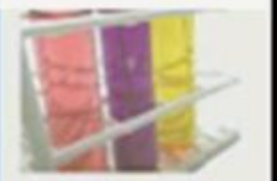


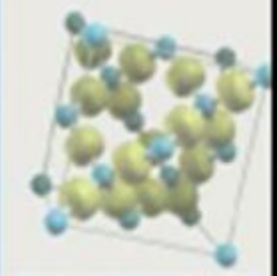

Faculty of Engineering - Cairo University
Credit Hour System Programs

Computer and Communications Engineering

CCE







Graduation Project Report

Spring 2021

NUTSHELL

Prepared by:

Samy Saeed Abdallah
Ahmad Nader Adel
Kamel Mohsen Kamel
Ziad AbdelHamid Sadek

Supervised by:

Prof. Magda Fayek

Cairo University
Faculty of Engineering
Department of Computer Engineering

A Graduation Project Report Submitted

to

Faculty of Engineering, Cairo University

in Partial Fulfillment of the requirements of the degree

of

Bachelor of Science in Computer Engineering.

Presented by

Ahmed Nader Adel
Kamel Mohsen Kamel
Samy Saeed Adballah
Ziad Abdelhamid Sadek

Supervised by

Dr. Magda Fayek

Abstract

Throughout time developing new technology has been used to ease and improve human lives. Nowadays, Humans' most precious resource is time, given enough time almost anything can be done. Given enough time, the chance to discover new things and humans' efficiency increases, it ranges from curing diseases and inventing new technologies to finishing everyday tasks. One aspect that consumes a lot of peoples' time is getting updated with news. With the increased flood of data over the past years it has been almost impossible for people to catch up with the daily news as time is limited and not enough to go through all news. Thus a problem has emerged over the past years, people have been missing out on many news that may be relevant or important to them. Summarization tools have thus recently been a hot topic in NLP society and are actually used to solve similar problems such as removing redundant data and extracting relevant data from social media platforms. The NUTSHELL system proposes an approach that uses a cascade of extractive and abstractive text summarization for solving the problem mentioned above. The approach proposed above has proved that it is an improvement to other approaches currently used for multi-document summarization such as, Maximal Marginal Relevance, with a Rouge(R1) score of 0.3348 compared to 0.32809 of MMR [1]. NUTSHELL will be the new source of news for people, as the news is summarized and only relevant data is shown to the user. The tool will allow the user to search the web for a specific topic and it will collect data and summarize it then show it to the user in an easy and readable way.

الملخص

على مر الزمن ، تم تطوير تقنيات جديدة لتسهيل حياة الإنسان وتحسينها. في الوقت الحاضر ، يعد الوقت أثمن مورد لدى البشر ، مع توفر الوقت الكافي ، يمكن فعل أي شيء تقريبًا. مع توفر الوقت الكافي ، تزداد فرصة اكتشاف أشياء جديدة وزيادة كفاءة البشر ، وتتراوح من علاج الأمراض وابتكار تقنيات جديدة إلى إنهاء المهام اليومية. و تعتبر متابعة الاخبار من اكثر المناحي التي تستهلك جزء كبير من وقت الأفراد . ومع تدفق البيانات المتزايد على مدى السنوات الماضية ، أصبح من المستحيل متابعة الأخبار اليومية لأن الوقت محدود وغير كافٍ لتصفح جميع الأخبار. أدى ذلك لجعل الناس تتخطى الكثير من الأخبار التي قد تكون ذات صلة أو مهمة بالنسبة لهم. أصبحت أدوات التلخيص مؤخرًا موضوعًا رائجا في مجتمع البرمجة اللغوية العصبية ويتم استخدامها بالفعل لحل مشكلات مماثلة مثل المعلومات المتكررة و استخراج المعلومات المهمة من نصوص الشبكات الاجتماعية. يقترح مشروع NUTSHELL نهجًا يستخدم سلسلة من تلخيص النص الاستخراجي(extractive) والتجريدي(abstractive) لحل المشكلة المذكورة أعلاه. لقد أثبت النهج المقترح أعلاه أنه يتفوق على الأساليب الأخرى المستخدمة حاليًا مثل MMR, بدرجة Rouge R1 متفوقة وهي 0.3348 مقارنة بـ 0.32809 لصالح MMR ي[1]. سيكون NUTSHELL هو المصدر الجديد للأخبار للأشخاص ، حيث يتم تلخيص الأخبار ويتم عرض البيانات ذات الصلة فقط للمستخدم. ستسمح الأداة للمستخدم بالبحث في الويب عن موضوع محدد مسبقا و تقوم بجمع البيانات وتلخيصها ثم عرضها على المستخدم بطريقة يمكن قراءتها بأسلوب سلس.

ACKNOWLEDGMENT

First, we would like to thank God for guiding us and providing us with the patience and perseverance to implement this project.

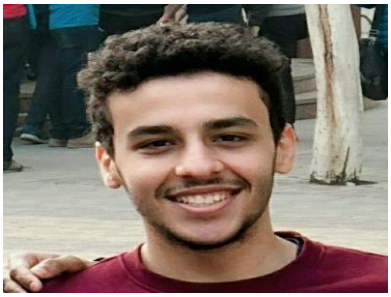
Secondly, we would like to express our utmost gratitude to Prof. Magda Fayek for her guidance, patience, and her unmatched expertise.

Contacts

<table>
<tr><td>Project Code</td><td colspan="2">CCEN481</td></tr>
<tr><td>Project Title</td><td colspan="2">NUTSHELL</td></tr>
<tr><td>Keywords</td><td colspan="2">Natural Language Processing, Machine learning, Summarization</td></tr>
<tr><td rowspan="2">Students</td><td>Name: Samy Saeed Abdallah
 
Email: samy.hafeiz97@eng-st.cu.edu.eg

Phone: +2 01225766651</td><td>Name: Ahmad Nader Adel
 
Email: ahmadnader98@gmail.com

Phone: +2 01121155885</td></tr>
<tr><td>Name: Kamel Mohsen Kamel
 
Email: kamelmohsenkamel@gmail.com

Phone:+2 01097000365</td><td>Name: Ziad AbdelHamid Sadek
 
Email: ziadzizo1999@gmail.com

Phone:+2 01013780013</td></tr>
<tr><td rowspan="2">Supervisor</td><td>Name: Prof Magda Fayek</td><td>Email:magdafayek@gmail.com</td></tr>
<tr><td>Phone:</td><td>Signature:</td></tr>
<tr><td>Project Summary</td><td colspan="2">Multi Document summarization tool for news articles</td></tr>
</table>

Table of Contents

List of Figures

List of Tables

Table of Abbreviations

Abbreviation	Definition
API	Application Programming Interface
UI	user interface
ML	Machine Learning
NLP	Natural Language Processing
ANN	Artificial Neural Network
AI	Artificial Intelligence
TF	Term Frequency
IDF	Inverse Document Frequency
SVD	Singular Value Decomposition
KL-sum	Kullback-Leiber Sum Algorithm
BERT	Bidirectional Encoder Representations from Transformers
ELM	Extreme Learning Machine
ELM-AE	Extreme Learning Machine Autoencoder
TPU	Tensor Processing Unit
CSS	Cascading style sheets
HTML	Hyper Text Markup Language
LSTM	Long Short Term Memory
MLP	Multilayer perceptron
OOV	out of vocabulary

This page is left intentionally empty

Chapter 1: Introduction

Summarization, whether extractive or abstractive, is perhaps the most common linguistic task. Summaries are so demanded in the modern world that their concepts are taught in secondary education all over the world. Progress Meetings, television programs, and even newspaper columns all employ the use of summaries. In this report, we focus on multi-document text summarization.

Abstractive summarization is one of the challenging tasks of NLP, yet in recent years a lot of advances have been made in this field. Although there's still a lot to be achieved in the field of single-page document summarization, the amount of unprocessed information that floods the internet every day has placed more urgency and importance on the field of multi-document summarization.

1.1. Motivation and Justification

Millions of articles and documents are uploaded to the internet every day. The constant enormous stream of information goes unanalyzed and unprocessed while it occupies terabytes and terabytes of server storage. As the amount of material grows, it is becoming more difficult for users to find the relevant information they are looking for. This system aims to summarize articles or news related to a specific topic in a meaningful way.

1.2. Objectives and Problem Definition

The system aims to make reaching the daily news an easier task, hence saving the users a lot of time and effort. This system aims to provide a means to summarize multiple documents or articles on the same topic, to better analyze this enormous stream of information. The need for this system arises from the flood of information in the past years to the extent that humans can no longer cope with this amount of information with their limited time.

1.3. Outcomes

A complete end to end system that allows users to search for particular topics/news and get a summarized version of the retrieved documents. The system is able to perform on all devices using a web interface that is responsive on all devices, thus allowing more users to access the system.

1.4. Document Organization

Chapters 1 and 2 give a brief introduction to the system and to the available alternatives and how they work. Chapters 3 and 4 briefly give a technical background information about the technologies needed in the system. Chapter 5 presents the architecture for NUTSHELL, the dataset preprocessing and preparation is visited in Chapter 6, and the implementation for this architecture is explained in Chapter 7. Finally we conclude our paper with an overall summary of the system and our future plans to improve the system.

Chapter 2: Market Feasibility Study

This chapter discusses and showcases the result of the market survey we conducted to obtain more information about the previous solutions and current direction of the community in solving the problem of multi document text summarization.

