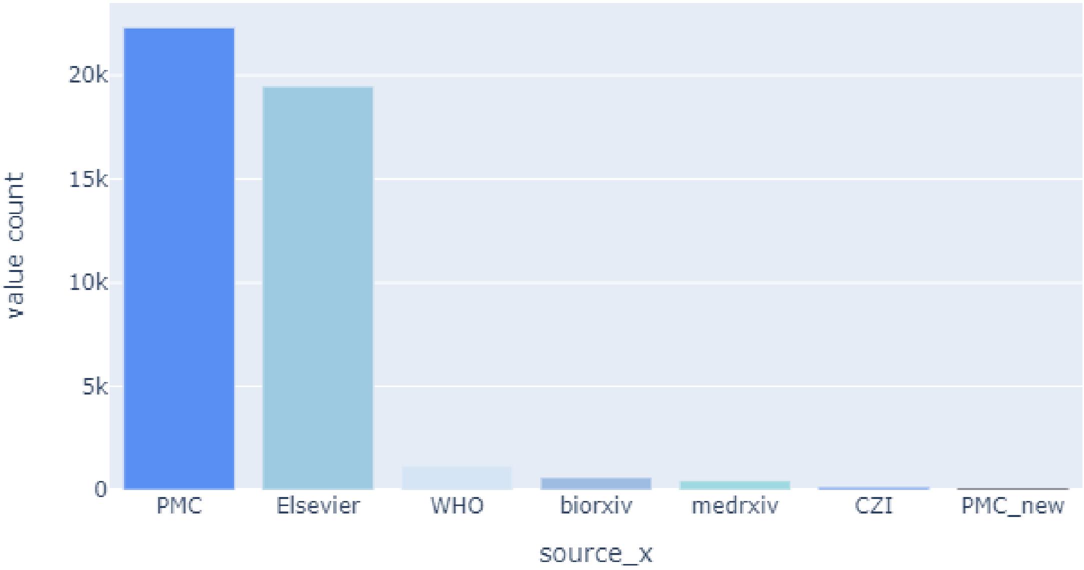


Figure 3.4: Source of articles

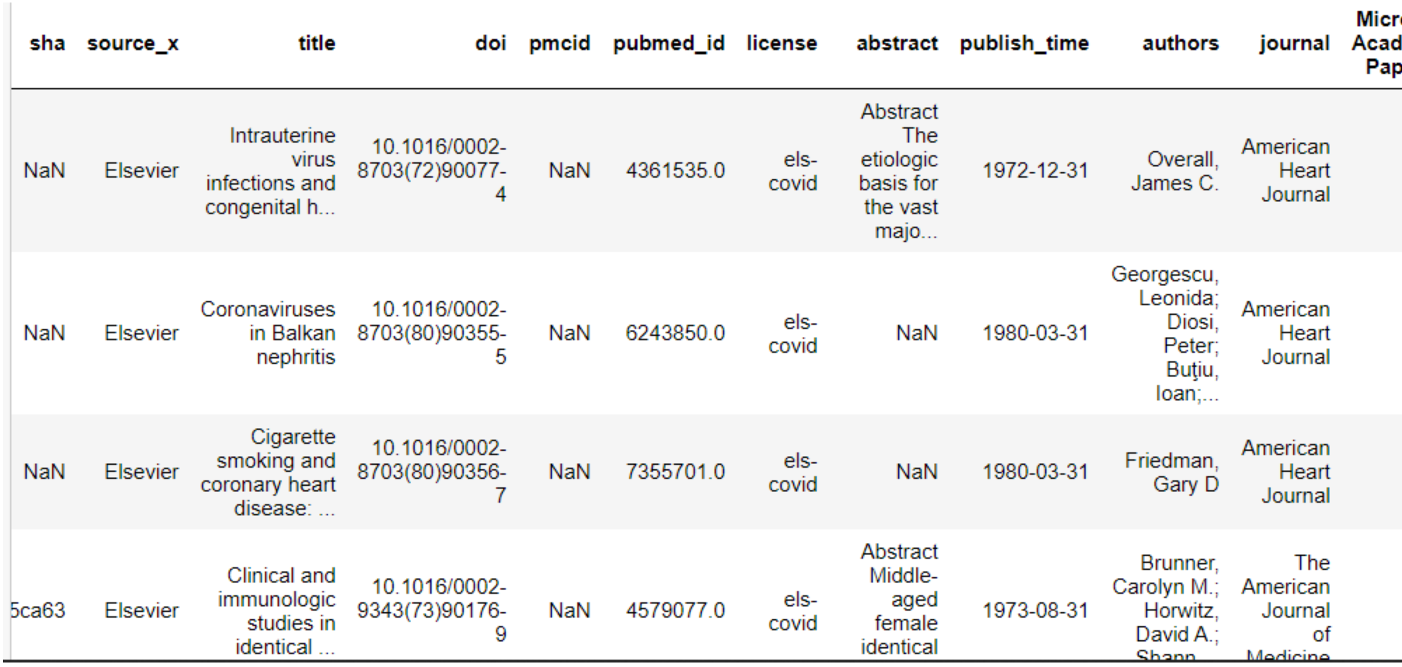
CORD-19 contains a meta-data file and four folders of articles separated by their source. Every paper is represented by a unique JSON object.

folder name	number of JSON files
biorxiv_medrxiv	885
comm_use_subset	9118
custom_license	19956
noncomm_use_subset	2353

Table 3.4: Content of the dataset

The meta-data file is a CSV file that contains 15 columns which provides details about articles. Here is a short overview of the most important columns:

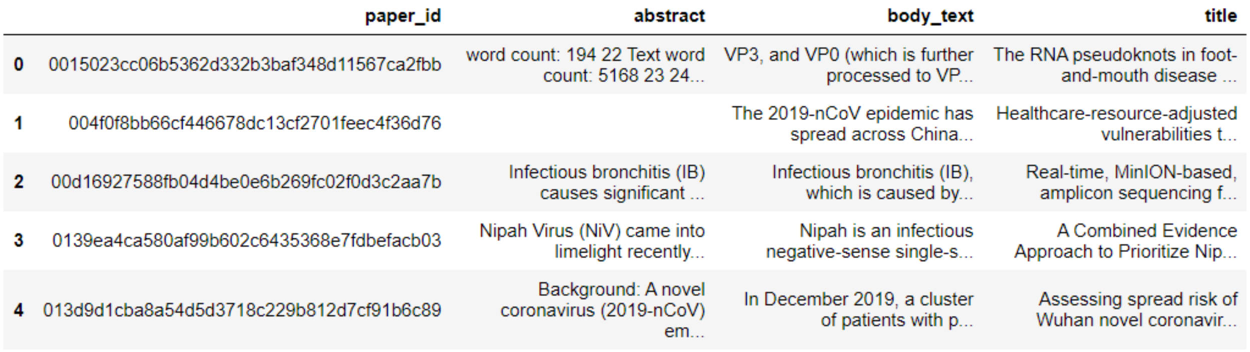
- **Column "sha":** represents the paper identifier(ID)
- **Column "source_x":** contains the article editor.
- **Column "title":** contains the title of the publication.
- **Column "doi":** represents a "digital object identifier" and seems to be an online link to the publication.
- **Column "pubmed_id":** contains the ID of the pubmed.
- **Column "license":** contains the license under which article has been published.
- **Column "abstract":** contains the abstracts of the publications.
- **Column "publish_time":** contains the date of the publication.
- **Column "authors":** contains a list of all authors of the publication.
- **Column "journal":** contains the journal in which the article has been published.



sha	source_x	title	doi	pmcid	pubmed_id	license	abstract	publish_time	authors	journal	Micr Acad Pap
NaN	Elsevier	Intrauterine virus infections and congenital h...	10.1016/0002-8703(72)90077-4	NaN	4361535.0	els-covid	Abstract The etiologic basis for the vast majo...	1972-12-31	Overall, James C.	American Heart Journal	
NaN	Elsevier	Coronaviruses in Balkan nephritis	10.1016/0002-8703(80)90355-5	NaN	6243850.0	els-covid	NaN	1980-03-31	Georgescu, Leonida; Diosi, Peter; Buţiu, Ioan;...	American Heart Journal	
NaN	Elsevier	Cigarette smoking and coronary heart disease: ...	10.1016/0002-8703(80)90356-7	NaN	7355701.0	els-covid	NaN	1980-03-31	Friedman, Gary D	American Heart Journal	
5ca63	Elsevier	Clinical and immunologic studies in identical ...	10.1016/0002-9343(73)90176-9	NaN	4579077.0	els-covid	Abstract Middle-aged female identical	1973-08-31	Brunner, Carolyn M.; Horwitz, David A.; Shann	The American Journal of Medicine	

Figure 3.5: meta-data file content

The JSON file details every paper using four columns: title, paper_id, abstract and the body_text. The content of JSON files is extracted and transformed, as follows:

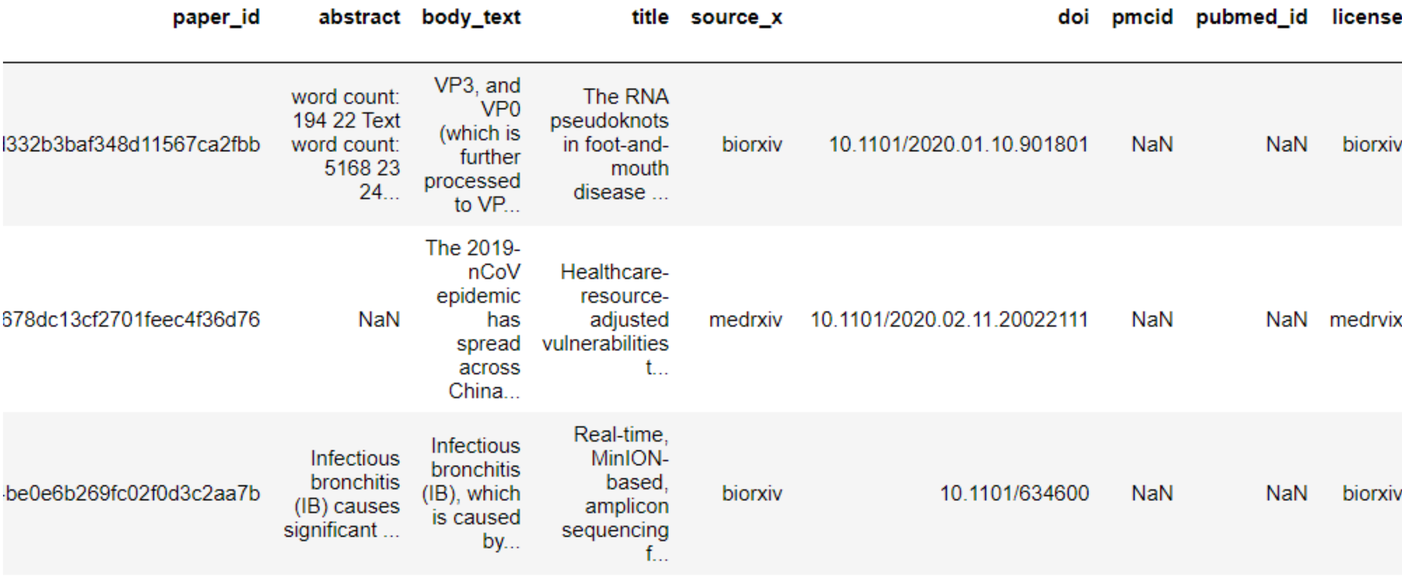


	paper_id	abstract	body_text	title
0	0015023cc06b5362d332b3baf348d11567ca2fbb	word count: 194 22 Text word count: 5168 23 24...	VP3, and VP0 (which is further processed to VP...	The RNA pseudoknots in foot-and-mouth disease ...
1	004f0f8bb66cf446678dc13cf2701feec4f36d76		The 2019-nCoV epidemic has spread across China...	Healthcare-resource-adjusted vulnerabilities t...
2	00d16927588fb04d4be0e6b269fc02f0d3c2aa7b	Infectious bronchitis (IB) causes significant ...	Infectious bronchitis (IB), which is caused by...	Real-time, MinION-based, amplicon sequencing f...
3	0139ea4ca580af99b602c6435368e7fdbefacb03	Nipah Virus (NiV) came into limelight recently...	Nipah is an infectious negative-sense single-s...	A Combined Evidence Approach to Prioritize Nip...
4	013d9d1cba8a54d5d3718c229b812d7cf91b6c89	Background: A novel coronavirus (2019-nCoV) em...	In December 2019, a cluster of patients with p...	Assessing spread risk of Wuhan novel coronavir...

Figure 3.6: JSON file content

The transformed JSON files are merged with their corresponding meta-data based on the paper ID to form a full and final dataset, as follows:

```
# append all json files to the corresponding metadata
full_df = papers\
    .merge(metadata.rename(columns={'sha':'paper_id'}).drop(['abstract','title'], axis=1),
           on='paper_id', how='left')

full_df.head()
```



paper_id	abstract	body_text	title	source_x	doi	pmcid	pubmed_id	license
l332b3baf348d11567ca2fbb	word count: 194 22 Text word count: 5168 23 24...	VP3, and VP0 (which is further processed to VP...	The RNA pseudoknots in foot-and-mouth disease ...	biorxiv	10.1101/2020.01.10.901801	NaN	NaN	biorxiv
678dc13cf2701feec4f36d76	NaN	The 2019-nCoV epidemic has spread across China...	Healthcare-resource-adjusted vulnerabilities t...	medrxiv	10.1101/2020.02.11.20022111	NaN	NaN	medrvix
-be0e6b269fc02f0d3c2aa7b	Infectious bronchitis (IB) causes significant ...	Infectious bronchitis (IB), which is caused by...	Real-time, MinION-based, amplicon sequencing f...	biorxiv	10.1101/634600	NaN	NaN	biorxiv

Figure 3.7: JSON files content with their corresponding meta-data

3.9 System architecture overview

Our proposed query based multi-document summarization system works in two steps, the output of the first step serves as an input for the second step:

- **FIRST STEP:** is a query based single-document summarization, each document in the corpus will be summarized extractively and saved temporarily.

- **SECOND STEP:** is an extractive generic multi-document summarization, the set of temporary summaries resulted from the first step will be used as an input to generate a single extractive summary.

The figure below shows how does the two steps interact:

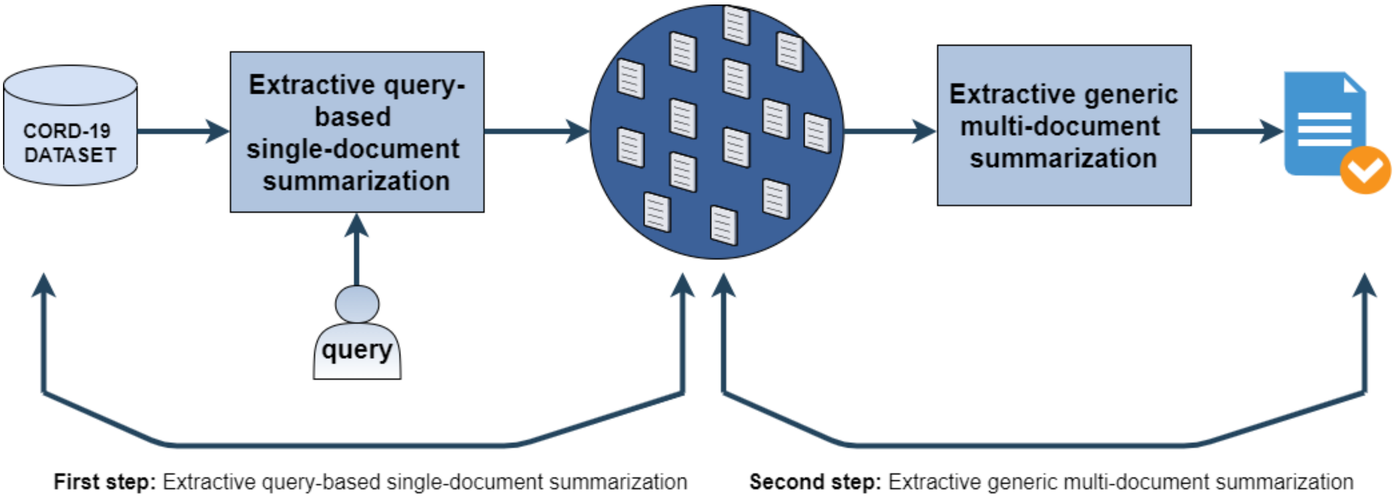

CORD-19 DATASET
Extractive query-based single-document summarization
query
Extractive generic multi-document summarization
First step: Extractive query-based single-document summarization
Second step: Extractive generic multi-document summarization

Figure 3.8: The proposed system architecture

3.9.1 FIRST STEP: Extractive query-based single-document summarization

This step is considered as query based single-document summarization, in which every document in the corpus will be processed individually and separately. In this phase, the model by applying an algorithm on the document assigns a score to each sentence in the document, then selects the top N sentences that are related to the input query or question to generate a summary.