2.1. Targeted Customers

With the current increase in news-reporting and increase in the frequency of urgent news, most people find themselves browsing the news daily to find critical information about current events, such as the Covid-19 crisis for example. This system targets news surfers everywhere. In addition, it can be especially useful to marketers, social media managers, reporters and journalists, as it is their daily task to know about current news and events.

2.2. Market Survey

As of the date of writing this report, there are no available commercial tools for multi-document summarization for news. We intend to create a tool that summarizes multiple news sources for the specified query that the user inputs.

To better understand the market and its needs, we conducted a survey on social media platforms. The survey was conducted in November of 2020, and was taken by 52 users, whose ages ranged from 18-52 years old, the female percentage was about 29 % compared to 71 % for males. The majority of responders, about 75 %, were engineering students, engineers, or professors at faculty of engineering, while the other 25 % percent had no previous knowledge or relation to the topic or field of engineering.

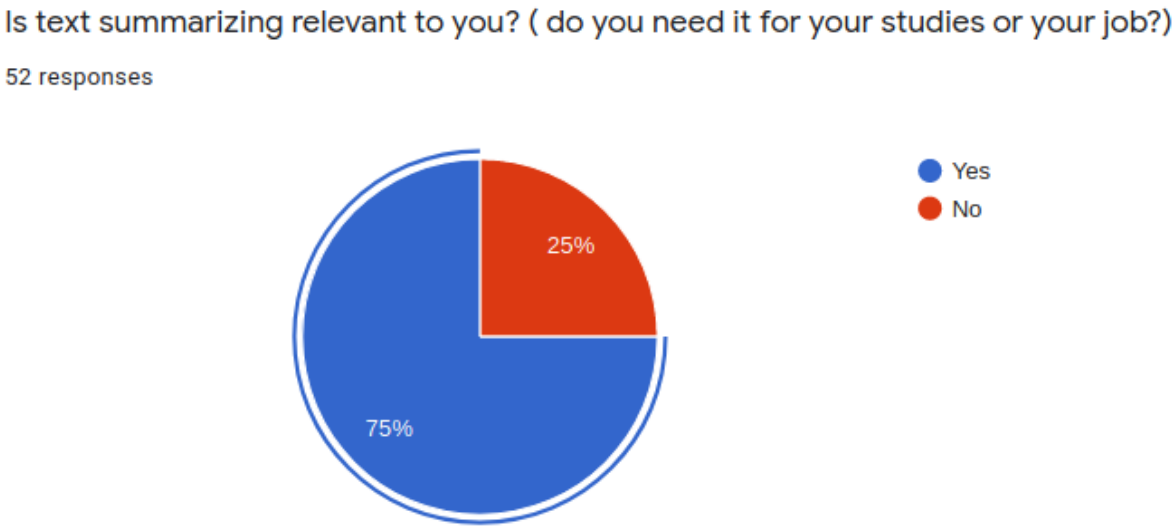


Figure 2.1 Percentage of people who find text summarization relevant

As for the frequency of needing summaries, 28.8 percent of users need summaries at least 3-4 times a week, and 46.2 percent need them at least 3-4 times a month.

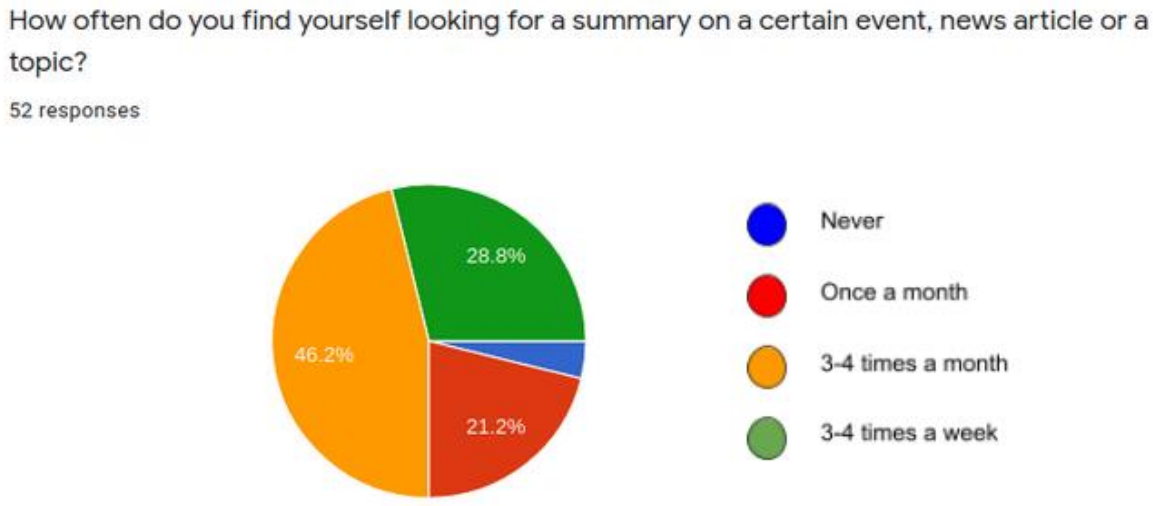


Figure 0.1.2 Frequency of people looking for summaries of news

Additionally, when given a scale of 1-5, with one being useless, and 5 being extremely useful, 53.8% of users reported it would be very useful to them.

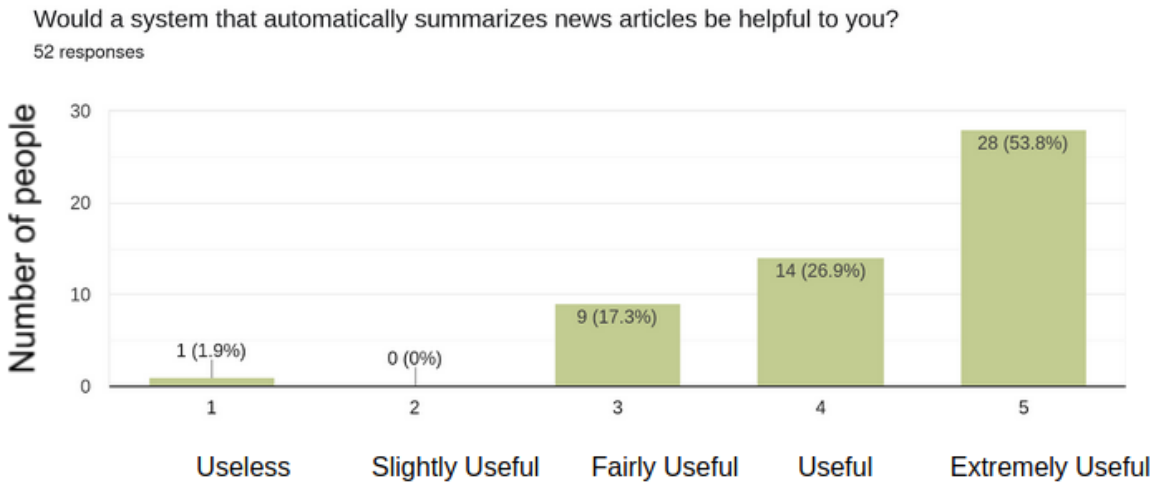


Figure 2.0.2 Number of people who find the tool useful

In the current market, there is no similar product or service available, however, Facebook is currently creating their first multi document summarization tool for news [2] and there are other services that provide single document summarization. We needed to provide a similar approach to our system, showing their points of strength as well as weaknesses. Although the services below are used for the text summarization, they are not targeted for multi-document summarization which NUTSHELL is aiming to handle.

2.2.1. Summarize Bot

It is an AI and Block chain-powered summarize bot, it summarizes any kind of information [3].

2.2.2. Resoomer

To help you summarize and analyze your argumentative texts, your articles, your scientific texts, your history texts as well as your well-structured analysis work of art, Resoomer provides you with a "Summary text tool": an educational tool that identifies and summarizes the important ideas and facts of your documents [4].

2.3. Automatic Text Summarization Applications

1. Search Engine Optimization

To perform search engine optimization, website creators have to research other websites in the same domain, to find out what type of content other websites employ in their articles/blogs. A summarization tool can help developers gain an understanding of competitors' content without wasting time.

2. E-Learning

As the world moves towards digitizing education and moving education operations online, teachers need to summarize new content to incorporate it into their lecture content.

3. News summaries

Due to the constant stream of news content, it has become hard to keep track of important information, such as new development in the COVID-19 situation. As such, a tool to summarize news in a meaningful way.

4. Academic Papers

When researching certain topics, researchers often need to decide whether academic papers are relevant to their course of action or not, and as such, they need a rough summarization of what the paper discusses and what approaches were used.

2.4. Technological Feasibility

In this subsection we are going to list the technologies used in the system. The technologies we used in this system are:

- Backend framework: Flask
- Frontend framework: ReactJS
- Machine Learning framework: PyTorch, Google Colab
- Cloud Services: Amazon Web Services (AWS), Google Cloud Services

2.5. Economic Feasibility

Several ways to serve a machine learning model exist today. There are two widely used options:

- Option A: Buy a domain model, deploy code to a hosting server, and the system is ready to serve users.
- Option B: Use a static web site for the client tier, serve requests through API call to server less computing.

After listing options, we had to choose the option that best fits the system requirements in the most efficient way possible. Therefore, the comparison below shows the advantages and disadvantages of each option and which option would better fit the system and why.

Option A:

1. Advantages:
    i. No cold start latency
    ii. Customizable and higher hardware can be rented for extra fees
    iii. Front-End and Back-End on the same server
    iv. No limit for usage of API calls

2. Disadvantages:
    i. Very expensive
    ii. Average server specs is not enough to serve many users

Option B:

1. Advantages:
    i. Server specs is not an issue
    ii. Fixed cost per API request
    iii. Number of users has almost no limit/restriction
2. Disadvantages:
    i. Cold start latency
    ii. Back-End and Front-End are separated

What Is Server less Computing?

"Server less computing is a method of providing backend services on an as-used basis. Servers are still used, but a company that gets backend services from a server less vendor is charged based on usage, not a fixed amount of bandwidth or number of servers".

-Cloud Flare

We chose option B, using a static web site and exposing our API on a server less compute service because of the low running cost and the increased user serving capability. The drawback with option B is the added latency at cold starts. A Cold Start is the first run in a while of a server less computing service, meaning that if a service is not used for a long time (defined by each server provider) it will go to an idle state that takes time to get out of; it usually takes longer than Warm Start, meaning that the service is not idle and is used frequently.

Hosting the static page

Multiple static hosting solutions exist. We chose Netlify. It's easy to get the job done in the least amount of time. Basic hosting, using a custom domain name, and SSL certificate are free on Netlify.

Serving the API with server less computing

Each cloud provider offers a server less computing service; we chose Google Cloud and its cloud functions.

Google cloud has a tutorial on how to serve machine learning models through cloud functions. With this tutorial as a baseline, we were able to serve our machine learning model.

On the cold start latency

We will add some UI elements to our front end to make it explicit that the first prediction may take some time.

Running Cost:

Deploying static website: Free

Server less computing: First 50,000 operations are for free (daily), then 0.05$ per operation rate is applied [5].

Therefore, we conclude that for the

2.6. Findings and Recommendations

The survey conducted above, had a majority of responses that would like to see a service as NUTSHELL. The economic feasibility study has shown that such a service would not cost much to produce, and may even be free for a limited number of users. If it goes commercial, the free or low tier solutions may not be satisfactory and more computational power will be needed to satisfy the demand, resulting in higher running cost.

Chapter 3: Background

In this chapter we will introduce Machine Learning and NLP since the system hugely depends on understanding these two sciences.

3.1. Machine Learning

In this section we give a brief explanation about Machine learning, how it operates, and the approaches of machine learning

3.1.1. Definition

Machine learning (ML) is the study of computer algorithms that improve themselves over time as a result of experience and data. It is considered to be a component of artificial intelligence. Machine learning algorithms create a model based on sample data, referred to as "training data," in order to make predictions or judgments without being explicitly programmed.

3.1.2. Operation

The learning agent, called model, attempts to generalize from its experience, and tries to classify new data based on past examples it learned. The more training data there is to learn from, the more accurate the model becomes at classifying. The complexity of the hypothesis should match the complexity of the function underlying the data for the optimal generalization results. The model has under fitted the data if the hypothesis is less complex than the function. The training error lowers when the model's complexity is increased in response. However, if the hypothesis is too complicated, the model will be prone to over fitting, resulting in poor generalization.

3.1.3. Artificial Neural Network

Artificial neural networks, commonly referred to as connectionist systems, are computer systems based on organic neural networks found in animal brains. Such systems learn to perform tasks by analyzing examples, frequently without the use of task-specific rules. An artificial neural network (ANN) is a model made up of "artificial neurons," which are connected units or nodes that are generally modelled after neurons in a biological brain. Each link may convey information, or a "signal," from one artificial neuron to the next, similar to the synapses in the human brain. An artificial neuron can receive a signal and process it before sending it to other artificial neurons.

3.1.4. Deep Learning

Deep learning is an artificial intelligence (AI) function that mimics the human brain's processing of data and pattern creation in order to make decisions. Deep learning is an artificial intelligence subset of machine learning that uses neural networks to learn unsupervised from unstructured or unlabeled data. Deep neural learning or deep neural network are two terms for the same thing.

3.2. NLP

The following sections define what NLP is and how it works and some key ideas that were used in NUTSHELL.

3.2.1. Definition

Natural language processing (NLP) is a branch of linguistics, computer science, and artificial intelligence that studies how computers interact with human language, particularly how to design computers to process and analyze massive amounts of natural language data. As a result, a computer can "understand" the contents of documents, including the intricacies of the language used within them. The system can then extract accurate information and insights from the papers, as well as categorize and organize them.

3.2.2. Applications of NLP

- Text and speech processing
- Morphological analysis
- Syntactic analysis
- Lexical semantics (of individual words in context)
- Relational semantics (semantics of individual sentences)
- Discourse (semantics beyond individual sentences)
- Higher-level NLP applications

Chapter 4: Literature Survey

In this section we give background information needed for understanding how our system is made, and then we review academic papers and their different methodologies that have been published on the subject of 'multi-document text summarization'. Various methods have been discussed, with many different specializations.

4.1. Background on Text Summarization

There are two approaches for summarizing text; extractive and abstractive.

4.1.1. Types of text Summarization

Extraction-based summarization

The extractive text summarizing approach entails extracting important words from a source material and combining them to create a summary without making any modifications to the texts, the extraction is done according to the given measure.

Abstraction-based summarization

The extractive text summarizing entails that, parts of the source material are paraphrased and condensed in the abstraction process. The grammatical inconsistencies of the extractive approach can be solved when abstraction is used for text summarization in deep learning issues.

4.1.2. Methods of text Summarization

This section gives a brief description of the methods and algorithms used for extractive and abstractive text summarization.

4.1.1.1. TextRank

TextRank is a technique for extracting information from documents. It is based on the idea that words that appear more frequently are more important. As a result, sentences with a high frequency of words are important. The algorithm then assigns scores to each sentence in the text based on this. The top-scoring sentences are included in the summary.

4.1.1.2. LexRank

A sentence that is similar to many other sentences in the text is likely to be significant. LexRank works on the principle that a sentence is recommended by other similar sentences and thus is ranked higher. The higher the rank, the more likely it is to be included in the summarized text.

4.1.1.3. Luhn

The TF-IDF technique is used in the Luhn Summarization algorithm (Term Frequency-Inverse Document Frequency). It's useful when both low-frequency words and high-frequency words (stop words) aren't important. Sentence scoring is done based on this, and the top-scoring sentences make it to the summary.

4.1.1.4. Latent Semantic Analysis, LSA

Unsupervised learning algorithm for extractive text summarization is Latent Semantic Analysis. By applying singular value decomposition (SVD) to the term-document frequency matrix, it recovers semantically meaningful sentences.

4.1.1.5. KL-Sum

It chooses sentences depending on how closely their word distribution matches that of the original text. The goal is to reduce the KL-divergence criteria. It employs a greedy optimization strategy in which it continues to add sentences until the KL-divergence reduces.

4.2. Deep learning approaches for summarization

The availability and ease of access of huge amounts of text data on the web presents both an opportunity as well as a challenge. Increased accessibility of data has led to the information overload problem. Huge research efforts have been expended on facilitating the automatic processing of such texts available online. An important task in the domain of natural language understanding is document summarization, and one of the most used methods for summarization is using deep learning.

4.2.1. Summarization with T5 Transformers

T5 is a pre-trained encoder-decoder model that has been trained on a multi-task mixture of unsupervised and supervised tasks, with each task transformed to text-to-text format.

4.2.2. Summarization with BART Transformers

In 2019, Facebook AI developed BART, which confusingly stands for Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. It makes use of a typical Transformer-based neural machine translation architecture that, despite its simplicity, can be thought of as a generalization of BERT (Bidirectional encoder).

4.2.3. Multi-Layer ELM

Extreme Learning Machine

ELMS are feed forward neural networks with a single layer. Its popularity comes from the fact that it has no back propagation, rapid learning speed and its ability to handle large datasets.

What is Multilayer ELM

An artificial neural network having multiple hidden layers called Multilayer ELM is proposed by Kasun et al. which possesses all the properties of ELM since it combines ELM with ELM-auto encoder (ELM-AE). Parameters that represent ML-ELM are trained layer-

wise using ELM-AE in an unsupervised manner. During the training time, no iteration takes place and hence the unsupervised training is very fast. The architectures of ELM-AE and ELM are almost similar except that ELM is supervised in nature, while ELM-AE is unsupervised and it can be stacked and trained in a progressive way. The stacked ELM-AEs will learn how to represent the data, the first level has a basic representation, and the second level combines that representation to create a higher-level representation and so on.

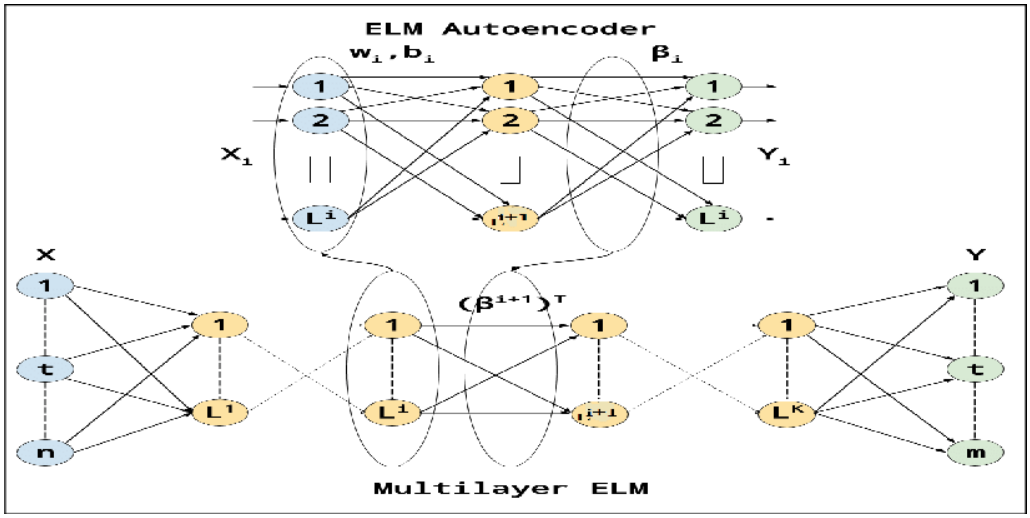


Figure 4.1 Multilayer ELM Block Diagram [6]

Modules

The system consists of 3 parts:

- Input Layer: receives the features
- Hidden Layers
- Output Layer

Extracted features from the text

- **Sentence length**: how long sentences are.
- **Sentence Weight:** by using Term Frequency (TF), which measures the frequency of a term in the document, and Inverse Document Frequency (IDF) which measures the importance of a document in the entire corpus.
- **Sentence Density:** the ratio between the total count of keywords in a sentence and the total count of words in a sentence (which includes all words like “a”, “an”, “the” and other semantically irrelevant words).