A list of initial key questions that we want to get informations about them can be found in the table below. These key scientific questions are drawn from the NASEM's SCIED (National Academies of Sciences, Engineering, and Medicine's Standing Committee on Emerging Infectious Diseases and 21st Century Health Threats) research topics and the World Health Organization's for COVID-19.

There are a lot of questions regrouped into 8 tasks, for each task we'll give three examples of questions:

Task:	questions
What is known incubation, and environmental stability?	-Natural history of the virus and shedding of it from an infected - Immune response and immunity. - Role of the environment in transmission.
What do we know about COVID-19 risk factors?	- Smoking, pre-existing pulmonary disease. - Neonates and pregnant women. - Public health mitigation measures could be effective for control.
What do we know about vaccines and therapeutics?	- Prevalence of asymptomatic shedding and transmission. - Approaches to evaluate risk for enhanced disease after vaccination. - Effectiveness of drugs being developed and tried to treat sars patients
What do we know about virus genetics, origin, and evolution?	- Sustainable risk reduction strategies. - Evidence that livestock could be infected. - Animal host and any evidence of continued spill-over to humans
What has been published about medical care?	- Resources to support skilled nursing facilities and long term care facilities. - Guidance on the simple things people can do at home to take care of sick people and manage disease. - Oral medications that might potentially work.
What do we know about non-pharmaceutical interventions?	- Research on the economic impact of this or any pandemic. - Guidance on ways to scale up NPIs in a more coordinated way. - Research on why people fail to comply with public health advice.
What do we know about diagnostics and surveillance?	- Technology roadmap for diagnostics . - Efforts to increase capacity on existing diagnostic platforms . - Development of a point-of-care test.
What has been published about information sharing and inter-sectoral collaboration?	- Sharing response information among planners, providers . - Misunderstanding around containment and mitigation. - Mitigating threats to incarcerated people from COVID-19, assuring access to information, prevention, diagnosis.

Table 3.5: Some examples of questions related to COVID-19

The figure below shows the whole process of this phase, and it will be explained in detail in the following paragraphs:

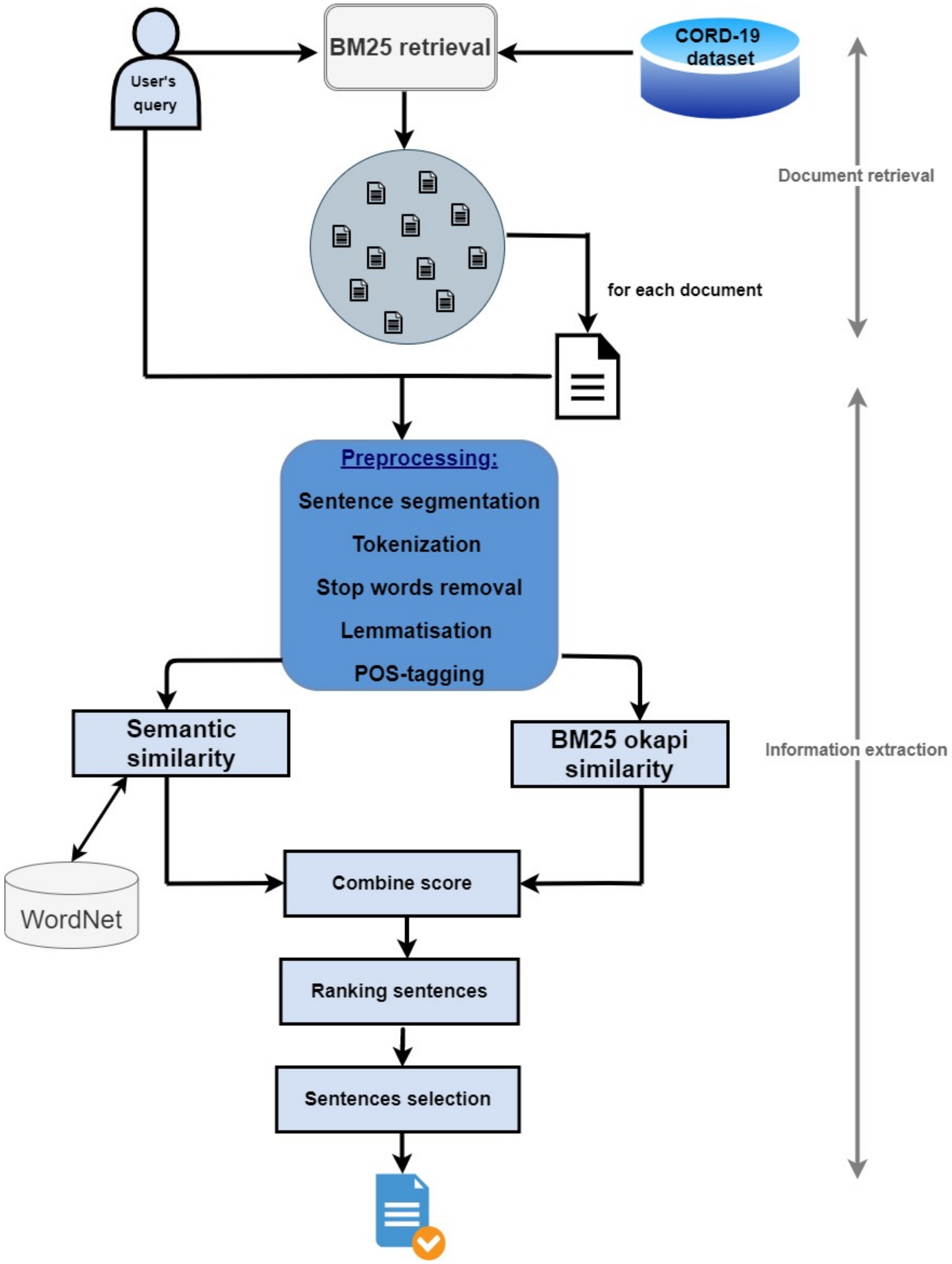


Figure 3.9: First step: query based single document summarization

Phase 1: Document retrieval

Document retrieval is the first phase in our proposed system because we are dealing with more than 50k papers, we need therefore to reduce the size of the corpus by using a document retrieval function.

The aim of the document retrieval is to find and select top matching articles for a query in the collection. To do so, BM25 OKAPI index is used as a similarity measure between the query and the set of documents. For a given query, it ranks the set of documents in the dataset by attributing a score to each document then retrieve documents that have score more that 0. The resulting set of documents is then used to retrieve the most relevant information.

Before further processing, text needs to be cleaned. Cleaning data generally refers to a series of related tasks meant to put all text on one level playing field:

- Remove empty and duplicated data especially from "body_text" column.
- Discard non-English articles.
- Identify the papers published after 2019 concerning precisely the novel COVID-19 disease: to find COVID-19 related articles, we have defined a list of key words, the article is considered COVID-19 related if, any of these fields (title, abstract and full text) has any of the key word: [’ncov’, ’covid19’, ’covid-19’, ’sars cov2’, ’sars cov-2’, ’sars-cov-2’, ’sars coronavirus 2’, ’2019-ncov’, ’2019 novel coronavirus’, ’2019-ncov sars’, ’cov-2’, ’cov2’, ’novel coronvirus’, ’coronavirus 2019-ncov’].
- Remove square brackets including numbers, corresponding to citations (e.g.’[6, 11]’).
- Remove punctuations such as "?", "!", ";" and special characters.
- Lowercase the text.

Then, the given corpus and the input query have undergone pre-processing.

Documents and query Preprocessing

Preprocessing is the process of preparing data by changing unstructured data to be structured data according to the needs. It’s an essential phase, if neglected or realized in a too simplistic manner, systems risk giving wrong results. Indexing engines or automatic summarization systems in particular are very sensitive to the amount of noise in a text. Preprocessing is applied exclusively to the body of each paper in the following way:

A) **Sentence Segmentation:** it is the process of dividing the text document into a sentences and converts a raw text document into a list of sentences[32]:

  - **Input text=**"Coronaviruses are a family of enveloped RNA viruses that cause diseases in animals and humans. Coronavirus infection in domestic animals has led to major economic loss worldwide."
  - **Output text=**["Coronaviruses are a family of enveloped RNA viruses that cause diseases in animals and humans","Coronavirus infection in domestic animals has led to major economic loss worldwide"].

B) Tokenization: identifies the word tokens from given sentence. Tokenization takes a sentence as an input and provides a list of tokens as output[32]. Following example shows the input and output of tokenization process:

  - **Input text**="Coronaviruses are a family of enveloped RNA viruses that cause diseases in animals and humans."
  - **Output text**=[’Coronaviruses’, ’are’, ,’a’, ’family’, ’of’, ’enveloped’, ’RNA’,’ viruses’, ’that’, ’cause’,’ diseases’, ’in’, ’animals’,’ and’, ’humans’].

C) Stop-Word removal: Stop words like a, an, at are removed as they do not convey in a document. Prepositions, pronouns, articles, connectives etc. are also considered as stop words. Since they carry very little information about the contents of a document.

  - **Input text**="Coronaviruses are a family of enveloped RNA viruses that cause diseases in animals and humans."
  - **Output text**=[’Coronaviruses’, ’family’, ’enveloped’, ’RNA’,’ viruses’, ’cause’,’ diseases’, ’animals’, ’humans’].

D) Lemmatization: is related to stemming, differing in that lemmatization is able to capture canonical forms based on a word's lemma [33]. For example, stemming the word "better" would fail to return its citation form, however, lemmatization would result in the following: better → good.

E) POS-tagging: Part-of-speech tagging is a process of marking up or tagging a word in a text or a sentence or a corpus corresponding to a particular part-of-speech based on its definition and context . Part-of-speech consists of automatically associating words in a text with corresponding grammatical information (verbs, adjectives, nouns, adverb) by its relationship with the adjacent word. **N** stands for noun phrase, **V** stands for verbs, **A** for adjectives and **AD** for adverbs[32].

– **Input** =[[Coronaviruses], [are], [family] ,[enveloped],[RNA], [viruses],[cause] ,[diseases]].

– **Output** =((Coronaviruses),n), ((family),n), ((enveloped),a), ((RNA),n), ((viruses),n) ((cause),v),((diseases),n).

Phase 2: Information extraction

1- Sentences score:

Every sentence is provided with an importance score and it reflects the measure of goodness for that sentence and how relevant is a sentence to the query. These scores can be made used for the ordering of the sentences and to pick out those which has more importance.

After preprocessing the documents and the input query, the sentences in the original document are scored by calculating semantic similarity score and BM25 score between each sentence within the document and the input query. We calculate the score of a sentence by using the following measures:

A- Semantic similarity using WordNet

```
Algorithm :Semantic_ similarity_score
Input: sentence 1, sentence 2
Output: similarity_score
Begin
    # Tokenization,lemmatization and tag
    1: sentence1 ← pos_tag(lemma(word_tokenize(sentence1)))
    2: sentence2 ← pos_tag(lemma(word_tokenize(sentence2)))
     # Get the synsets for the tagged words
     3:synsets1 ← [Synset(tagged_word) for tagged_word in sentence 1]
    4: synsets2 ← [Synset(tagged_word) for tagged_word in sentence2]
    5: score←0, count ← 0
     # For each word in the first sentence
    6: for synset in synsets1 :
     # Get the similarity value of the most similar word in the other sentence
    7: best_score ← max([synset.WUP_similarity(ss) for ss in synsets2])
     # Check that the similarity could have been computed
    8: If best_score is not None:
     9: similarity_score ←similarity_score + best_score
     10: count ←count+ 1
    12: End if
    13: End for
    15: return similarity_score/count
End
```

Table 3.6: semantic similarity algorithm

Explanation

The above algorithm shows how to calculate the semantic similarity between two sentences s1 and s2. The inputs are two English sentences, and the output is a semantic similarity score. The process is as follow:

First, each sentence is tokenized into set of words, then these words are lemmatized in the aim to remove inflectional and derivationally related forms of a word to a common base, then apply part of speech tagging to the words in the sentences, this is essential to pick the correct meaning of the word in WordNet. The resulting words will be used to find the equivalent synsets (Synsets 1, Synsets 2) respectively for (Words of sentence 1, Words of sentence 2) in WordNet, obtain and save synsets for each sentence. Then loop to process all words pairs by comparing all synsets in sentence 1 to all synsets in sentence 2 (for each word in the first sentence, get the similarity value of the most similar word in the other sentence)using wup_similarity measure. At the end return the similarity score.

B- BM25 OKAPI algorithm

Usually, BM25 is used to retrieve relevant documents based on a user input query, but in our approach, BM25 will be used to select important sentences related to input query. The algorithm BM25 is the same with only one difference, instead of providing a set of documents as input, a set of sentences are provided.

```
Algorithm :compute_bm25_score
Input: Corpus, Query
Output: bm25_score
Begin
    #build bag of words for corpus first
    corpus_features =Bag_of_word(normalized_corpus,normalized_Query)
    # get document length and average document length of the corpus (avgdl)
    corpus_doc_lengths =  doc_lengths(normalized_corpus)
    avg_doc_length=doc_avg(normalized_corpus)
    # compute inverse document frequencies of all the terms in a corpus of documents by using
    its Bag of Words features
    term_idfs =compute_corpus_term_idf(corpus_feature,normalized_corpus)
    # compute numerator expression in BM25 equation
    numerator_coeff = corpus_features * (k1 + 1)
    numerator = term_idfs * numerator_coeff
    # compute denominator expression in BM25 equation
    denominator_coeff = k1 * (1 - b + (b * (corpus_doc_lengths / avg_doc_length)))
    denominator = corpus_features + denominator_coeff
    # compute the BM25 score combining the above equations
    bm25_scores = numerator / denominator
    return bm25_scores
End
```

Table 3.7: BM25 OKAPI algorithm

Explanation

To compute BM25 scores for sentences, we must go through several steps:

- Build a function to get Bag of Words–based features for corpus.
- Build a function to get inverse document frequency(IDF) values for terms in corpus by using its Bag of Words features, which will contain the term frequencies, and then convert them to IDF.

- Compute the length and the average length of corpus sentences.
- Build a function for computing BM25 scores between a query sentence and corpus sentences, we first compute the numerator expression in the BM25 mathematical equation we specified earlier and then compute the denominator expression.
- Finally, we divide the numerator by the denominator to get the BM25 scores for all the corpus sentences. Then we sort them in descending order and return the top n relevant sentences with the highest BM25 score.

2- Combine scores and get the average score:

Each sentence is ranked based on the average of combination of the semantic similarity score and BM25 score. The top N sentences are then selected to form the summary.

Score(sentence_i, query)= (semantic_similarity_score(sentence_i,query) + compute_bm25_similarity(sentence_i,query))/2

3- Sentence selection:

Each sentence is given an importance score and this acts as a goodness measure for the sentence. The score can be used to order sentences and picks most important sentences. After associating score for each sentence. The generated summary is done by ranking them in descending order. Finally, top N highest scoring sentences are selected and combined together into a single generated summary.

A Walk-Through Example

To see the difference between a summary generated with WordNet and a summary generated with WordNet and BM25, the following example is provided:

Take as an example an article from the dataset titled **"The landscape of lung bronchoalveolar immune cells in COVID-19 revealed by single-cell RNA sequencing"**, and which can be found here: https://www.medrxiv.org/content/10.1101/2020.02.23.20026690v. The content of this article is visualized as follows:

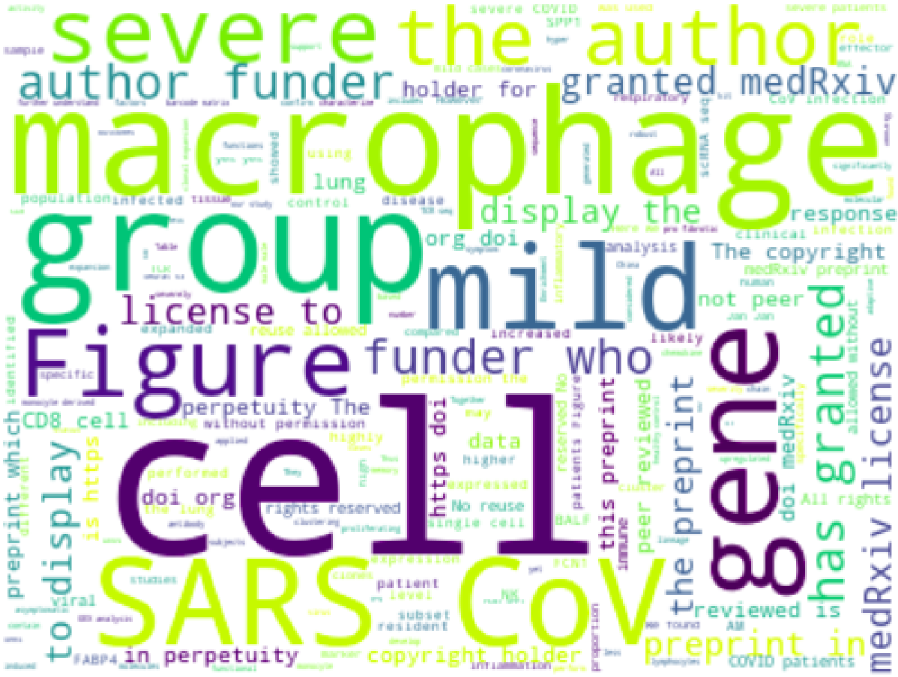
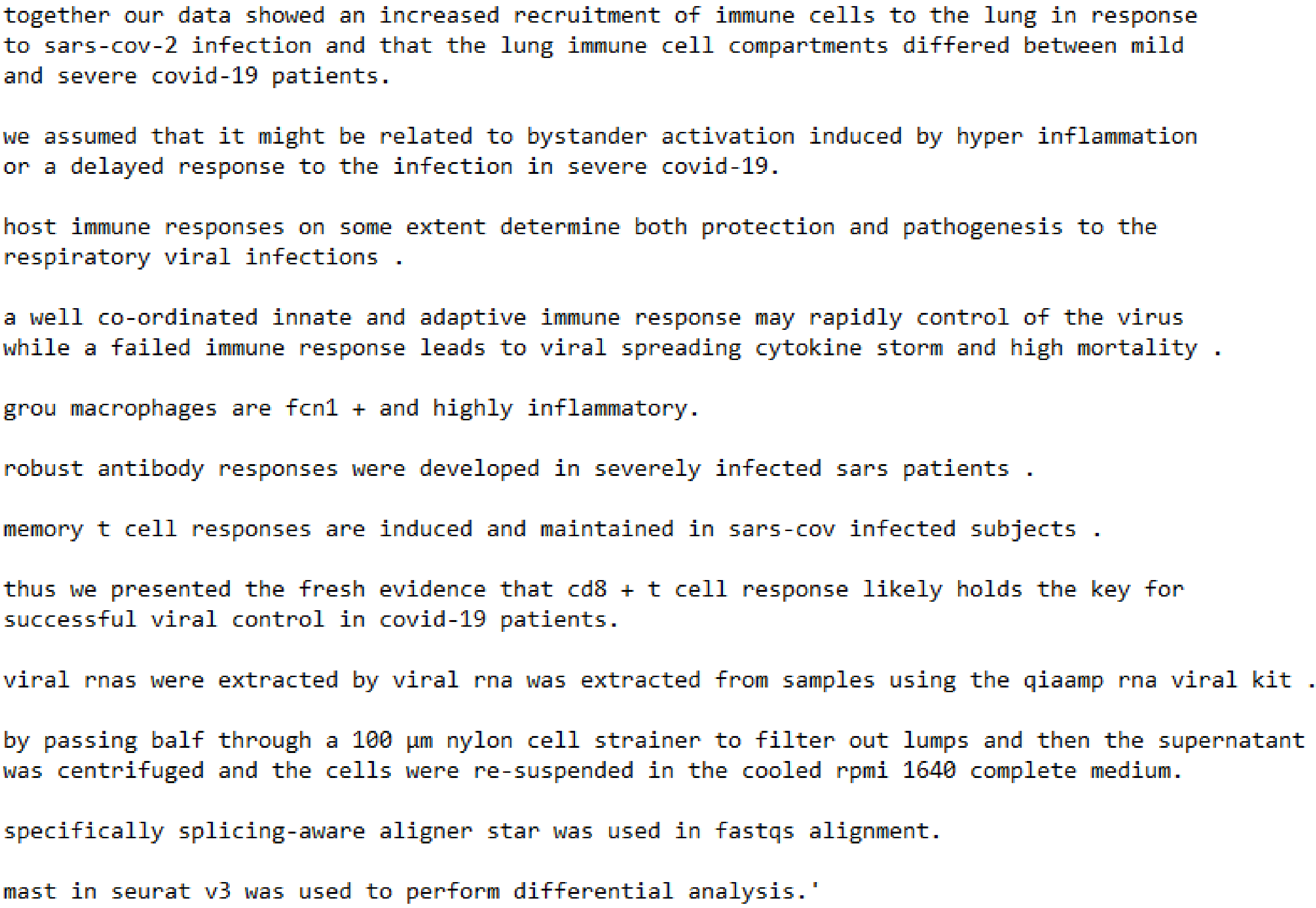


Figure 3.10: Visual representation of the article

From this article, we want to extract the most related sentences to the following question: **"what is the immune system response to COVID-19 ?"**

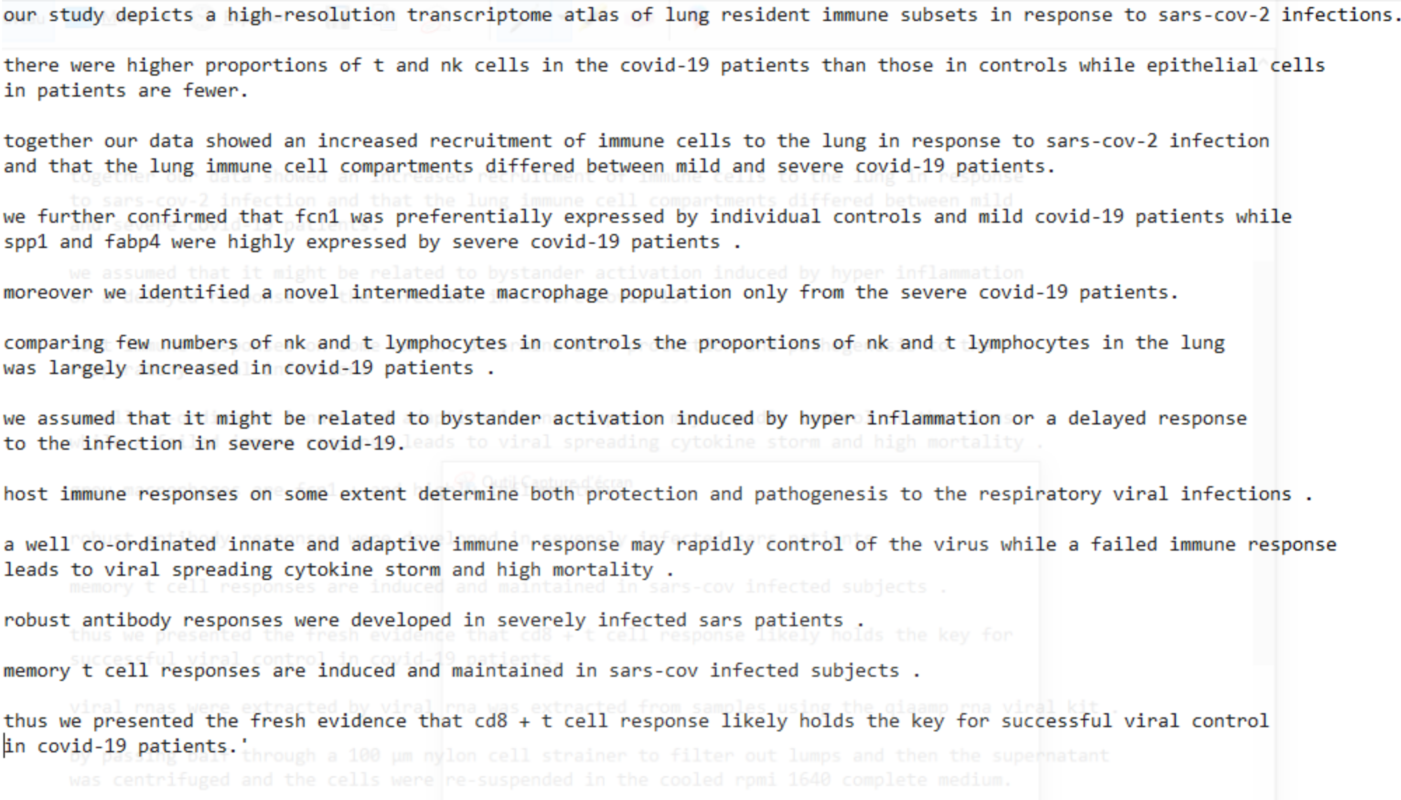
The summary generated(top 12 sentences) using WordNet algorithm is:



```
together our data showed an increased recruitment of immune cells to the lung in response
to sars-cov-2 infection and that the lung immune cell compartments differed between mild
and severe covid-19 patients.

we assumed that it might be related to bystander activation induced by hyper inflammation
or a delayed response to the infection in severe covid-19.

host immune responses on some extent determine both protection and pathogenesis to the
respiratory viral infections .

a well co-ordinated innate and adaptive immune response may rapidly control of the virus
while a failed immune response leads to viral spreading cytokine storm and high mortality .

grou macrophages are fcn1 + and highly inflammatory.

robust antibody responses were developed in severely infected sars patients .

memory t cell responses are induced and maintained in sars-cov infected subjects .

thus we presented the fresh evidence that cd8 + t cell response likely holds the key for
successful viral control in covid-19 patients.

viral rnas were extracted by viral rna was extracted from samples using the qiaamp rna viral kit .

by passing balf through a 100 μm nylon cell strainer to filter out lumps and then the supernatant
was centrifuged and the cells were re-suspended in the cooled rpmi 1640 complete medium.

specifically splicing-aware aligner star was used in fastqs alignment.

mast in seurat v3 was used to perform differential analysis.'
```

Figure 3.11: Summary generated with WordNet algorithm

And the summary generated(top 12 sentences) after combining WordNet with BM25 algorithm is:



```
our study depicts a high-resolution transcriptome atlas of lung resident immune subsets in response to sars-cov-2 infections.

there were higher proportions of t and nk cells in the covid-19 patients than those in controls while epithelial cells
in patients are fewer.

together our data showed an increased recruitment of immune cells to the lung in response to sars-cov-2 infection
and that the lung immune cell compartments differed between mild and severe covid-19 patients.

we further confirmed that fcn1 was preferentially expressed by individual controls and mild covid-19 patients while
spp1 and fabp4 were highly expressed by severe covid-19 patients .

moreover we identified a novel intermediate macrophage population only from the severe covid-19 patients.

comparing few numbers of nk and t lymphocytes in controls the proportions of nk and t lymphocytes in the lung
was largely increased in covid-19 patients .

we assumed that it might be related to bystander activation induced by hyper inflammation or a delayed response
to the infection in severe covid-19.

host immune responses on some extent determine both protection and pathogenesis to the respiratory viral infections .

a well co-ordinated innate and adaptive immune response may rapidly control of the virus while a failed immune response
leads to viral spreading cytokine storm and high mortality .

robust antibody responses were developed in severely infected sars patients .

memory t cell responses are induced and maintained in sars-cov infected subjects .

thus we presented the fresh evidence that cd8 + t cell response likely holds the key for successful viral control
in covid-19 patients.'
```

Figure 3.12: Summary generated using WordNet and BM25 algorithms

Comparing these two summaries, we notice that both of them have selected 7 relevant sentences in common from 12 sentences and 5 different sentences. The five different sentences of each approach are:

wordnet	wordnet+BM25
1- grou macrophages are fcn1 + and highly inflammatory. 2- viral rnas were extracted by viral rna was extracted from samples using the qiaamp rna viral kit. 3- by passing balf through a 100 um nylon cell strainer to filter out lumps and then the supernatant was centrifuged and the cells were re-suspended in the cooled rpmi 1640 complete medium. 4- mast in seurat v3 was used to perform differential analysis. 5- specifically splicing-aware aligner star was used in fastqs alignment.	1- our study depicts a high-resolution transcriptome atlas of lung resident immune subsets in in response to sars-cov-2 infections. 2- there were higher proportions of t and nk cells in the covid-19 patients than those in controls while epithelial cells in patients are fewer 3- moreover we identified a novel intermediate macrophage population only from the severe covid-19. 4- comparing few numbers of nk and t lymphocytes in controls the proportions of nk and t lymphocytes in the lung was largely increased in covid-19 patients 5- memory t cell responses are induced and maintained in sars-cov infected subjects.

Table 3.8: Different sentences resulted from each approach

Reading these five sentences of each approach and compare them, we found that the sentences selected using only WordNet are not really important and could be ignored because they do not provide any meaningful informations to the question, and this is the opposite of the sentences chosen using WordNet and BM25 together, they are almost relevant and provide useful informations related to the question.

3.9.2 SECOND STEP: Extractive generic multi-document summarization

The second step is considered as a continuation of the first step, during this phase the set of summaries resulted from the first step are used as an input to the second step to generate one final summary.