- **Presence of named entities:** terms that refer to names of people, organizations, locations or others. The presence of names often signifies the importance of the parent sentence.
- **Presence of cue-phrases:** Usually, sentences that have specific phrases like “In summary”, “To conclude”, “in particular” and other similar phrases signify that parent sentences are an important source of information.
- **Location of sentence:** Sentences at the beginning of a paragraph or at the end of a paragraph are usually more important than the rest.
- **Title words:** If words present in the title are present in a sentence, that sentence is considered important.
- **Quoted Text:** Quoted text is usually an important source of information.
- **Upper-case words:** sentences with upper-case words and phrases likely refer to important acronyms, places or names. These sentences are considered important.

Chapter 5: System Design and Architecture

In this chapter we give a brief explanation of the technology used in the design and implementation of the NUTSHELL system.

5.1. Overview and Assumptions

The system is designed to be a web service that can be accessed from anywhere and any device.

5.2. System Architecture

The system includes 4 distinct modules:

- Front-End: responsible for user interface and collecting query and query parameters from the user, and displaying the results of the summarization.
- Back-End: responsible for receiving the query from the user, scraping the web to find articles concerning this query, and then sending the articles to the trained model.
- Pointer Generator Module: Takes the scrapped articles and returns multiple abstractive summaries from them.
- LexRank Module: Takes the multiple abstractive summaries from the Pointer Generator module and generates the final summary.

5.2.1. Block Diagram

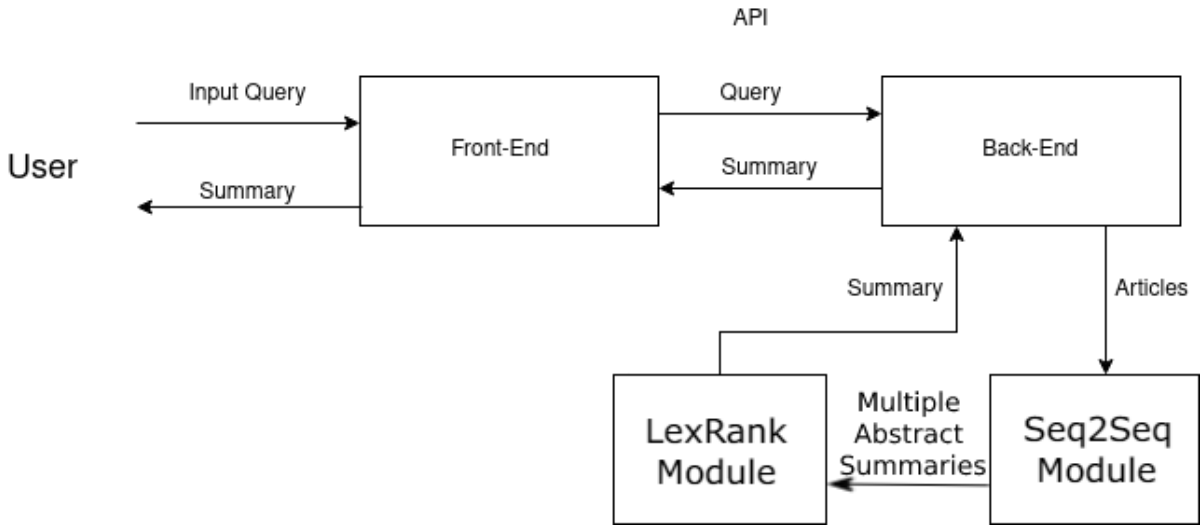


Figure 5.1 System Block Diagram

5.3. Front-End

Nutshell uses ReactJS, a JavaScript library, and Bootstrap, a CSS framework to create the front-end of the website.

5.3.1. ReactJS

ReactJS is a JavaScript library made by Facebook with the intent of building user interfaces in a quick and easy way. ReactJS relies on embedding HTML tags in JavaScript code, which facilitates creating dynamic HTML tags based on dynamic data.

Additionally, ReactJS is a free and easy to use tool. One of its variants, React Native, is a library to build Android and IOS apps by following the same principles of React, whilst utilizing the power of native views on different platforms to conserve efficiency.

5.3.2. Bootstrap

Bootstrap is a front-end open-source toolkit, widely used as a CSS framework that makes writing CSS code much easier and more compact, which improves readability and modularity of CSS code. The framework also has themes, icons and templates that help make the front-end development experience easier and more streamlined.

5.3.3. Design

The main interface presented to the user is shown in fig.6. It allows the user to enter his query in English, with no limitation on minimum or maximum length.

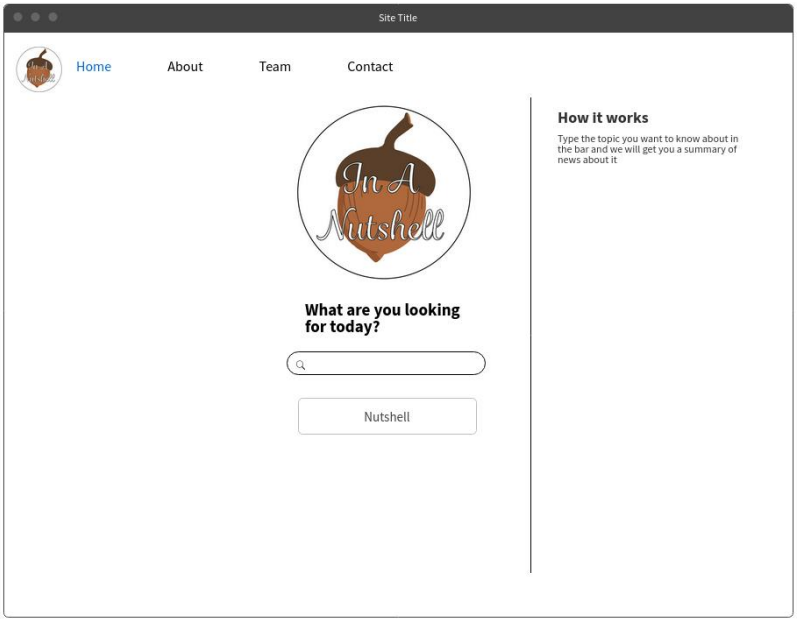


Figure 5.2 Website Landing Page

The user will also be able to control the percentage of the reduction of the text, as required. It is also allowed to specify the links he wants to be summarized. Additionally, the user can specify whether the summary should be key points or paragraphs.

The results will be returned in a page like this one shown in Figure 5.3:

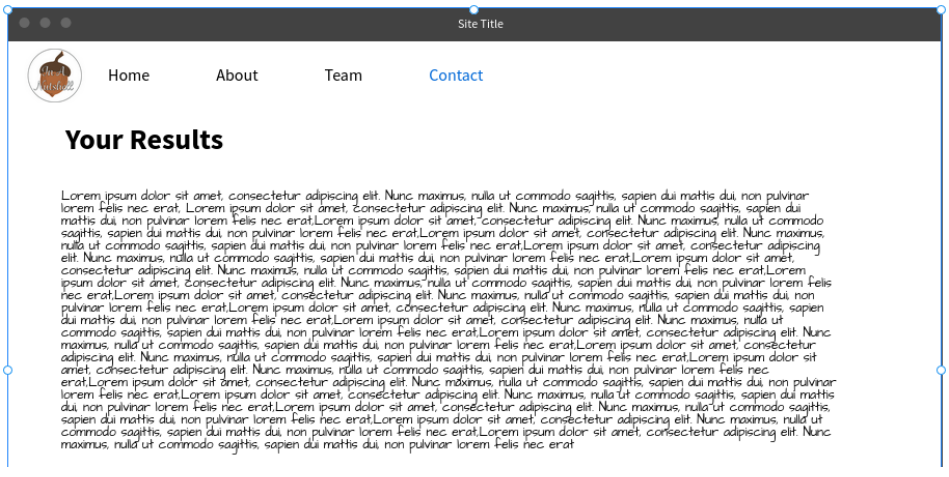


Figure 5.3 Website Result Page

Finally, the user will have a detailed analysis of what the sources of the summary were, how relevant each link was, and the percentage of the summary extracted from each source. The user can also choose to have the important keywords highlighted.