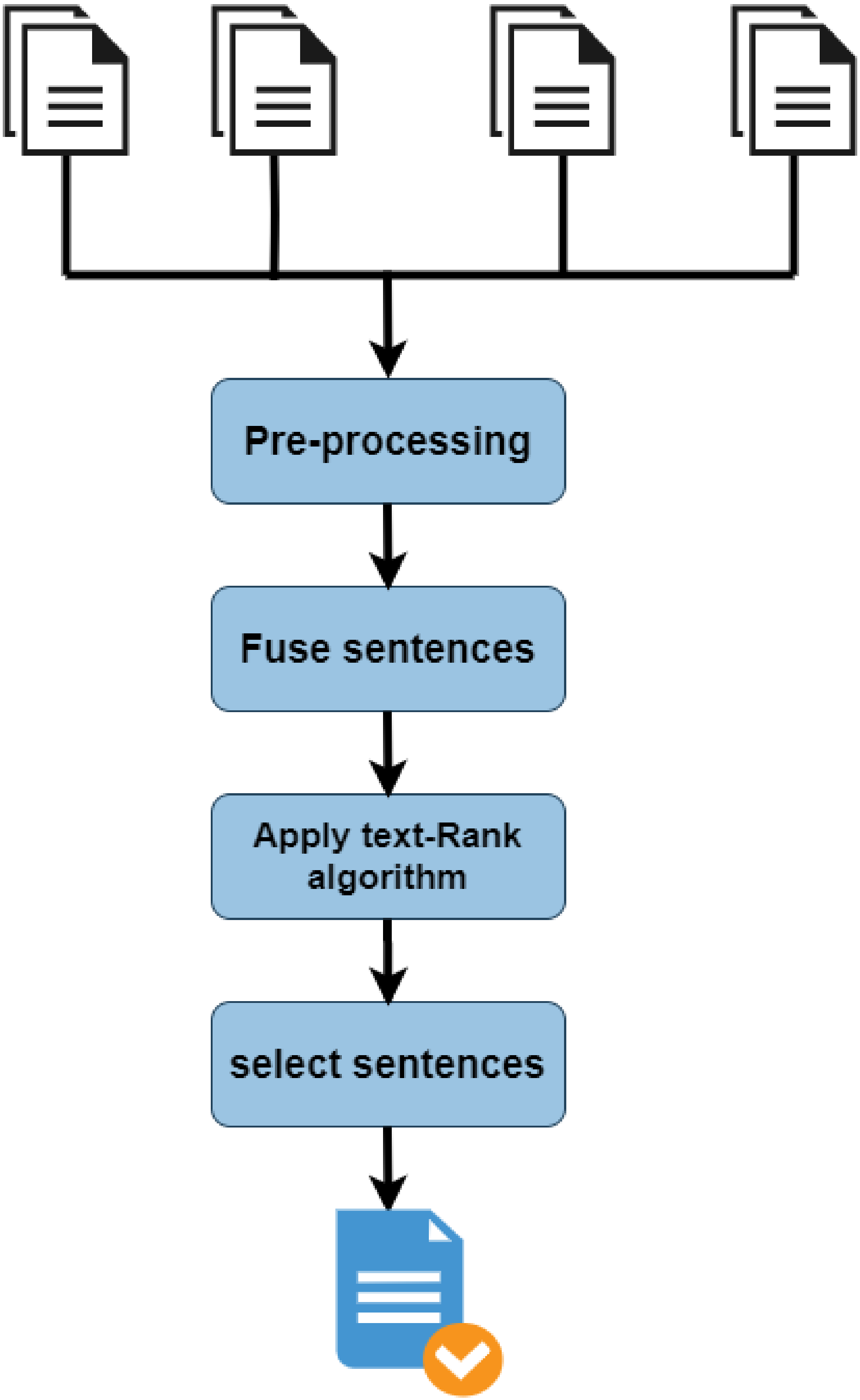


Figure 3.13: Second step process

0. **Preprocessing:** is very similar to the preprocessing done in the first step without part-of-speech (POS) tagging.

0. Concatenate sentences: the sentences are then combined together in one single document.

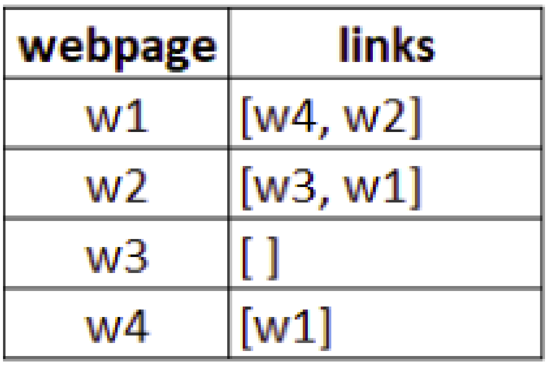
0. Apply TextRank algorithm:

DEFINITION: TextRank is an unsupervised algorithm can be used to obtain the most relevant sentences in text and also to find keywords in a document.

The algorithm applies a variation of PageRank over a graph constructed. This produces a ranking of the elements in the graph, the most important elements are the ones that better describe the text [34].

TextRank finds its roots associated with Google's PageRank (by Larry Page) used for ranking web-pages for on-line search results, but before unfolding TextRank, we must understand PageRank and the intuition behind it by explaining a simple example:

Suppose we have 4 web pages: w1, w2, w3, and w4. These pages contain links pointing to one another. Some pages might have no link – these are called dangling pages.



webpage	links
w1	[w4, w2]
w2	[w3, w1]
w3	[ ]
w4	[w1]

Figure 3.14: Example of application of Page-Rank algorithm

- Web page w1 has links directing to w2 and w4.

- w2 has links for w3 and w1.

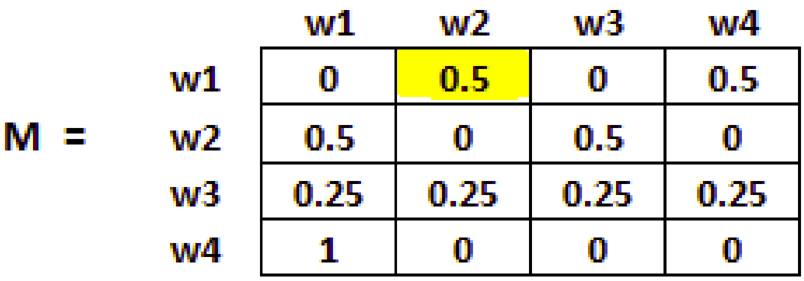
- w4 has links only for the web page w1.

- w3 has no links and hence it will be called a dangling page.

In order to rank these pages, we would have to compute a score called the PageRank score. This score is the probability of a user visiting that page.

To capture the probabilities of users navigating from one page to another, we will create a square matrix M, having n rows and n columns, where n is the number of web pages.

Each element of this matrix denotes the probability of a user transitioning from one web page to another. For example, the highlighted cell below contains the probability of transition from w1 to w2. The initialization of the probabilities is explained in the steps below:



$$M = \begin{array}{c|c|c|c|c|} & w1 & w2 & w3 & w4 \\ \hline w1 & 0 & 0.5 & 0 & 0.5 \\ \hline w2 & 0.5 & 0 & 0.5 & 0 \\ \hline w3 & 0.25 & 0.25 & 0.25 & 0.25 \\ \hline w4 & 1 & 0 & 0 & 0 \\ \hline \end{array}$$

Figure 3.15: Probability matrix

1- Probability of going from page i to j, i.e., M[i][j], is initialized with 1/(number of unique links in web page wi).

2- If there is no link between the page i and j, then the probability will be initialized with 0.

3- If a user has landed on a dangling page, then it is assumed that he is equally likely to transition to any page. Hence, M[i][j] will be initialized with 1/(number of web pages).

4- Finally, the values in this matrix will be updated in an iterative fashion to arrive at the web page rankings.

From PageRank to TextRank:

The concept of TextRank inspired from PageRank with some modifications where:

- WebPages are replaced with text sentences.

- The probability of going from page A to page B is equal to the similarity of two sentences.

- The similarity scores are stored in a square matrix, similar to the matrix M used for PageRank.

- TextRank graph is undirected. Meaning that all edge are bidirectional.

- The similarity matrix for index [A, B] is filled with similarity values between sentences A and B rather than 1/total_links from Page B to A.

- There are a lot of alternative to calculate the similarity measure like TF-IDF,cosine similarity,BM25...

The figure below shows the flow of TextRank algorithm:

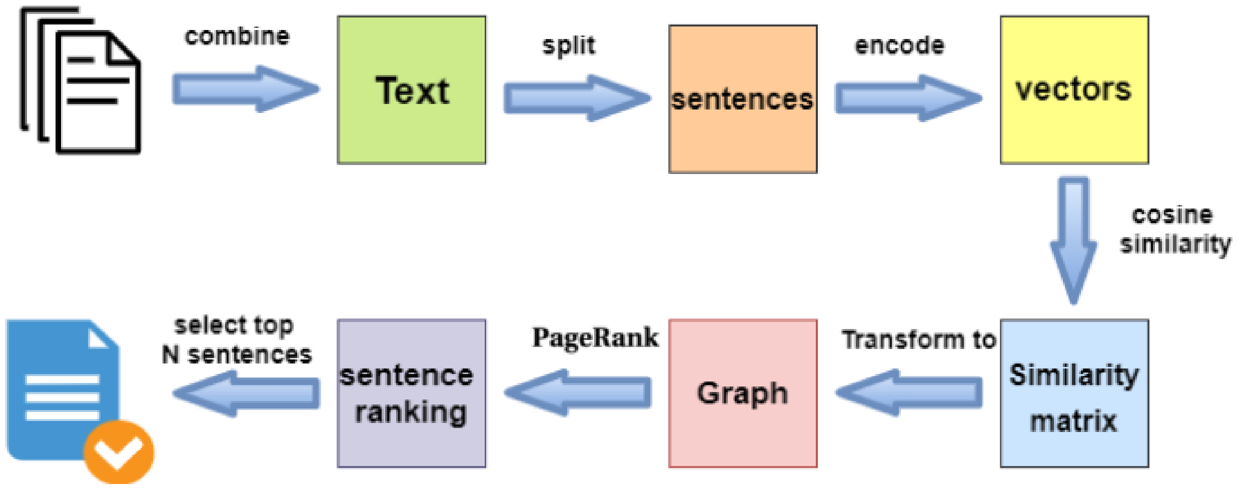


Figure 3.16: Flow of TextRank algorithm

The table below explains how does TextRank algorithm work:

```
Algorithm :TextRank
Input:document_text
Output: Top N ranked_sentences
Begin
    # break the text into individual sentences
    sentences ← sentences_tokenize(document_text)
    # represent each sentence with a vector
    For sent in sentences :
        sentence_vectors_list.add( vector_representation_sentences(sent))
    End For
    # create similarity matrix and calculate similarities between the sentences using the
    cosine similarity approach
    For i in length(sentences):
        For j in length(sentences):
            similarity_matrix[i][j] ← cosine_similarity(sentence_vectors_list[i], sentence_vectors_list[j])
        End For
    End for
    #convert the similarity matrix into a graph and apply PageRank algorithm.
    score ← PageRank(graph(similarity_matrix))
    #extract the top N sentences based on their rankings
    ranked_sentences ← sorted(((scores[i],s) for i,s in enumerate(sentences)), reverse=True)
    For i in range(N):
        print(ranked_sentences[i][1])
    End For
End
```

Table 3.9: TextRank algorithm

Explanation

- We start by combining all text contained in documents in one single text.
- Then split the text into sentences and apply the preprocessing steps.
- Tokenize each sentence into a collection of words and represent it with a vector.

– Calculate the similarity between sentences vectors using the cosine similarity approach and store it in a matrix.

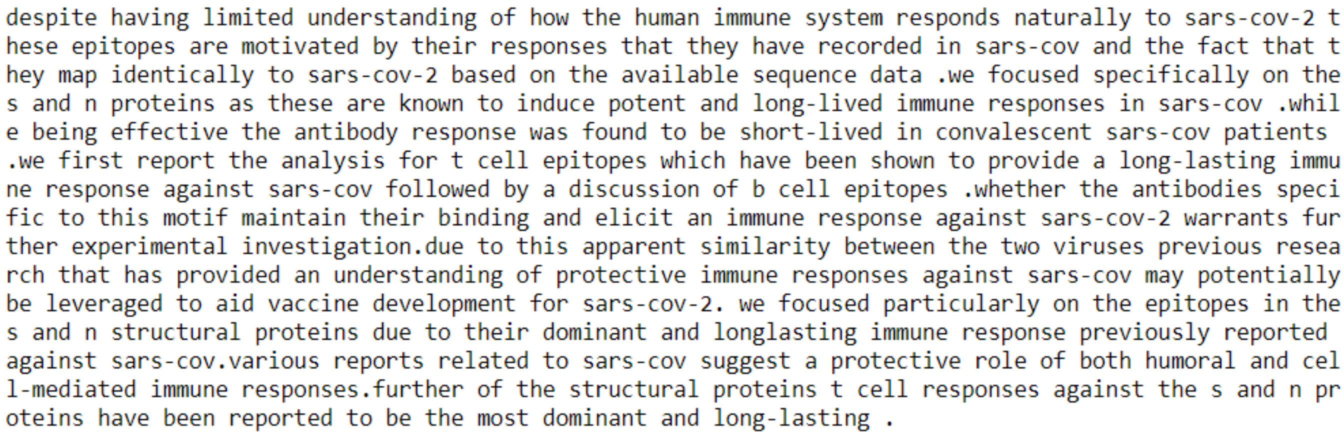
– Transform the cosine similarity matrix into a graph by representing the sentences as nodes and the similarity between two sentences as a edge.

– Apply the PageRank algorithm on the graph to calculate scores for each sentence.

– Rank sentences based on their score.

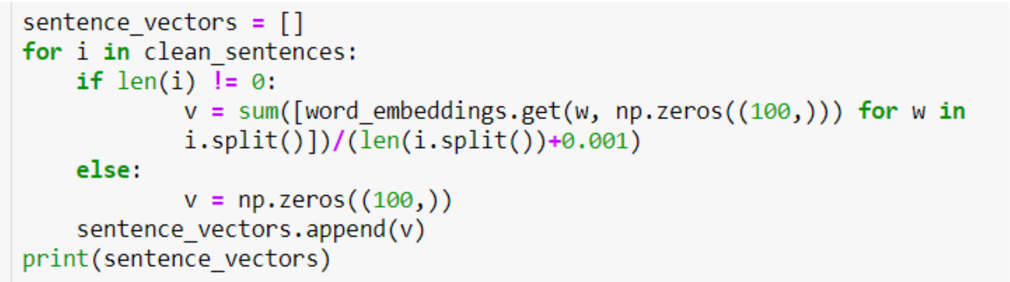
– Finally, extract the top N sentences based on their rank to generate a summary .

The application of TextRank is illustrated with an example. Given as example, the text below:



```
despite having limited understanding of how the human immune system responds naturally to sars-cov-2 t
hese epitopes are motivated by their responses that they have recorded in sars-cov and the fact that t
hey map identically to sars-cov-2 based on the available sequence data .we focused specifically on the
s and n proteins as these are known to induce potent and long-lived immune responses in sars-cov .whil
e being effective the antibody response was found to be short-lived in convalescent sars-cov patients
.we first report the analysis for t cell epitopes which have been shown to provide a long-lasting immu
ne response against sars-cov followed by a discussion of b cell epitopes .whether the antibodies speci
fic to this motif maintain their binding and elicit an immune response against sars-cov-2 warrants fur
ther experimental investigation.due to this apparent similarity between the two viruses previous resea
rch that has provided an understanding of protective immune responses against sars-cov may potentially
be leveraged to aid vaccine development for sars-cov-2. we focused particularly on the epitopes in the
s and n structural proteins due to their dominant and longlasting immune response previously reported
against sars-cov.various reports related to sars-cov suggest a protective role of both humoral and cel
l-mediated immune responses.further of the structural proteins t cell responses against the s and n pr
oteins have been reported to be the most dominant and long-lasting .
```

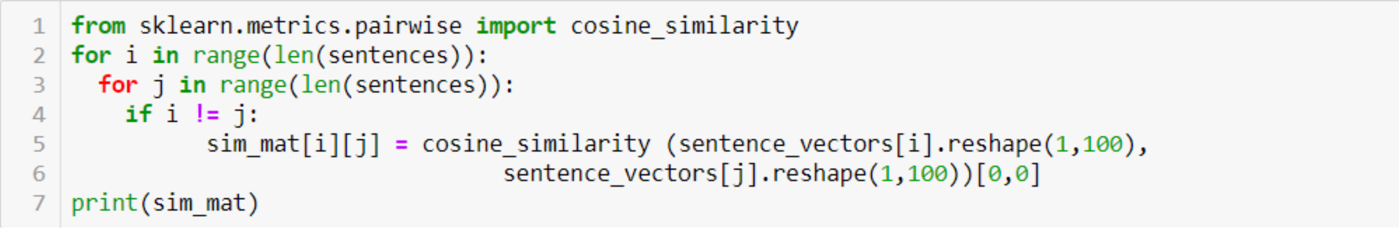
After tokenizing the text into sentences, we represent each sentence as a vector using these instructions:



```
sentence_vectors = []
for i in clean_sentences:
    if len(i) != 0:
            v = sum([word_embeddings.get(w, np.zeros((100,))) for w in
            i.split()])/(len(i.split())+0.001)
    else:
            v = np.zeros((100,))
    sentence_vectors.append(v)
print(sentence_vectors)
```

Figure 3.17: vector representation

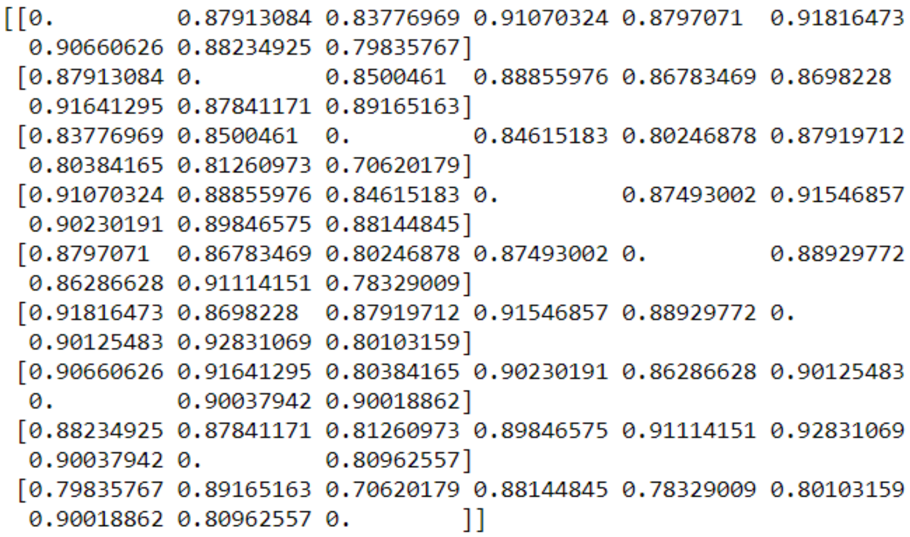
Then use these vectors to construct the cosine similarity matrix by writing these instructions:



```
from sklearn.metrics.pairwise import cosine_similarity
for i in range(len(sentences)):
  for j in range(len(sentences)):
    if i != j:
          sim_mat[i][j] = cosine_similarity (sentence_vectors[i].reshape(1,100),
                                sentence_vectors[j].reshape(1,100))[0,0]
print(sim_mat)
```

Figure 3.18: Build the similarity matrix

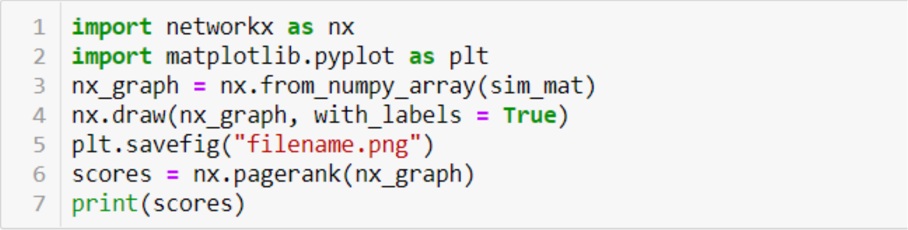
The constructed matrix is shown in the following figure:



```
[[0.         0.87913084 0.83776969 0.91070324 0.8797071  0.91816473
  0.90660626 0.88234925 0.79835767]
 [0.87913084 0.         0.8500461  0.88855976 0.86783469 0.8698228
  0.91641295 0.87841171 0.89165163]
 [0.83776969 0.8500461  0.         0.84615183 0.80246878 0.87919712
  0.80384165 0.81260973 0.70620179]
 [0.91070324 0.88855976 0.84615183 0.         0.87493002 0.91546857
  0.90230191 0.89846575 0.88144845]
 [0.8797071  0.86783469 0.80246878 0.87493002 0.         0.88929772
  0.86286628 0.91114151 0.78329009]
 [0.91816473 0.8698228  0.87919712 0.91546857 0.88929772 0.
  0.90125483 0.92831069 0.80103159]
 [0.90660626 0.91641295 0.80384165 0.90230191 0.86286628 0.90125483
  0.         0.90037942 0.90018862]
 [0.88234925 0.87841171 0.81260973 0.89846575 0.91114151 0.92831069
  0.90037942 0.         0.80962557]
 [0.79835767 0.89165163 0.70620179 0.88144845 0.78329009 0.80103159
  0.90018862 0.80962557 0.        ]]
```

Figure 3.19: cosine similarity matrix

Transform this matrix into graph and apply the PageRank algorithm using these instruction:



```
import networkx as nx
import matplotlib.pyplot as plt
nx_graph = nx.from_numpy_array(sim_mat)
nx.draw(nx_graph, with_labels = True)
plt.savefig("filename.png")
scores = nx.pagerank(nx_graph)
print(scores)
```

Figure 3.20: The instructions needed to generate the graph

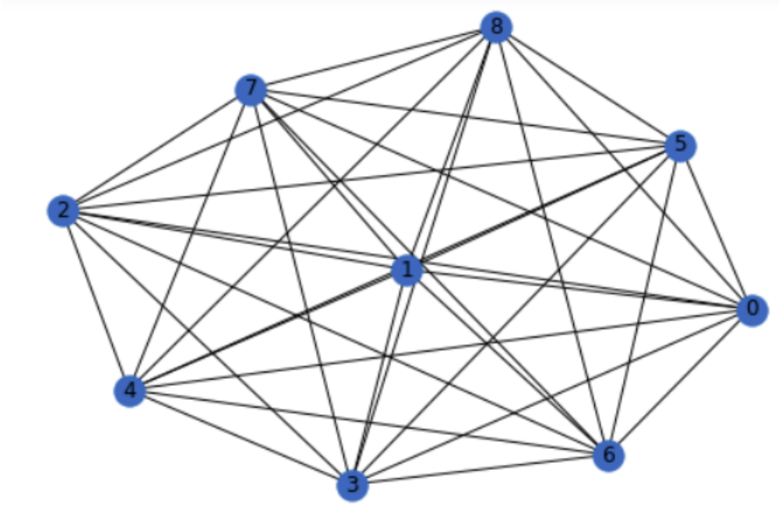
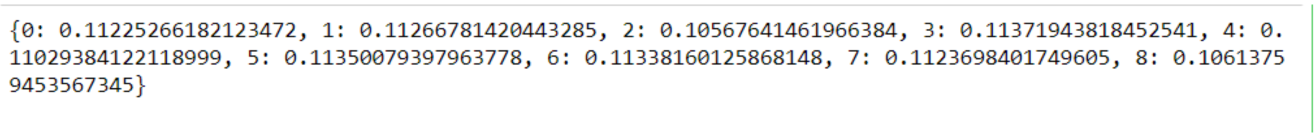


Figure 3.21: graph generated by PageRank

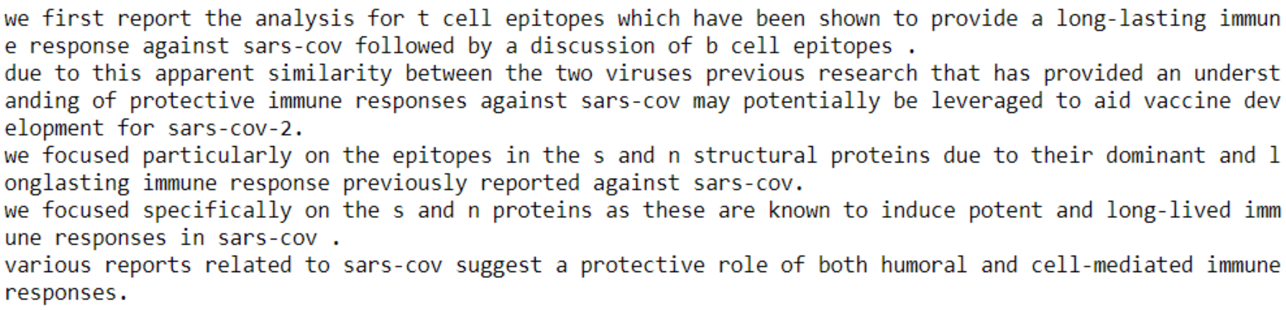
The algorithm associates a score to each sentence to each sentence a score, then selects let's say the 5 top sentences:



```
{0: 0.11225266182123472, 1: 0.11266781420443285, 2: 0.10567641461966384, 3: 0.11371943818452541, 4: 0.
11029384122118999, 5: 0.11350079397963778, 6: 0.11338160125868148, 7: 0.1123698401749605, 8: 0.1061375
9453567345}
```

Figure 3.22: sentences score

Finally, the generated summary is shown in the following figure :



```
we first report the analysis for t cell epitopes which have been shown to provide a long-lasting immun
e response against sars-cov followed by a discussion of b cell epitopes .
due to this apparent similarity between the two viruses previous research that has provided an underst
anding of protective immune responses against sars-cov may potentially be leveraged to aid vaccine dev
elopment for sars-cov-2.
we focused particularly on the epitopes in the s and n structural proteins due to their dominant and l
onglasting immune response previously reported against sars-cov.
we focused specifically on the s and n proteins as these are known to induce potent and long-lived imm
une responses in sars-cov .
various reports related to sars-cov suggest a protective role of both humoral and cell-mediated immune
responses.
```

Figure 3.23: summary generated using TextRank

3.10 Conclusion

In this chapter, the proposed solution have been detailed. First, we have started by citing some related works, we have defined the data set used in our work , then we have explained the approaches used to achieve this work. Finally, we have dived deeper into the architecture of our proposed solution to extend our objective, this solution consists of dividing the query based multi-document summarization into two step, query based single-document summarization and generic multi-document summarization. The evaluation and the tools used to implement this solution will be presented in the next chapter.

Chapter 4

Evaluation and discussion

4.1 Introduction

This chapter speaks about the different libraries used, the environment and the programming language chosen to implement our model. It presents then the evaluation of our model, analyzes and discusses the results of the proposed system and then displays the Django application.

4.2 Programming language

The proposed query based multi-document summarization system was implemented using python. Python is an object-oriented, high-level programming language with integrated dynamic semantics primarily for web and application development. It is extremely attractive in the field of Rapid Application Development because it offers dynamic typing and dynamic binding options.

Python is relatively simple, so it's easy to learn since it requires a unique syntax that focuses on readability. Developers can read and translate Python code much easier than other languages. In turn, this reduces the cost of program maintenance and development because it allows teams to work collaboratively without significant language and experience barriers [35].

Additionally, Python supports the use of modules and packages, which means that programs can be designed in a modular style and code can be reused across a variety of projects.

Once you have developed a module or package you need, it can be scaled for use in other projects, and it's easy to import or export these modules. Most important packages that helped us to implement our solution are:

- **NLTK**: stands for Natural Language Toolkit. This toolkit is one of the most powerful NLP libraries which contains packages to make machines understand human language and reply to it with an appropriate response. Tokenization, Stemming, Lemmatization, Punctuation, Character count, word count are some of these packages [36].

Task	package
Sentence segmentation	nltk.sent_tokenize
Word tokenization	nltk.tokenize.word_tokenize
stop word removal	nltk.corpus.stopwords.words('english')
Stemming	nltk.stem.porter.PorterStemmer
Lemmatization	nltk.stem.WordNetLemmatizer
POS-tagging	nltk.pos_tag

Table 4.1: NLTK packages used

- **Pandas**: is a Python package providing fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language [1].

- **Numpy**: which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. It is ideal for operations related to linear algebra, Fourier transformations, or crunching of random numbers [37].

- **Matplotlib**: is a comprehensive library for creating static, animated, and interactive visualizations in Python,it generates plots, histograms, power spectra, bar charts,

[1] https://pypi.org/project/pandas/

error-charts, scatter-plots, etc., with just a few lines of code [38].

- **Seaborn:** is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics [2]

HTML and CSS

HTML stands for Hypertext Markup Language. It allows the user to define the meaning and structure of web content like sections, paragraphs, headings, links, and block-quotes for web pages and applications.

HTML is not a programming language, meaning it doesn't have the ability to create dynamic functionality. Instead, it makes it possible to organize and format documents, similarly to Microsoft Word [39].

CSS stands for Cascading Style Sheets. It is a simple design language intended to simplify the process of making web pages presentable. Using CSS, make possible to control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects[40].

4.3 Development platform

Anaconda

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, which was founded by Peter Wang and Travis Oliphant in 2012. It is also known as Anaconda Distribution or Anaconda Individual Edition. It comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package

[2] https://seaborn.pydata.org/

and virtual environment manager which makes it easy to install/update packages and create/load environments. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI). Applications available by default in Navigator are: JupyterLab,Jupyter Notebook, Spyder, Glue, Orange, RStudio, Visual Studio Code [3].

Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more [4].

Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your application without needing to reinvent the wheel. It's free and open source. Its main goals are simplicity, flexibility, reliability, and scalability. Django's template language is designed to feel comfortable and easy-to-learn to those used to working with HTML, like designers and front-end developers. But it is also flexible and highly extensible, allowing developers to augment the template language as needed [5]. [41].

4.4 Automatic Summary Evaluation using ROUGE

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is essentially a set of metrics for evaluating automatic summarization of texts as well as machine translation. It works by comparing an automatically produced summary or translation against a set of reference summaries (typically human-produced)[42]. The ROUGE system calculates several measures for evaluation of system-generated summaries human-generated

[3] https://www.anaconda.com/products/individual

[4] https://jupyter.org

[5] https://www.djangoproject.com/

summaries also known as model summaries. These measures show how well the peer summaries correlate to the model summaries and the measures are based on many different methods.

Precision, Recall, and F-measure in the Context of rouge

To evaluate how accurate our machine generated summaries are, we compute the Precision, Recall and F-measure for any metric:

RECALL

In the context of ROUGE, RECALL means how much of the reference summary is the system summary recovering or capturing? If we are just considering the individual words, it can be computed as:

$$Recall = \frac{number\_of\_overlappig\_words}{total\_words\_in\_reference\_summary}$$

Precision

In the context of ROUGE, precision refers that how much candidate summary words are relevant. Formula to calculate RECALL:

$$Precision = \frac{number\_of\_overlappig\_words}{total\_words\_in\_system\_summary}$$

F-measure

F-measure is the harmonic mean of precision and recall:

$$F - meaure = 2 * \frac{precision * recall}{precision + recall}$$

For better understanding how to calculate those three metrics, an example given in [42] will be explained. Supposed that :

The summary generated by the system is: **"the tiny little cat was found under the big funny bed. "**

The summary generated by human is:**"the cat was under the bed."**

The number of overlapping words between the system summary and reference summary is six (the, cat, was, the, bed, under), total words in reference summary is six and total words in system summary is ten, then :

Recall = 6/6 =1 , Precision = 6/11 =0.55 , and F-measure = 0.17

ROUGE includes 5 evaluation metrics which are:

- **ROUGE-N**: It measures the overlap of n-grams between automatically generated summary and reference summary. In n-grams, value of N can vary from 1 to n. Mostly used n-gram metrics are uni-gram and bi-gram.
  - **ROUGE-1:** refers to overlap of unigrams between the system summary and reference summary.
  - **ROUGE-2**: refers to the overlap of bi-grams between the system and reference summaries.
- **ROUGE-L**: 'L' stands for Longest Common Subsequence(LCS). It computes the Longest Common Subsequence between reference summary and candidate summary. Each sentence in a summary is considered as a sequence of words. Two summaries which have longer common sequence of words are more similar to each other.
- **ROUGE-W**: 'W' here stands for Weighted Longest Common Subsequence. It is a variant of the ROUGE-L measure which takes into consideration the fact that some matches are consecutive in nature and hence they should be given a higher weight.
- **ROUGE-S:** measures the skip bi-gram co-occurrences in reference summary and candidate summary. Order of bi-grams is important. The skip-bi-grams of the above example are: the cat, the was, the under, the the, the bed, cat was, cat under, cat the, cat bed, was under, was the, was bed, under the, under bed, the bed.

A Simple Example

Let's consider an example to illustrate score calculation for some of the above mentioned metrics. If we want to compute the ROUGE-2 precision and recall scores from the example above:

System summary bi-grams are: the cat, cat was, was found, found under, under the, the bed.

Reference summary bi-grams are: the cat, cat was, was under, under the, the bed.

The ROUGE-2 recall is as follows: $rouge2_{recall}$ = 4/5 =0.8.
The system summary has recovered 4 bi-grams out of 5 bi-grams from the reference summary which is good.