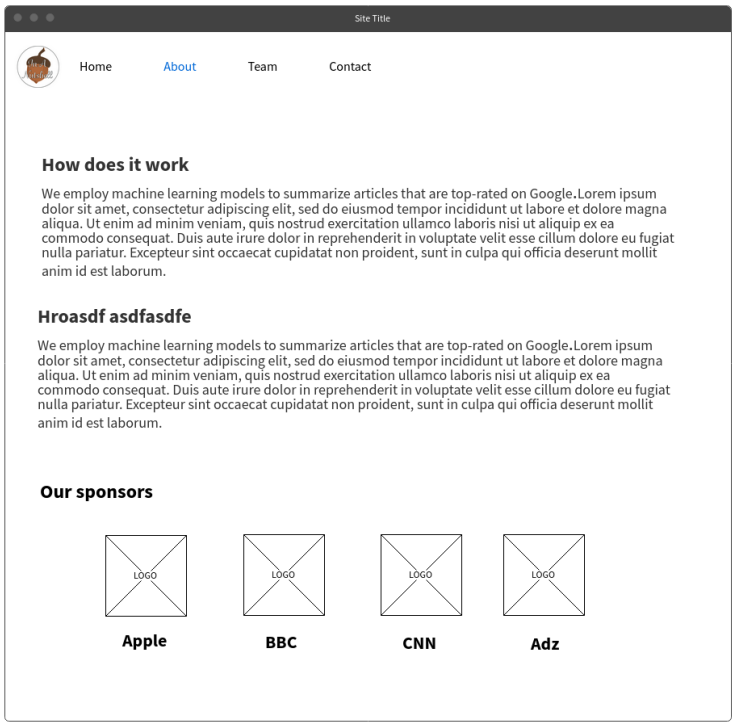


Figure 5.4 Website about page

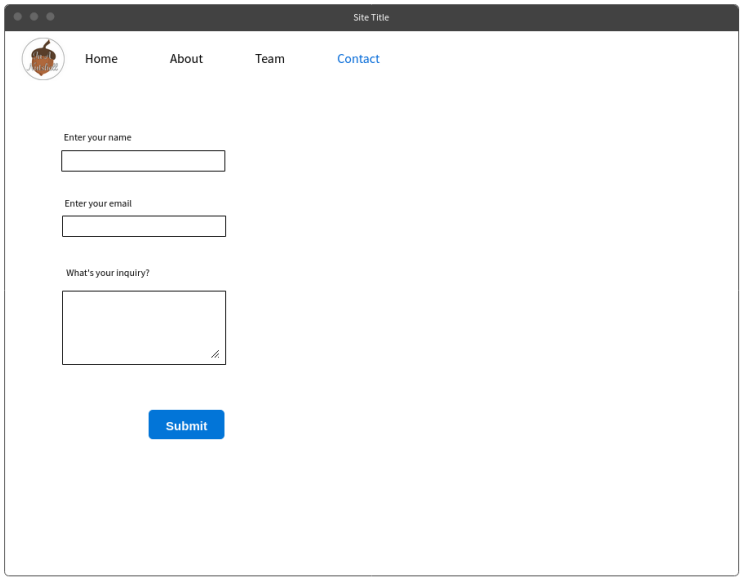


Figure 5.5 Website Contact page

5.4. Back-End

The technologies used are Flask, Pytorch and Google Colab.

5.4.1. Flask

Flask is a high-level Python web framework that makes the process of creating back-ends easy and fast, while still maintaining the quality of the codebase [7]. Flask was chosen because it is easy to learn, efficient, and compatible with scraping libraries which will be used to find articles relevant to the query that the user inputs.

5.4.2. Pytorch

PyTorch is an open-source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing, primarily developed by Facebook's AI Research lab [8]. It is free and open-source software released under the Modified BSD license.

PyTorch can train and run neural networks used for handwritten text classification, image recognition and other tasks.

5.4.3. Google Colaboratory

Colaboratory (also known as *Colab*) is a free Jupyter notebook environment that runs in the cloud and stores its notebooks on Google Drive.

The advantages of using Google Colab are:

- Zero configuration required
- Free Access to GPUs
- Easy sharing and collaboration between different teammates.
- Access to Google's TPUs (Tensor Processing Unit), which is Google's accelerator application specific integrated circuits developed specifically for neural network machine learning.

Google Colab also allows combining executable code and rich text in a single document. So users can add images and graphs that help visualize the output of the code, which makes the development process significantly easier [9].

Chapter 6: Dataset and preprocessing

This chapter gives a detailed explanation on the dataset used, how it was preprocessed and prepared for the model.

6.1. CNN/Dailymail dataset

The CNN/ Dailymail dataset is an English language dataset that has over 300,000 documents collected from the websites of CNN and Dailymail [10]. Initially, this dataset was created to serve the purpose of abstractive question answering. However, it has been modified to support extractive and abstractive summarization.

6.2. Languages

The BCP-47 code for English as generally spoken in the United States is en-US and the BCP-47 code for English as generally spoken in the United Kingdom is en-GB. It is unknown if other varieties of English are represented in the data.

6.3. Dataset Structure

This section will demonstrate how the data set used is structured.

6.3.1 Data Instances

For each instance, there is a string that defines the article, a string for the highlights, and a string for the id.

```
{'id': '0054d6d30dbcad772e20b22771153a2a9cbeaf62',
 'article': '(CNN) -- An American woman died aboard a cruise ship that docked at Rio de Janeiro on Tuesday, the same ship on which 86 passengers previously fell ill, according to the state-run Brazilian news agency, Agencia Brasil. The American tourist died aboard the MS Veendam, owned by cruise operator Holland America. Federal Police told Agencia Brasil that forensic doctors were investigating her death. The ship's doctors told police that the woman was elderly and suffered from diabetes and hypertension, according the agency. The other passengers came down with diarrhea prior to her death during an earlier part of the trip, the ship's doctors said. The Veendam left New York 36 days ago for a South America tour.'
```

'highlights': 'The elderly woman suffered from diabetes and hypertension, ship's doctors say .\nPreviously, 86 passengers had fallen ill on the ship, Agencia Brasil says .'} [10]

The average token count for the articles and the highlights are provided below:

Feature	Mean Token Count
Article	781
Highlights	56

Table 6.1 Dataset average token count [10]

6.3.2 Data Fields

An explanation of the fields of the instance.

- id: a string containing the hexadecimal formatted SHA1 hash of the url where the story was retrieved from
- article: a string containing the body of the news article
- highlights: a string containing the highlight of the article as written by the article author

Each document in the dataset consists of two parts, the article itself, and a highlight section that includes the important sentences extracted from the document.

The document looks like this example:

```
Kharel said he feared the actual number of people killed by the leopard could be higher than 15 , bec

`` It could be the same leopard , '' he said .

Of the 15 victims in Nepal so far , two-thirds are children below the age of 10 . The others are olde
 a common practice in Nepal .

`` No adult male has been killed , '' Kharel said .

All the victims are from villages bordering the dense forests in the district , he said .

After killing its victim , the leopard takes the body away into the forest to eat .

`` In the case of the children it just leaves behind the head , eating everything , but some parts of

The district administration has announced a Rs . 25,000 -LRB- about $ 300 -RRB- reward to anyone who

The local administration has sought to raise public awareness of the dangers of going alone into near
armed police force and local people who have licensed guns to hunt for the animal .

Controlling this particular leopard has been a challenge for the wildlife officials in Kathmandu .

`` We are sending a veterinary doctor to the district to understand the situation , '' Dhakal , the e

The chief district administrator has granted permission for this particular leopard to be killed . No

Leopards are common in the low mountain areas , as compared to the high Himalayas , across the countr

While cases of leopards killing domestic animals are common , and there are sometimes instances of le

@highlight

A 4-year-old boy is the latest victim of a man-eating leopard , a local police chief says

@highlight

He suspects one leopard is behind the deaths of 15 people in the past 15 months

@highlight

A reward has been offered to anyone who captures or kills the man-eating creature

@highlight

Leopards are common in low mountain areas of Nepal but usually eat wild prey like deer
```

Figure 6.1 Document Format [10]

6.4. Preprocessing Dataset

Firstly, a vocabulary dictionary is constructed, assigning every word a numerical value. When the model encounters a word that does not exist in the vocabulary, it is treated as a special word called <unk>, short for unknown.

The next step in the dataset preprocessing pipeline is to add HTML-like tags to identify sentences, along with wrapping the highlight sentences that constitute the summary in summary tags. This facilitates processing by the model later.

After each article has been processed and given identifying tags, the article is written into one batch file. 80% of the articles (287,113) are reserved for training, 15% (13368) for testing, and 5% (11490) for validation.

The batches are then processed and converted into word embeddings (numerical values), which are then fed as input to the model.

Chapter 7: System Implementation

In this section we dive deep into the implementation of the NUTSHELL system, giving detailed description of each module.

7.1. Abstractive summarization model

This section explains how the model uses sequence to sequence with attention and pointer generator to extract the abstractive summary.

7.1.1 Sequence-to-sequence with attention (Base model)

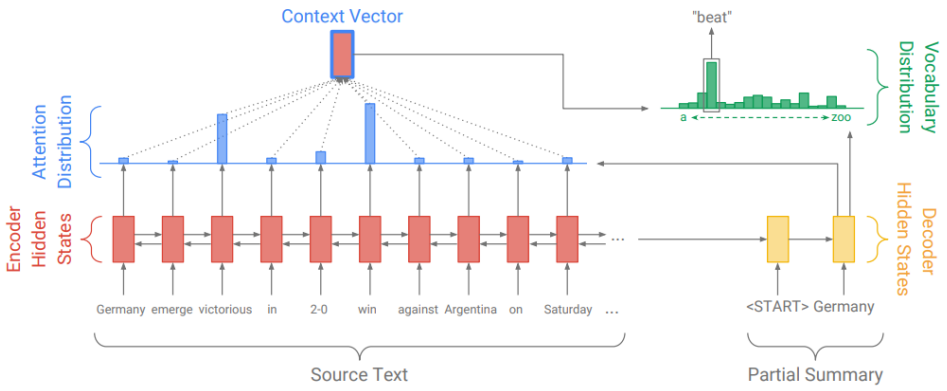


Figure 7.1 Seq2Seq model with attention [11]

A seq2seq model takes a sequence of words as input, and consequently outputs another sequence of words. Hence the name. It serves the purpose of generating an abstractive summary for the input sequence of words.

The model is composed of an encoder and a decoder. The encoder reads a source article, denoted by $x = (x_1 x_2, ..., x_J)$ and transforms it to hidden states $h^e = (h^e_1, h^e_2, ..., h^e_j)$, the hidden states represent the numerical values of the words after a series of forward propagations through the layers of the network.

The decoder takes these hidden states as the context input and outputs a summary $y = (y^1, y2, ..., yT)$. Here, x_i and y_j are one-hot representations of the tokens in the source article and summary, respectively. T is used to represent the number of tokens of the original document and the summary, respectively. To summarize a document, a summary y is to be deduced from a given article x.

Encoder is bi-directional LSTM. The input sequence is encoded as h^e forward and backward. The shortcut notation to indicate encoder input has superscript e.

The decoder is a unidirectional LSTM. The decoder takes the encoded representations of the source article as the input and generates the summary y.
In the Encoder-Decoder base model the encoder vectors are used to initialize hidden and cell states of the LSTM decoder as follows:

$$h_0^d = \tanh\left(W_{\text{e2d}}\left(\overrightarrow{h}_J^e \oplus \overleftarrow{h}_1^e\right) + b_{\text{e2d}}\right), c_0^d = \overrightarrow{c}_J^e \oplus \overleftarrow{c}_1^e$$

Eq. 7.1 Hidden state equation [11]

Superscript d denotes the decoder and $\oplus$ is a concatenation operator. At each decoding step, we first update the hidden state h_t^d conditioned on the previous hidden states and input tokens, where the $h_t^d = LSTM(h_{t-1}^d, E_{y_{t-1}})$. After that we didn't explicitly express the cell states in the input and output of LSTM as only hidden states are passed to other parts of the model.

We calculated the vocabulary distribution of each word in the source document using the following equation:

$$P_{\text{vocab},t} = \text{softmax}(W_{\text{d2v}} h_t^d + b_{\text{d2v}})$$

Eq. 7.2 Vocabulary distribution [11]

Where $P_{vocab,t}$ a vector is whose dimension is the size of the vocabulary **V** and the probability of generating the target token **w** in the vocabulary **V** is denoted as $P_{vocab,t}(\text{w})$

Then in the attention mechanism we compute the attention distribution of the source tokens and then let the decoder know where to attend to produce a target token. In the sequence-to-sequence model Figure 7.1, given all the hidden states of the encoder and the current decoder hidden state h_t^d , the attention distribution α_t^e over the source tokens is calculated as follows:

$$\alpha_{tj}^e = \frac{\exp(s_{tj}^e)}{\sum_{k=1}^{J} \exp(s_{tk}^e)}$$

Eq. 7.3 Attention distribution [11]

Where the alignment score $s_{tj}^{e} = s(h_j^e, h_t^d)$ is obtained by the content-based score function, which has three alternatives:

$$s(h_j^e, h_t^d) = \begin{cases} (h_j^e)^\top h_t^d & \text{dot} \\ (h_j^e)^\top W_{\text{align}} h_t^d & \text{general} \\ (v_{\text{align}})^\top \tanh\left(W_{\text{align}}\left(h_j^e \oplus h_t^d\right) + b_{\text{align}}\right) & \text{concat} \end{cases}$$

Eq. 7.4 Alignment score function [11]

7.1.2 Sequence-to-sequence attentional + pointer generator network model

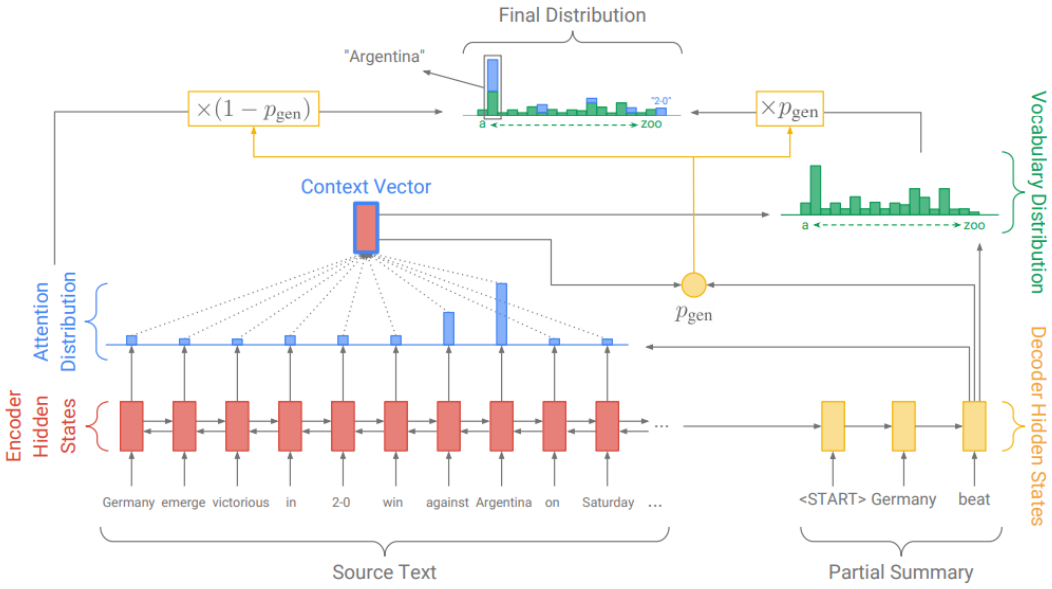


Figure 7.2 Seq2Seq attention and pointer generator network [11]

The pointing mechanism represents a class of approach that generates target tokens by directly copying them from input sequences based on their attention weight. We used the Pointer Softmax into the seq2seq attention base model. The basic architecture of the pointer softmax network is consist of three components:

1. short-list Softmax
2. location Softmax
3. switching network

At decoding step t , a short-list Softmax $P_{vocab,t}$ is calculated by the following equation:

$$P_{\text{vocab},t} = \text{softmax}\left(W_{\text{d2v}}\tilde{h}_t^d + b_{\text{d2v}}\right)$$

Eq. 7.6 Vocabulary distribution [11]

$P_{vocab,t}$ is used to predict target tokens in the vocabulary. The location Softmax gives locations of tokens that will be copied from the source article **x** to the target y_t based on attention weights α_t^e. With these two components, $P_{vocab,t}$ and α_t^e, a switching network is designed to determine whether to predict a token from the vocabulary or copy one from the source article if it is an OOV token. The switching network is a multilayer perceptron (MLP) with a sigmoid activation function, which estimates the probability $P_{gen,t}$ of generating tokens from the vocabulary based on the context vector z_t^e and hidden state h_t^d with the following equation:

$$p_{\text{gen},t} = \sigma(W_{s,z}z_t^e + W_{s,h}h_t^d + b_s)$$

Eq. 7.7 Probability of generating tokens [11]

Where $P_{gen,t}$ is a scalar and $\sigma(a) = \frac{1}{1+\exp(-a)}$ is a sigmoid activation function. The final probability of producing the target token y_t is given by the concatenation of vectors $P_{gen,t}$, $P_{Vocab,t}$ and $(1-P_{gen,t})\,\alpha_t^e$.

7.1.3 Beam search algorithm for decoding seq2seq models

Input: Source article x, beam size B, summary length T, model parameters θ;
Output: B-best summaries;

Beam search algorithm is a compromise between greedy search and exact inference and has been commonly employed in different language generation tasks.
Beam search is a graph search algorithm that generates sequences from left to right by retaining only B top scoring (top-B) sequence-fragments at each decoding step. More formally, we denote decoded top-B sequence fragments, also known as hypotheses at time-step t − 1 as $y_{<t,1}, y_{<t,2}, \ldots, y_{<t,B}$ and their scores as $S_{<t,1}^{bm}, S_{<t,2}^{bm} \ldots\ldots, S_{<t,b}^{bm}$.

For each fragment $y_{<t,b}$, we first calculate $P_\theta(y_{t,b}^{cand} | y_{<t,b}, x)$ which determines B most probable words to expand it. This yields B × B expanded fragments, i.e., new hypotheses, in which only the top-B of them along with their scores are retained for the next decoding step. This procedure will be repeated until the 'EOS' token is generated.

7.2. LexRank

LexRank is an unsupervised text summarizing method based on graph-based sentence centrality rating. The essential premise is that sentences "recommend" to the reader more similar ones. If one sentence is highly similar to many others, it is likely to be a very important sentence. It is an extractive summarization approach.

7.2.1 Sentence Centrality and Centroid-based Summarization

Extractive summarization works by selecting a subset of the original documents' sentences. The sentences that contain more terms from the cluster's centroid are regarded central in centroid-based summarization. The centrality of a sentence refers to how near it is to the cluster's centroid.

7.2.2. Centrality-based Sentence Salience

This definition of centrality needs to be clarified on two aspects. The first issue is determining what constitutes similarity between two sentences. The second problem is determining how to calculate a sentence's overall centrality given its resemblance to other sentences.