The ROUGE-2 precision is as follows: $rouge2_{precison}$ = 4/6 =0.67.
The precision here tells that out of all the system summary bi-grams, there is a 67% overlap with the reference summary.

4.5 Evaluation of the first step

The evaluation of our summarization system is performed on ten documents, each one of them is evaluated using two different approaches against humans reference summaries.

The table 4.2 below shows the evaluation score for each document using just WordNet approach, and the table 4.3 shows the evaluation score for each document using a combination of WordNet and BM25 okapi:

Doc	**WordNet**											
Metric	**Rouge-1(%)**			**Rouge-2(%)**			**Rouge-L(%)**			**Rouge-W(%)**		
	P	R	F	P	R	F	P	R	F	P	R	F
D1	44.44	44.86	44.65	26.17	26.42	26.29	39.10	39.41	39.25	27.76	11.00	5.76
D2	51.96	51.46	51.71	45.54	45.10	45.32	50.56	50.15	50.36	43.52	17.06	24.51
D3	54.29	55.88	55.07	35.58	36.63	36.10	43.81	44.88	44.34	33.35	13.61	19.33
D4	28.57	29.41	28.99	2.88	2.97	2.93	21.93	22.47	22.20	10.75	4.39	6.23
D5	70.75	70.09	70.42	59.05	58.49	58.77	64.82	64.31	64.56	53.89	20.97	30.19
D6	69.02	56.11	61.90	42.46	34.50	38.07	45.88	38.61	41.93	29.26	7.17	11.52
D7	42.03	38.15	40.00	8.54	7.75	8.13	21.33	19.68	20.47	8.62	2.36	3.70
D8	40.47	55.50	46.81	13.76	18.89	15.92	23.28	30.29	26.32	10.80	5.05	6.88
D9	61.39	68.00	64.53	41.23	45.68	43.34	52.20	56.84	54.42	36.40	12.68	18.81
D10	50.12	56.55	53.14	23.76	26.82	25.20	28.55	31.57	29.99	17.50	6.09	9.03

Table 4.2: Evaluation score for each document using WordNet algorithm

Doc	**WordNet+BM25**											
Metric	**Rouge-1(%)**			**Rouge-2(%)**			**Rouge-L(%)**			**Rouge-W(%)**		
	P	R	F	P	R	F	P	R	F	P	R	F
D1	83.33	83.33	83.33	78.50	78.50	78.50	85.11	85.11	85.11	80.07	31.39	45.10
D2	69.61	68.27	68.93	59.41	58.25	58.82	64.26	63.23	63.74	53.22	20.62	29.72
D3	77.14	76.42	76.78	69.23	68.57	68.90	77.22	76.62	76.92	69.89	27.24	39.20
D4	92.38	91.51	91.94	88.46	87.62	88.04	93.61	92.87	93.24	89.46	34.87	50.18
D5	71.70	69.72	70.70	65.71	63.89	64.79	74.12	72.42	73.26	67.99	25.87	37.48
D6	86.50	62.53	72.59	72.62	52.44	60.90	73.69	56.23	63.78	51.29	10.92	18.01
D7	59.89	66.67	63.10	241.32	46.01	43.54	47.77	52.23	49.90	33.08	11.57	17.14
D8	50.17	50.68	50.42	24.83	25.08	24.96	34.71	35.01	34.86	17.91	5.80	8.76
D9	63.06	70.06	66.37	41.78	46.44	43.99	52.46	57.28	54.76	36.34	12.71	18.83
D10	47.90	48.62	48.26	23.51	23.87	23.69	30.13	30.51	30.32	18.30	5.61	8.59

Table 4.3: Evaluation score for each document using WordNet + BM25 algorithm

The table below represents the average RECALL, average precision, and average F-score of our system summaries generated by ROUGE package using wordnet:

Metric	average-precision(%)	average-recall(%)	average-F-measure(%)
Rouge-1	58.08	59.56	58.09
Rouge-2	36.39	38.72	37.31
Rouge-L	38.41	39.27	38.52
Rouge-W	24.06	7.84	11.7

Table 4.4: Average recall, precision, F-score values of our system using WordNet

The table below represents the average RECALL, average precision, and average F-score of our system summaries generated by ROUGE package using wordnet and BM25 OKAPI:

Metric	average-precision(%)	average-recall(%)	average-F-measure(%)
Rouge-1	69.69	66.72	67.68
Rouge-2	54.16	51.29	52.28
Rouge-L	58.19	56.19	56.90
Rouge-W	45.18	13.18	20.52

Table 4.5: Average recall, precision, F-score values of our system using WordNet + BM25

4.5.1 Comparison of RECALL

We compare the average Recall values of our summaries generated by the system using WordNet with average Recall values of our summaries generated by system using WordNet and BM25 OKAPI. The Table below gives the comparison for ROUGE- 1, 2, L, W metrics:

metric	**average-recall(WordNet)(%)**	**average-recall(WordNet+BM25)(%)**
Rouge-1	59.56	66.72
Rouge-2	38.72	51.29
Rouge-l	39.27	56.19
Rouge-w	7.84	13.18

Table 4.6: Comparison of average RECALL of our system

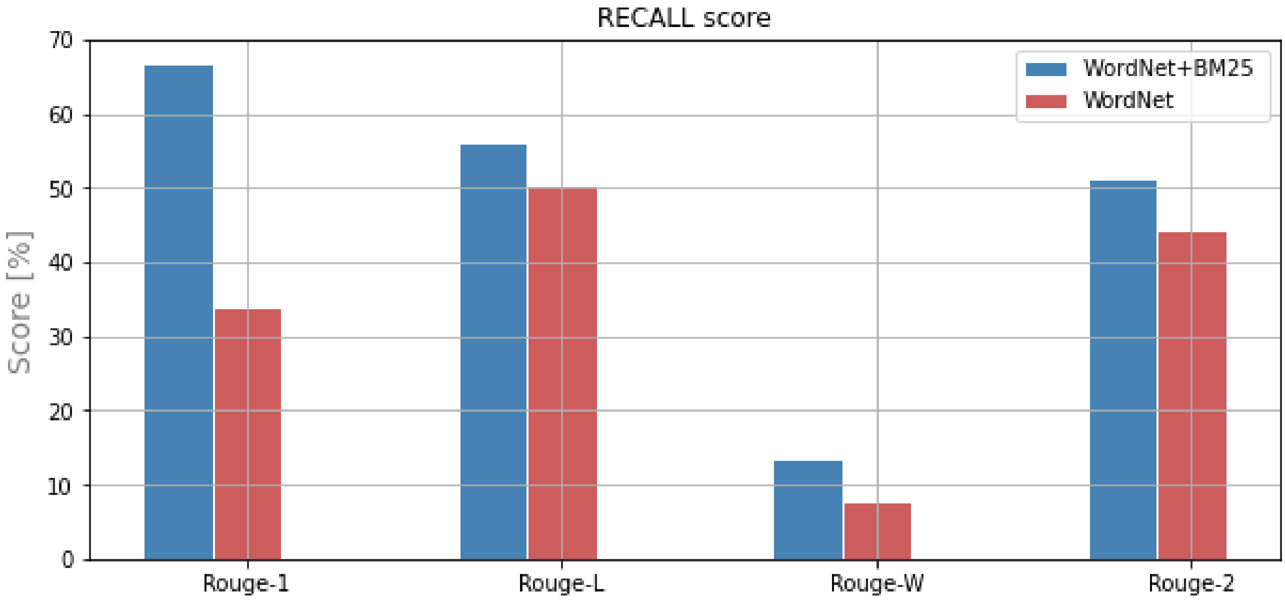


Figure 4.1: Comparison of Average RECALL values for 4 metrics of ROUGE

4.5.2 Comparison of PRECISION

We compare the average precision values of our summaries generated by the system using WordNet with average Recall values of our summaries generated by THE system using WordNet and BM25. The Table below gives the comparison for ROUGE- 1, 2, L, W metrics:

metric	**average-precision(WordNet)(%)**	**average-precision(WordNet+BM25)(%)**
Rouge-1	58.08	69.69
Rouge-2	39.39	54.16
Rouge-L	38.41	58.19
Rouge-W	24.06	45.18

Table 4.7: Comparison of average precision of our system

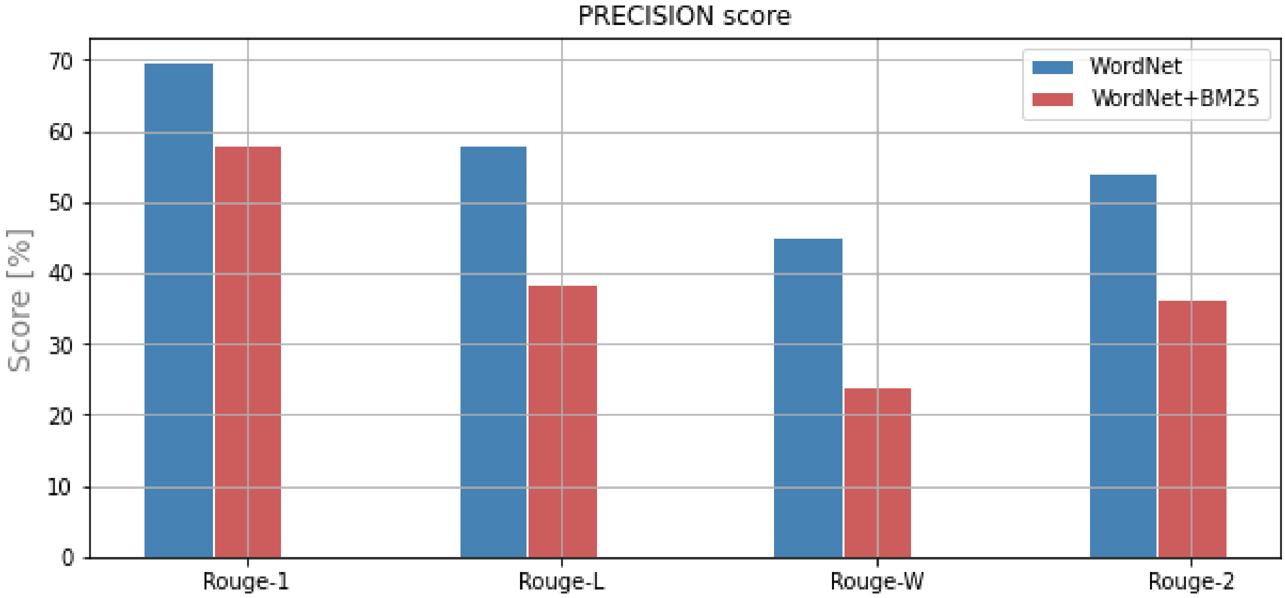


Figure 4.2: Comparison of Average F-measure values for 4 metrics of ROUGE

4.5.3 Comparison of F-measure

We compare the average F-measure values of our summaries generated by our system using WordNet with average F-measure values of our system using WordNet and BM25. The Table below gives the comparison for ROUGE- 1, 2, L, W metrics:

metric	**average-F-measure(WordNet)**	**average-F-measure(WordNet+BM25)(%)**
Rouge-1	58.09	67.68
Rouge-2	37.31	52.28
Rouge-l	38.52	56.90
Rouge-w	11.70	20.52

Table 4.8: Comparison of average F-measure of our system

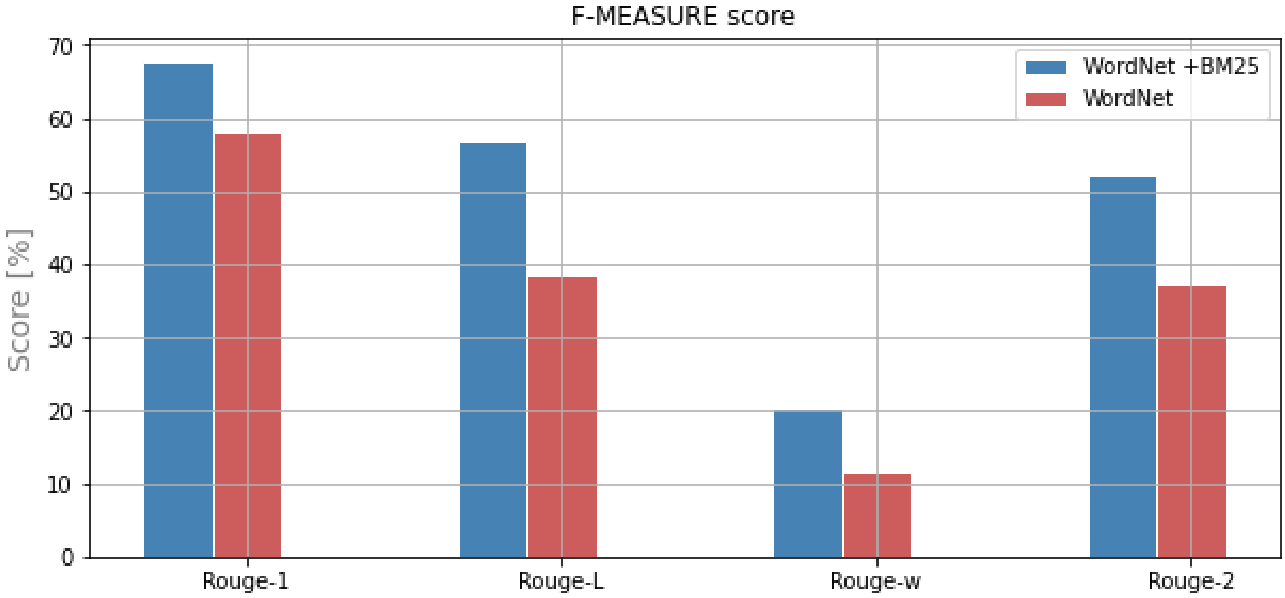


Figure 4.3: Comparison of Average F-measure values for 4 metrics of ROUGE

Results and Discussion:

- A good summary is a summary in which precision and RECALL are high.
- The tables and figures above are an analysis of the performance of two approaches used: WordNet vs WordNet +BM25 against some reference summaries. The documents analyzed are related to Corona virus and extracted from our database.
- The proposed method(combine WordNet with BM25) produces good results compared with human written extractive summaries. Our algorithm has proved to perform well for most summarization purposes.
- From the table 4.2 and 4.3, it can be noted from the results of each document that the

implemented system using WordNet and BM25 performs well than using just WordNet.

- RECALL, is the measure of how many sentences retrieved by the algorithm are actually relevant to the human summary, a higher RECALL measures the effectiveness of the algorithm. From the figure 4.1, it can be observed that the proposed system performed well for Recall score across all metrics compared to WordNet model.
- Precision is the measure of how many sentences considered for summary were actually relevant in comparison to the text in the parent document. The results from figure 4.2 show that the algorithm that uses combination of WordNet and BM25 performed reasonably well for precision performance against WordNet algorithm.
- The figure 4.3 shows that the proposed system(WordNet and BM25) correlates best with human summaries for F-measure performance.

4.6 Evaluation of the second step

To evaluate this phase, we have used the summaries resulted from the ten documents in the first phase using WordNet and BM25 to generate a single extractive summary, then we compared this summary with two human-generated summaries.

The table below shows the precision, RECALL, and F-measure score of the summary generated from the 5 summaries results from the first step using TextRank algorithm compared with two references(human summaries):

Metric	**reference 1,reference 2**			**reference 1, summary**			**reference 2, summary**		
	P	R	F	P	R	F	P	R	F
Rouge-1 (%)	80.63	88.03	84.17	73.90	77.05	75.44	56.76	62.24	59.38
Rouge-2 (%)	71.43	78.00	74.57	57.95	60.43	59.16	30.24	33.16	31.63
Rouge-L (%)	83.06	89.36	86.10	76.89	79.61	78.23	60.36	65.18	62.67
Rouge-W (%)	43.76	23.60	30.66	43.56	22.44	29.62	29.67	16.24	20.99

Table 4.9: recall, precision, and F-measure values of the generated summary compared with two references

The figures below compare the precision, recall, and F-measure values of the generated summary with two references

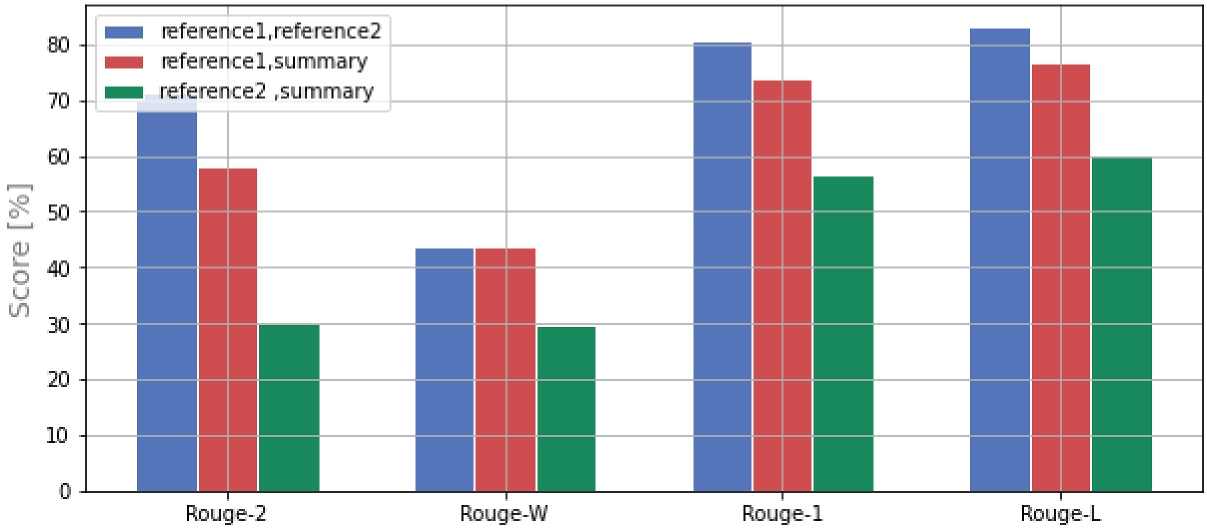


Figure 4.4: Comparison of precision- second step

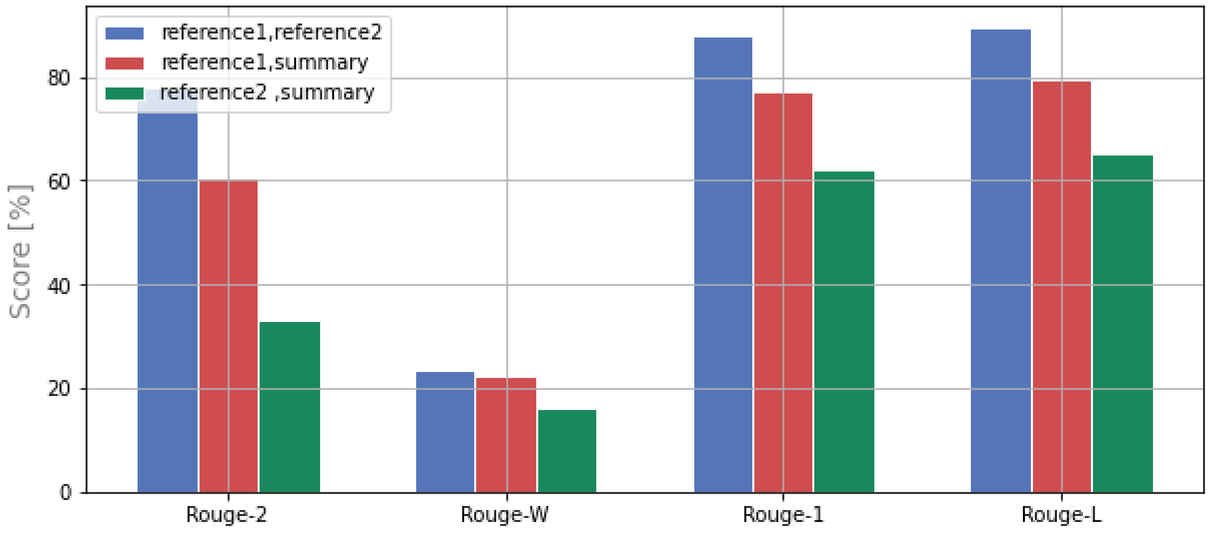


Figure 4.5: Comparison of RECALL- second step

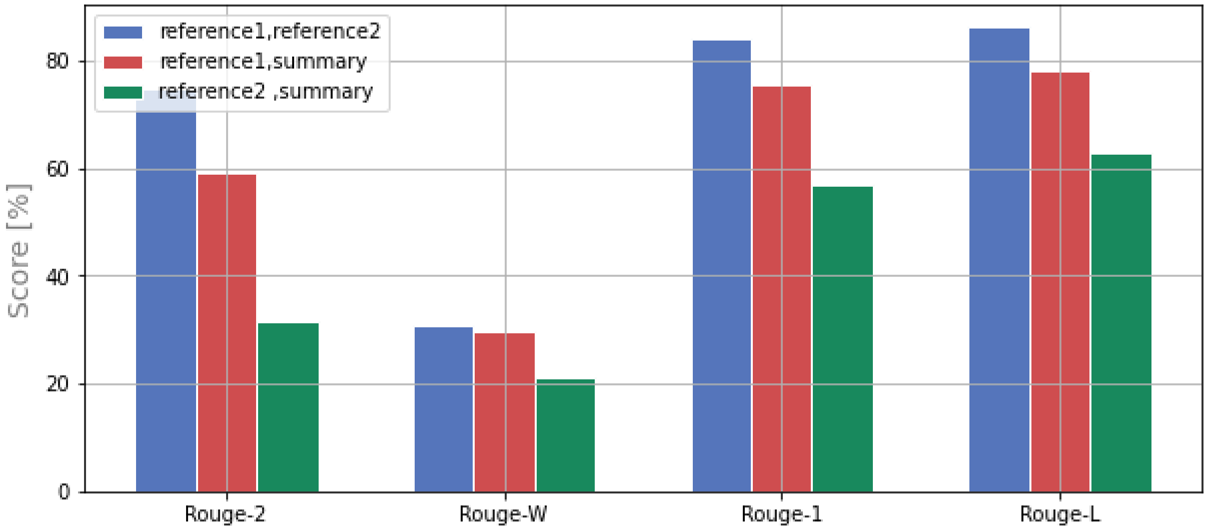


Figure 4.6: Comparison of F-measure - second step

The evaluation of an automatic text summarization is a complex task area in which a considerable amount of work has been done by researchers because the references used differ from one person to another which gives different results, and since our dataset does not contain any references for this kind of summarization, we had to do them by ourselves so that we could evaluate the summary results by our system.

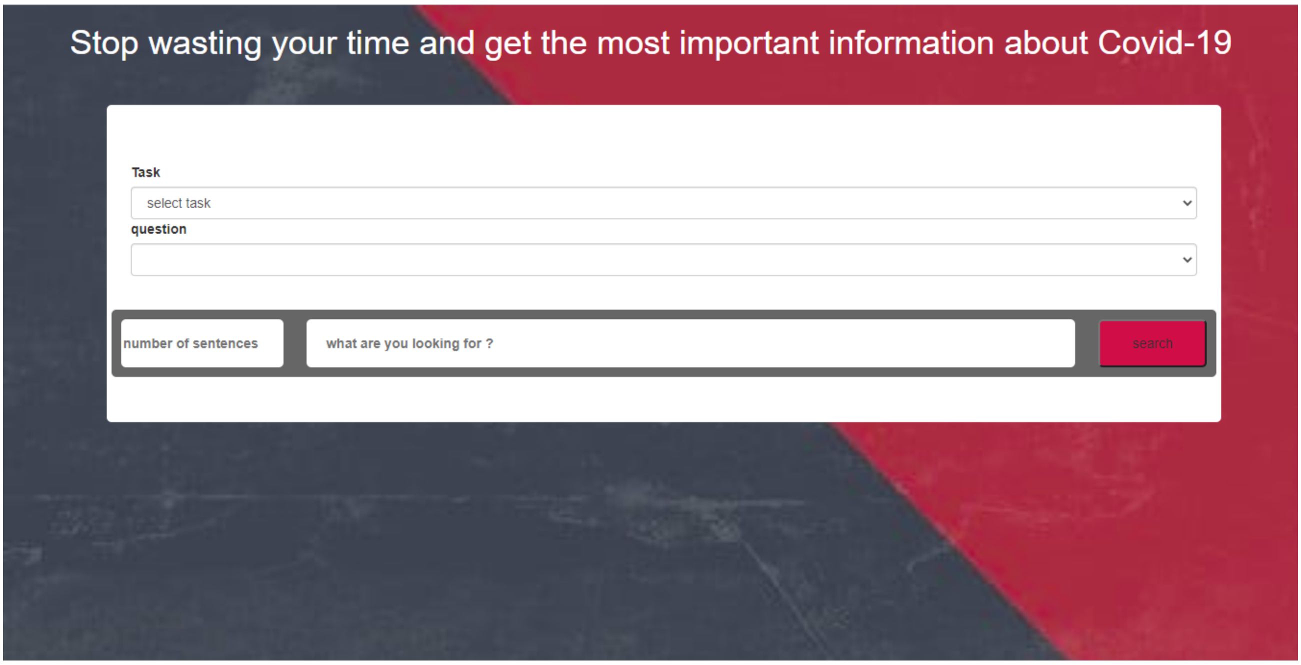
The results given in the table above show that comparing our system with reference summaries performed well and gives a very satisfied results (over 50%) in terms of precision, recall and F-measure, which is pretty good.

4.7 Web Application with Django Framework

The proposed system was created to summarize articles based on the user's need. Its mission is to provide an efficient manner of understanding text, which is done primarily by providing just the most important sentences related to user's query. Our system accomplishes its mission by:

- Retrieving the user's query.
- Removing unnecessary clauses and excessive examples.
- Looking for sentences that have relations with the query.
- Ranking sentences by importance using the core algorithm and combine them in one single text.

We have created this simple web interface in order to make it easier to find what does the user look for and display the generated summary in an attractive way.



Stop wasting your time and get the most important information about Covid-19

Task: select task

question:

Figure 4.7: web-interface of the proposed system

The web-interface contains a chained drop-down list and a search bar, the drop-down list contains all the questions proposed by the Word Health Organization, these questions are grouped into tasks.

The user starts by selecting a task from the drop-down list, each task has its own questions, then for this task, he chooses one of the displayed questions, enters the number of sentences that must be in the final summary, click on the search button and get the result.

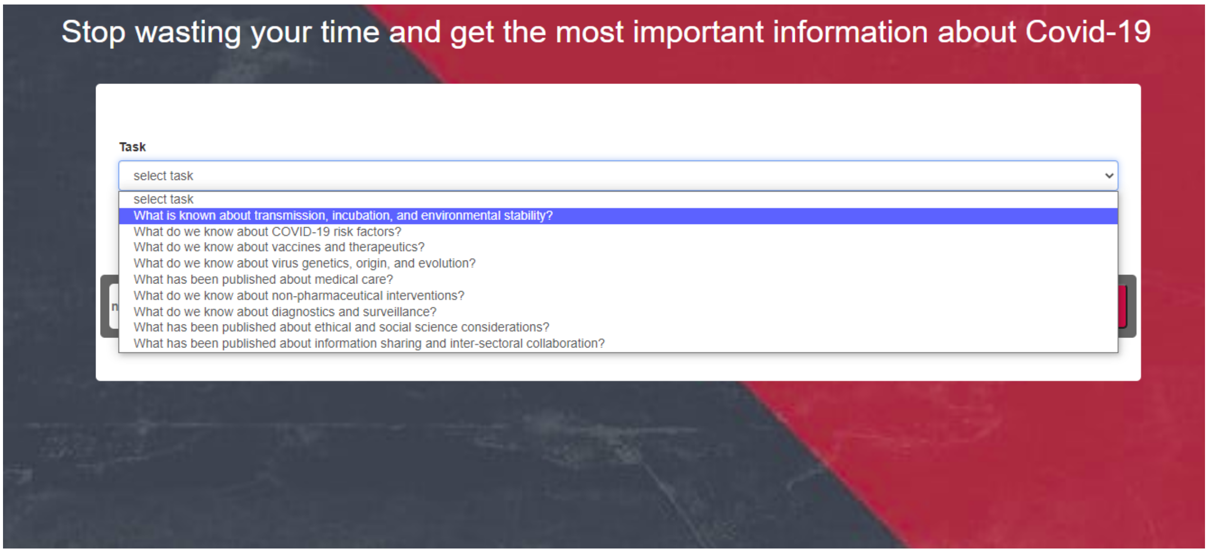
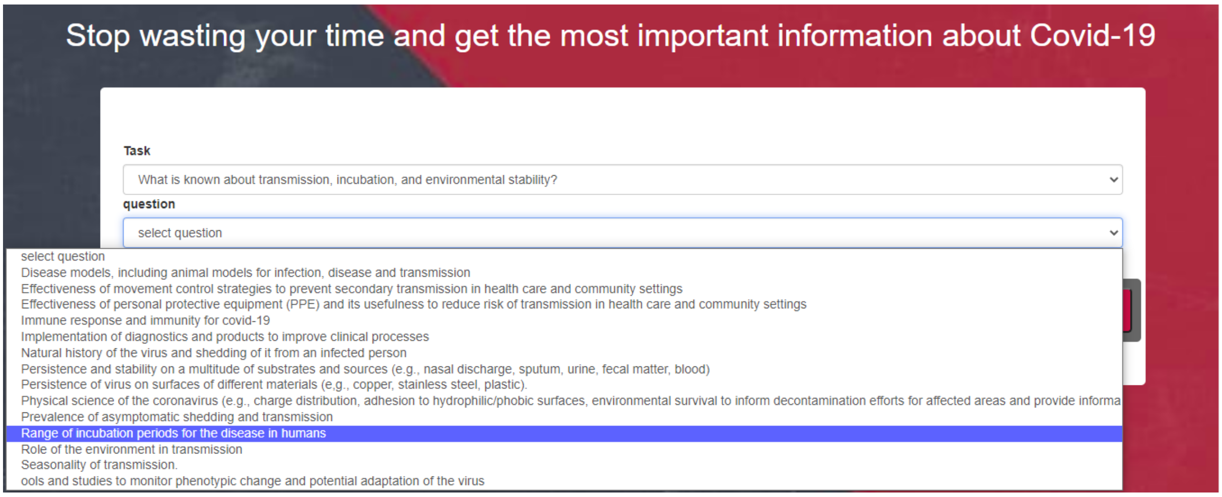

Stop wasting your time and get the most important information about Covid-19
Task
select task
select task
What is known about transmission, incubation, and environmental stability?
What do we know about COVID-19 risk factors?
What do we know about vaccines and therapeutics?
What do we know about virus genetics, origin, and evolution?
What has been published about medical care?
What do we know about non-pharmaceutical interventions?
What do we know about diagnostics and surveillance?
What has been published about ethical and social science considerations?
What has been published about information sharing and inter-sectoral collaboration?

Figure 4.8: Task selection


Stop wasting your time and get the most important information about Covid-19
question
select question
select question
Disease models, including animal models for infection, disease and transmission
Effectiveness of movement control strategies to prevent secondary transmission in health care and community settings
Effectiveness of personal protective equipment (PPE) and its usefulness to reduce risk of transmission in health care and community settings
Immune response and immunity for covid-19
Implementation of diagnostics and products to improve clinical processes
Natural history of the virus and shedding of it from an infected person
Persistence and stability on a multitude of substrates and sources (e.g., nasal discharge, sputum, urine, fecal matter, blood)
Persistence of virus on surfaces of different materials (e,g., copper, stainless steel, plastic).
Physical science of the coronavirus (e.g., charge distribution, adhesion to hydrophilic/phobic surfaces, environmental survival to inform decontamination efforts for affected areas and provide informa
Prevalence of asymptomatic shedding and transmission
Range of incubation periods for the disease in humans
Role of the environment in transmission
Seasonality of transmission.
ools and studies to monitor phenotypic change and potential adaptation of the virus

Task: What is known about transmission, incubation, and environmental stability?