7.2.2.1 Computing Centroid Scores

To compute the centroid the following algorithm is used :

Computing Centroid Algorithm [12]

```
input : An array S of n sentences, cosine threshold t
output: An array C of Centroid scores
Hash WordHash;
Array C;
/* compute tf×idf scores for each word */
for i ← 1 to n do
        foreach word w of S[i] do
                WordHash{ w }{ “tfidf” } = WordHash{ w } { “tfidf” } + idf{ w };
        end
end
/* construct the centroid of the cluster */
/* by taking the words that are above the threshold*/
foreach word w of WordHash do
        if WordHash{ w }{ “tfidf” } > t then
                WordHash{ w } { “centroid” } = WordHash{ w } { “tfidf” };
        end
        else
                WordHash{ w } { “centroid” } = 0;
        end
end
/* compute the score for each sentence */
for i ← 1 to n do
        C[i] = 0;
        foreach word w of S[i] do
                C[i] = C[i] + WordHash{ w }{ “centroid” };
        end
end
return C;
```

7.2.2.2 Computing Sentences Similarity

The bag-of-words model is used to represent the number of words in a sentence. The similarity between two sentences is then defined by the cosine between two corresponding vectors.

$$\text{idf-modified-cosine}(x,y) = \frac{\sum_{w \in x,y} \text{tf}_{w,x}\text{tf}_{w,y}(\text{idf}_w)^2}{\sqrt{\sum_{x_i \in x}(\text{tf}_{x_i,x}\text{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y}(\text{tf}_{y_i,y}\text{idf}_{y_i})^2}}$$

Eq. 7.8 idf-modified-cosine [12]

tfw,s is the number of occurrences of the word w in the sentences

7.2.3. Degree Centrality

Because the documents are all about the same subject, many of the sentences in a group of related documents are likely to be similar. Each sentence in the cluster is a node, and significantly similar sentences are connected to one another in a (undirected) graph. The degree of the corresponding node in the similarity graph is used to determine the degree centrality of a sentence.

7.2.4. Eigenvector Centrality and LexRank

Degree centrality may have a negative effect in the quality of summaries where several unwanted sentences vote for each other and raise their centrality. The idea can be applied to extractive summarization as well. A straightforward way of formulating this idea is to consider every node having a centrality value and distributing this centrality to its neighbors. This formulation can be expressed by the equation:

$$p(u) = \sum_{v \in adj[u]} \frac{p(v)}{deg(v)}$$

Eq 7.9 Degree Centrality Equation [12]

Where p(u) is the centrality of node u, adj[u] is the set of nodes that are adjacent to u, and deg(v) is the degree of the node v. Equivalently, we can write the above equation in the matrix notation as:

$$p(u) = \frac{d}{N} + (1 - d) \sum_{v \in adj[u]} \frac{p(v)}{deg(v)}$$

Eq. 7.10 Modified Degree Centrality Equation [12]

where p(u) is the centrality of node u, adj[u] is the set of nodes that are adjacent to u, and deg(v) is the degree of the node v, N is the total number of nodes in the graph, and d is a "damping factor", which is typically chosen in the interval [0.1, 0.2]

The power method is a simple iterative algorithm based on the convergence property of Markov chains. A uniform distribution is used as the starting point for the algorithm. The eigenvector is updated at each iteration by multiplying the eigenvector with the stochastic matrix's transpose. The algorithm is guaranteed to terminate because the Markov chain is irreducible and aperiodic.

Power Method Algorithm [12]

input : A stochastic, irreducible and aperiodic matrix M
input : matrix size N, error tolerance ϙ
output: eigenvector p

p_0= 1/N 1;
t=0;
repeat
t=t+1;
pt = $M^T p_{t-5}$ 1;
δ = $||p_t - p_{t-1}||$;
until δ < ϙ;
return pt;

7.2.5. LexRank Algorithm [12]

MInputAn array S of n sentences, cosine threshold t output: An array 1 L of LexRank scores
Array CosineMatrix[n][n];
Array Degree[n];
Array L[n];
for i ← 1 to n **do**
for j ← 1 to n do
CosineMatrix[i][j] = idf-modified-cosine(S[i],S[j]);
if CosineMatrix[i][j] > t then
CosineMatrix[i][j] = 1;
Degree[i] + +;
end
else
CosineMatrix[i][j] = 0;
end
end
end
for i ← 1 to n **do**
for j ← 1 to n **do**
CosineMatrix[i][j] = CosineMatrix[i][j]/Degree[i];
end
end
L = PowerMethod(CosineMatrix,n,ϙ);
return L;

7.2.6. LexRank Example

SNo	ID	Text
1	d1s1	Iraqi Vice President Taha Yassin Ramadan announced today, Sunday, that Iraq refuses to back down from its decision to stop cooperating with disarmament inspectors before its demands are met.
2	d2s1	Iraqi Vice president Taha Yassin Ramadan announced today, Thursday, that Iraq rejects cooperating with the United Nations except on the issue of lifting the blockade imposed upon it since the year 1990.
3	d2s2	Ramadan told reporters in Baghdad that "Iraq cannot deal positively with whoever represents the Security Council unless there was a clear stance on the issue of lifting the blockade off of it.
4	d2s3	Baghdad had decided late last October to completely cease cooperating with the inspectors of the United Nations Special Commission (UNSCOM), in charge of disarming Iraq's weapons, and whose work became very limited since the fifth of August, and announced it will not resume its cooperation with the Commission even if it were subjected to a military operation.
5	d3s1	The Russian Foreign Minister, Igor Ivanov, warned today, Wednesday against using force against Iraq, which will destroy, according to him, seven years of difficult diplomatic work and will complicate the regional situation in the area.
6	d3s2	Ivanov contended that carrying out air strikes against Iraq, who refuses to cooperate with the United Nations inspectors, "will end the tremendous work achieved by the international group during the past seven years and will complicate the situation in the region."
7	d3s3	Nevertheless, Ivanov stressed that Baghdad must resume working with the Special Commission in charge of disarming the Iraqi weapons of mass destruction (UNSCOM).
8	d4s1	The Special Representative of the United Nations Secretary-General in Baghdad, Prakash Shah, announced today, Wednesday, after meeting with the Iraqi Deputy Prime Minister Tariq Aziz, that Iraq refuses to back down from its decision to cut off cooperation with the disarmament inspectors.
9	d5s1	British Prime Minister Tony Blair said today, Sunday, that the crisis between the international community and Iraq "did not end" and that Britain is still "ready, prepared, and able to strike Iraq."
10	d5s2	In a gathering with the press held at the Prime Minister's office, Blair contended that the crisis with Iraq "will not end until Iraq has absolutely and unconditionally respected its commitments" towards the United Nations.
11	d5s3	A spokesman for Tony Blair had indicated that the British Prime Minister gave permission to British Air Force Tornado planes stationed in Kuwait to join the aerial bombardment against Iraq.

Figure 7.3 LexRank Test Sentences [12]

The above sentences are taken from five different documents from the BBC, and are used to show how LexRank works.

	1	2	3	4	5	6	7	8	9	10	11
1	1.00	0.45	0.02	0.17	0.03	0.22	0.03	0.28	0.06	0.06	0.00
2	0.45	1.00	0.16	0.27	0.03	0.19	0.03	0.21	0.03	0.15	0.00
3	0.02	0.16	1.00	0.03	0.00	0.01	0.03	0.04	0.00	0.01	0.00
4	0.17	0.27	0.03	1.00	0.01	0.16	0.28	0.17	0.00	0.09	0.01
5	0.03	0.03	0.00	0.01	1.00	0.29	0.05	0.15	0.20	0.04	0.18
6	0.22	0.19	0.01	0.16	0.29	1.00	0.05	0.29	0.04	0.20	0.03
7	0.03	0.03	0.03	0.28	0.05	0.05	1.00	0.06	0.00	0.00	0.01
8	0.28	0.21	0.04	0.17	0.15	0.29	0.06	1.00	0.25	0.20	0.17
9	0.06	0.03	0.00	0.00	0.20	0.04	0.00	0.25	1.00	0.26	0.38
10	0.06	0.15	0.01	0.09	0.04	0.20	0.00	0.20	0.26	1.00	0.12
11	0.00	0.00	0.00	0.01	0.18	0.03	0.01	0.17	0.38	0.12	1.00

Table 7.1 Similarity Matrix [12]

Computing the similarity matrix between all the 11 sentences produces the matrix form shown in Table 7.1 where each sentence has a similarity with every other sentence. The above similarity matrix is represented in the following weighted graph where each node is a sentence and the weighted edges are the similarity scores.