Figure 4.9: Question selection

If the desired question that he is looking for doesn't exist in the drop-down list, he can write the query directly in the search bar.

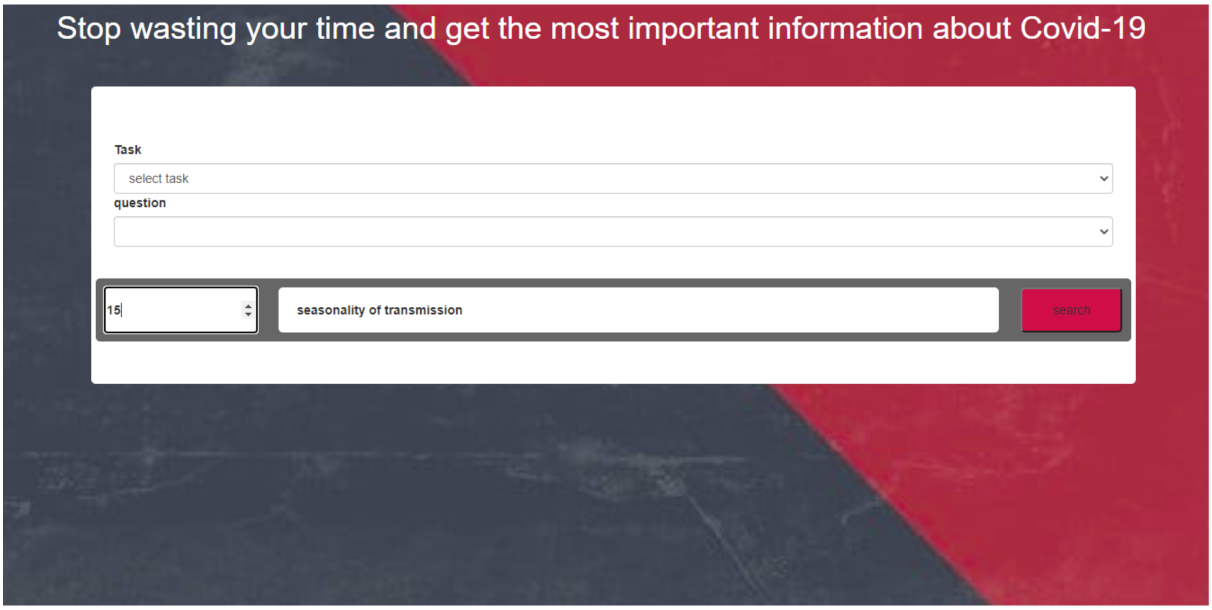
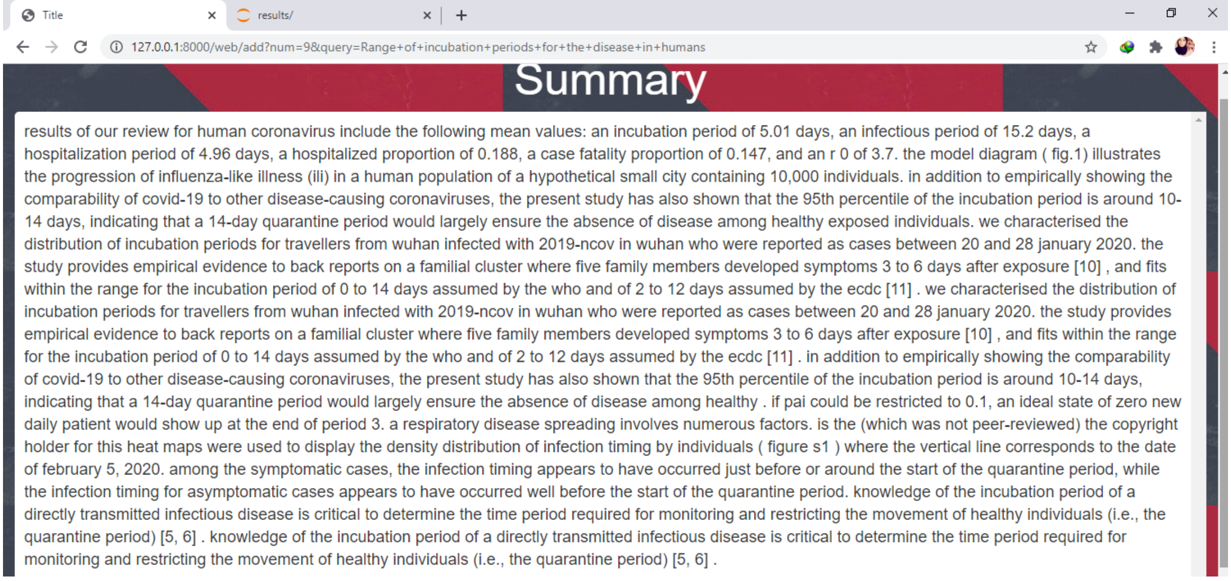

Stop wasting your time and get the most important information about Covid-19
Task
select task
question
15
seasonality of transmission
search

Figure 4.10: search bar

The generated summary is displayed as the following picture:


Title
results/
127.0.0.1:8000/web/add?num=9&query=Range+of+incubation+periods+for+the+disease+in+humans
Summary

results of our review for human coronavirus include the following mean values: an incubation period of 5.01 days, an infectious period of 15.2 days, a hospitalization period of 4.96 days, a hospitalized proportion of 0.188, a case fatality proportion of 0.147, and an r 0 of 3.7. the model diagram (fig.1) illustrates the progression of influenza-like illness (ili) in a human population of a hypothetical small city containing 10,000 individuals. in addition to empirically showing the comparability of covid-19 to other disease-causing coronaviruses, the present study has also shown that the 95th percentile of the incubation period is around 10-14 days, indicating that a 14-day quarantine period would largely ensure the absence of disease among healthy exposed individuals. we characterised the distribution of incubation periods for travellers from wuhan infected with 2019-ncov in wuhan who were reported as cases between 20 and 28 january 2020. the study provides empirical evidence to back reports on a familial cluster where five family members developed symptoms 3 to 6 days after exposure [10] , and fits within the range for the incubation period of 0 to 14 days assumed by the who and of 2 to 12 days assumed by the ecdc [11] . we characterised the distribution of incubation periods for travellers from wuhan infected with 2019-ncov in wuhan who were reported as cases between 20 and 28 january 2020. the study provides empirical evidence to back reports on a familial cluster where five family members developed symptoms 3 to 6 days after exposure [10] , and fits within the range for the incubation period of 0 to 14 days assumed by the who and of 2 to 12 days assumed by the ecdc [11] . in addition to empirically showing the comparability of covid-19 to other disease-causing coronaviruses, the present study has also shown that the 95th percentile of the incubation period is around 10-14 days, indicating that a 14-day quarantine period would largely ensure the absence of disease among healthy . if pai could be restricted to 0.1, an ideal state of zero new daily patient would show up at the end of period 3. a respiratory disease spreading involves numerous factors. is the (which was not peer-reviewed) the copyright holder for this heat maps were used to display the density distribution of infection timing by individuals (figure s1) where the vertical line corresponds to the date of february 5, 2020. among the symptomatic cases, the infection timing appears to have occurred just before or around the start of the quarantine period, while the infection timing for asymptomatic cases appears to have occurred well before the start of the quarantine period. knowledge of the incubation period of a directly transmitted infectious disease is critical to determine the time period required for monitoring and restricting the movement of healthy individuals (i.e., the quarantine period) [5, 6] . knowledge of the incubation period of a directly transmitted infectious disease is critical to determine the time period required for monitoring and restricting the movement of healthy individuals (i.e., the quarantine period) [5, 6] .

Figure 4.11: Generated summary

4.8 Conclusion

In this chapter, we have talked first about programming languages and development platform used to implement our system which are python, anaconda, and Django, then we have evaluated our system by using the Rouge package and discussed the results that have proved the effectiveness of our system. Finally we have shown some screen-shots of our web application.

Conclusion

In order to help the medical community to find answer to their important questions about COVID-19 from a huge number of published scientific papers and grab the essential knowledges quickly, we have proposed to create an extractive query-based multi-document summarization system.

The aim of the proposed system is to capture the user's query and from a large number of documents extracts the most important informations related to this query and combines them into one short summary.

At first, we have started this thesis with a generality about crisis management and its four steps, then we have explained what is the situational awareness for crisis management and the role of information and communication technology during a crisis.

Secondly, we have introduced the automatic text summarization and its characteristics, we have talked about its various types and the different methods exists to generate an automatic summary.

After that, we have mentioned some related work, defined the problem addressed and the proposed solution, we have found that using two algorithms or more gives more effectiveness because we benefit from the strength of each algorithm and also each algorithm overcomes the weakness of the second algorithm. The proposed approach was based on dividing the extractive query based multi-document summarization process into two steps, the output of the first step serves as an input to the second step.

The first step uses two algorithms which are BM25 OKAPI and the semantic similarity algorithm using WordNet to generate for each document an extractive summary corresponding to user's query. The second step uses TextRank algorithm to generate a global summary from the set of summaries that are resulted in the first step.

Finally, we have evaluated the system's effectiveness by using the Rouge package which have proved that the proposed solution provides a good quality of summaries.

Some perspectives :

Automatic text summarization was always and is still a complex process that gained the attention of researchers due to the challenges it presents in providing a well-understandable summary, and as perspective,we aim also to use the Word embedding technique to measure the semantic similarity between sentence,s and for the combination of scores (BM25 and Semantic Similarity), it would be better to combine it with a variable combination factor to get best configuration ? for example:

$$SCORE = \beta * (BM25 - score) + (1 - \beta) * (semantic - similarity - score)$$

with between 0 and 1. It will also be very interesting if we try to generate an extractive or abstractive summary using the deep learning algorithms.

Bibliography

[1] Aicha Aid. *Formulation d'un environnement générique d'un service dans un système pervasif public en cas de situation d'urgence.* PhD thesis, Université Mouloud Mammeri, 2016.

[2] Carine Rongier. *Gestion de la réponse à une crise par la performance: vers un outil d'aide à la décision. Application à l'humanitaire.* PhD thesis, 2012.

[3] Amina Saoutal. *Amélioration de l'awareness informationnelle dans la collaboration inter-organisations pendant la gestion de crise.* PhD thesis, Troyes, 2015.

[4] Himayatullah Khan, Laura Giurca Vasilescu, Asmatullah Khan, et al. Disaster management cycle-a theoretical approach. *Journal of Management and Marketing*, 6(1):43–50, 2008.

[5] Naveen Ashish, Ronald Eguchi, Rajesh Hegde, Charles Huyck, Dmitri Kalashnikov, Sharad Mehrotra, Padhraic Smyth, and Nalini Venkatasubramanian. Situational awareness technologies for disaster response. In *Terrorism informatics*, pages 517–544. Springer, 2008.

[6] Sarah Elizabeth Vieweg. *Situational awareness in mass emergency: A behavioral and linguistic analysis of microblogged communications.* PhD thesis, University of Colorado at Boulder, 2012.

[7] Vagelis Hristidis, Shu-Ching Chen, Tao Li, Steven Luis, and Yi Deng. Survey of data management and analysis in disaster situations. *Journal of Systems and Software*, 83(10):1701–1714, 2010.

[8] Dipanjan Das and André FT Martins. A survey on automatic text summarization, 2007, 2007.

[9] Juan-Manuel Torres-Moreno. *Automatic text summarization*. John Wiley & Sons, 2014.

[10] Mohamed Hedi Maaloul. *Approche hybride pour le résumé automatique de textes. Application à la langue arabe*. PhD thesis, 2012.

[11] N Nazari and MA Mahdavi. A survey on automatic text summarization. *Journal of AI and Data Mining*, 7(1):121–135, 2019.

[12] Rafael Ferreira, Luciano de Souza Cabral, Rafael Dueire Lins, Gabriel Pereira e Silva, Fred Freitas, George DC Cavalcanti, Rinaldo Lima, Steven J Simske, and Luciano Favaro. Assessing sentence scoring techniques for extractive text summarization. *Expert systems with applications*, 40(14):5755–5764, 2013.

[13] Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, page 113679, 2020.

[14] Florian Boudin, Juan-Manuel Torres-Moreno, and Marc El-Béze. Improving update summarization by revisiting the mmr criterion. *arXiv preprint arXiv:1004.3371*, 2010.

[15] Abdelkrime Aries, Walid Khaled Hidouci, et al. Automatic text summarization: What has been done and what has to be done. *arXiv preprint arXiv:1904.00688*, 2019.

[16] Mahak Gambhir and Vishal Gupta. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66, 2017.

[17] Mahak Gambhir and Vishal Gupta. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66, 2017.

[18] Mohamed Atef Mosa, Arshad Syed Anwar, and Alaa Hamouda. A survey of multiple types of text summarization with their satellite contents based on swarm intelligence optimization algorithms. *Knowledge-Based Systems*, 163:518–532, 2019.

[19] Josef Steinberger and Karel Ježek. Evaluation measures for text summarization. *Computing and Informatics*, 28(2):251–275, 2012.

[20] European Centre for Disease Prevention and Control. Covid-19 situation update worldwide,as of 10 july 2020. Accessed: 10-07-2020.

[21] World Health Organization. Coronavirus, 1948. Accessed: 10-07-2020.

[22] United Nations Development Programme. Covid-19 pandemic humanity needs leadership and solidarity to defeat the coronavirus. Accessed: 10-07-2020.

[23] Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[24] Daniel M Dunlavy, Dianne P O'Leary, John M Conroy, and Judith D Schlesinger. Qcs: A system for querying, clustering and summarizing documents. *Information processing & management*, 43(6):1588–1605, 2007.

[25] Frank Schilder and Ravikumar Kondadadi. Fastsum: fast and accurate query-based multi-document summarization. In *Proceedings of ACL-08: HLT, short papers*, pages 205–208, 2008.

[26] Michael Rosner and Carl Camilleri. Multisum: query-based multi-document summarization. pages 25–32, 2008.

[27] Christiane Fellbaum. Wordnet: An electronic lexical database cambridge. *MA: MIT Press*, 1998.

[28] Bridget T McInnes and Ted Pedersen. Evaluating measures of semantic similarity and relatedness to disambiguate terms in biomedical text. *Journal of biomedical informatics*, 46(6):1116–1124, 2013.

[29] Shailesh Padave. Incorporating wordnet in an information retrieval system,wikipidia. 2014.

[30] elastic. Similarity module. Accessed: 06-10-2020.

[31] Kaggle. Covid-19 open research dataset challenge (cord-19). Accessed: 09-04-2020.

[32] Sagar M Patel, Vipul K Dabhi, and Harshadkumar B Prajapati. Extractive based automatic text summarization. *JCP*, 12(6):550–563, 2017.

[33] KDnuggets. Natural language processing key terms, explained, 2017.

[34] Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. Variations of the similarity function of textrank for automated summarization. *arXiv preprint arXiv:1602.03606*, 2016.

[35] pythonforbeginners. What is python, 2012. Accessed: 19-08-2020.

[36] Krishna Guru99. Nltk (natural language toolkit) tutorial in python. Accessed: 19-08-2020.

[37] tutorialspoint. Numpy tutorial, 2006. Accessed: 19-08-2020.

[38] Eric Firing Michael Droettboom John Hunter, Darren Dale and the Matplotlib development team. matplotlib, 2012. Accessed: 19-08-2020.

[39] HOSTINGER tutorials. What is html? the basics of hypertext markup language explained, 2004. Accessed: 19-08-2020.

[40] Tutorialspoint. Css tutorial, 2006. Accessed: 05-10-2020.

[41] Django. https://www.djangoproject.com/, 2005. Accessed: 19-08-2020.

[42] RxNLP. What is rouge and how it works for evaluation of summarization tasks? Accessed: 05-10-2020.