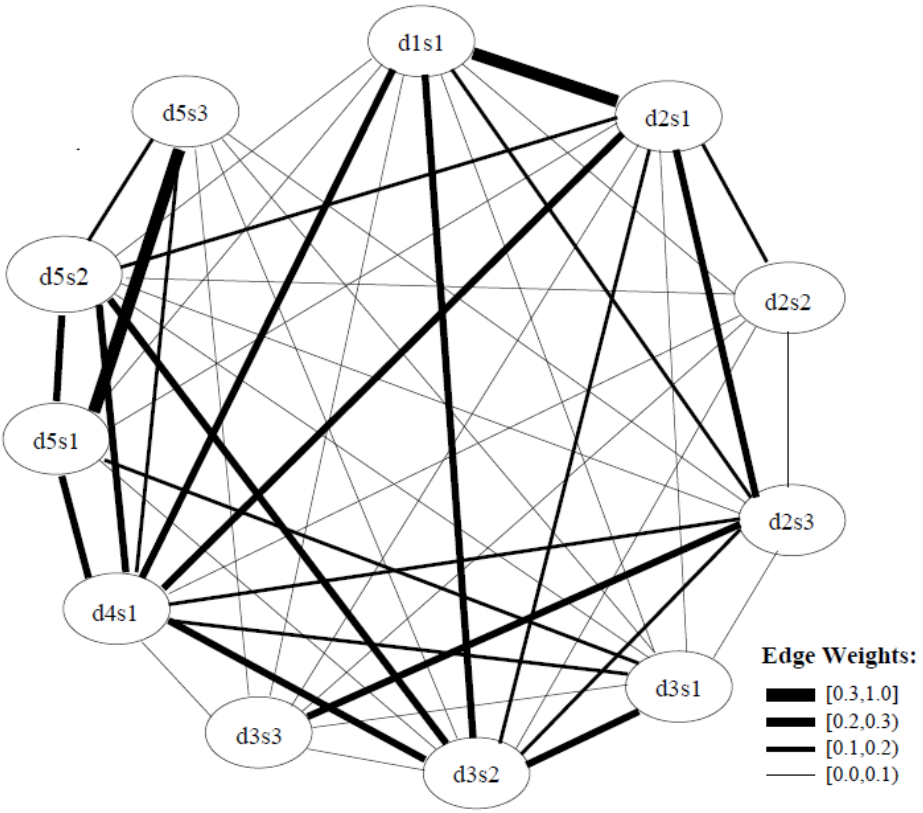


Figure 7.4 Weighted Graph Representation [12]

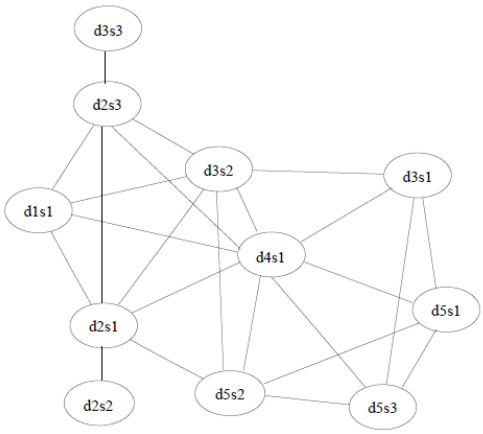


Figure 7.5 Similarity Graph Threshold 0.1 [12]

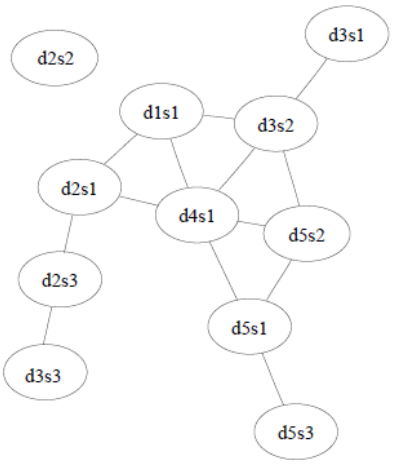


Figure 7.6 Similarity Graph Threshold 0.2 [10]

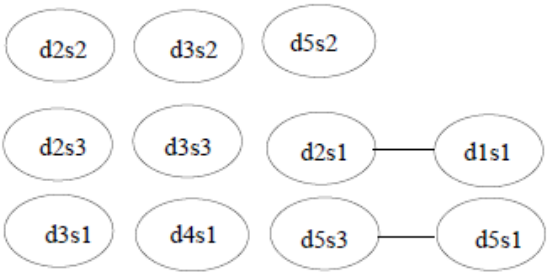


Figure 7.7 Similarity Graph Threshold 0.3 [12]

The above Figures (7.5-7.7) show the sentences' relations after applying threshold to identify important and non-important sentences and as threshold increases the sentences that are more related are only left in a group.

ID	LR (0.1)	LR (0.2)	LR (0.3)	Centroid
d1s1	0.6007	0.6944	1.0000	0.7209
d2s1	0.8466	0.7317	1.0000	0.7249
d2s2	0.3491	0.6773	1.0000	0.1356
d2s3	0.7520	0.6550	1.0000	0.5694
d3s1	0.5907	0.4344	1.0000	0.6331
d3s2	0.7993	0.8718	1.0000	0.7972
d3s3	0.3548	0.4993	1.0000	0.3328
d4s1	1.0000	1.0000	1.0000	0.9414
d5s1	0.5921	0.7399	1.0000	0.9580
d5s2	0.6910	0.6967	1.0000	1.0000
d5s3	0.5921	0.4501	1.0000	0.7902

Table 7.2 LexRank Scores [12]

The above table in Table 7.2 shows the final LexRank Score of each sentence with thresholds 0.1, 0.2, 0.3 respectively; these scores are used to order sentences from most important to least important ones.

7.3 Scraper

This module is concerned with finding news articles from the internet relating to the user's query. The scraper uses Google News' rss feed integration in order to retrieve formatted results, rather than manually parsing the HTML response. Using this method the Google News server returns an XML response, which is then parsed accordingly.

This method has a lot of advantages, starting from the structured XML response, finding articles published in a certain time frame, searching for articles from specific sources, finding articles relating to a specific geo location, and much more.

It is also worth mentioning that the Google News RSS feed response is not a full-fledged API, and has a limit to the number of requests per IP address.

We ran into those aforementioned limitations when testing our system, so we decided to use a python library called "pygooglenews", It also uses Google News' RSS feed. The only difference is that it allows us to use proxy servers to simulate multiple IP addresses, to avoid us hitting the limit.

Parsing the article text from websites was a tricky situation. Since every website uses its own formatting. So it was essentially impossible to extract only the article's text, but this was left to the machine learning model to discard any irrelevant words.

A python library "BeautifulSoup" was used for manipulating and filtering of the HTML document of the website. It was used to remove all unwanted tags like labels, headers and buttons.

7.4. Web Application

The website backend was implemented using Flask and the frontend was built using NextJS, a variant of ReactJS. When a request is sent to the backend, it uses the scraper to find news articles with relevant queries and then after generating a final summary it sends the response to the user.

7.5. Challenges

In this section we discuss some of the challenges we faced while implementing this system.

7.5.1. Finding a proper dataset

It took around 3 weeks to finally settle on this dataset. We tried to obtain the DUC dataset at first, but it proved difficult to acquire, since we would have had to sign a lot of documents and possibly pay a fee to obtain it. This process would have consumed a lot of time and money. We went through a few more datasets until we settled on CNN/Dailymail.

7.5.2. Model Training

Due to the limited capabilities of our hardware, the graphics cards we had would run out of memory. Access to better hardware was limited. Hence, we trained the model on Google's graphics cards, using Google Colab Pro service.

7.5.3. Google Colab Pro

Colab Pro does not offer uninterrupted service, and can disconnect if the browser window is left for a long time. Therefore, we had to develop some code that runs in the background and does something trivial to keep the session active (move a cursor).

7.5.4. Training Time

The model would often take almost 5 days to train on the CNN/Dailymail dataset. A lot of time has been wasted because the training was interrupted due to the issue described in the previous section. This wasted around 2 weeks and stalled further work on the system.

7.5.5. Web Scraping

News websites do not standardize their content. Therefore, it was very challenging to tune the scraper to find the article text in the right tag, since some websites place article text in <p> tags, others in <article> tag and so on. And even after thorough tuning, upon extensive testing, some articles did not get parsed correctly.

7.5.6. Covid-19 restrictions

Due to Covid-19 fears, we could not meet up in person, and we had to conduct our meetings and working sessions online to avoid physical contact. This severely impeded our progress, due to internet connection issues. In addition, debugging sessions became one sided since it was complicated to debug code in groups.

Chapter 8: System Testing and Results

This chapter will be divided into three main subsections where the first two sections will assess the first two modules separately using ROGUE R1 scores that will be generated upon comparing the output with a previously supplied golden summary that is written by professionals, and the last subsection will assess the final output of the overall system of the two modules, comparing it with other systems. As those other systems are single document summarizing systems, we are using our comments and manual inspection of the result due to the lack of the golden summary for multi document texts.

8.1. Testing the Abstractive summarizer

The section will present and discuss the ROGUE scores of the abstractive summaries that results from our first stage of the system.

Source	R-1	Training Dataset	Testing Dataset
Paper score	39.36	DUC	DUC
Nutshell score	33.48	CNN/DailyMail	DUC

Table 8.1 R-1 Scores for the abstractive summarizer using DUC [13]

Table 8.1 above shows the results for our abstractive summarization module, when tested on a sample of the DUC dataset. It can be seen that there is a noticeable difference (about 6%) in the scores. This can be explained by the fact that the paper's model was trained and tested using the DUC dataset itself, while our implementation was trained on the cnn/dailymail dataset. This obviously gives the paper's model an advantage over Nutshell. Therefore we have decided to repeat the test with the paper's model [14], a pre trained model on the DUC dataset, using our testing set from the cnn/dailymail and found the following:

Source	R-1	Training Dataset	Testing Dataset
Paper score	28.76	DUC	CNN/DailyMail
Nutshell score	38.13	CNN/DailyMail	CNN/DailyMail

Table 8.2 R-1 Scores for the abstractive summarizer using cnn/dailymail [13]

The results in Table 8.2 show that when applying the concept of differentiating between training and testing data, the paper's model [14] which is tested on our blind test cases achieved poorly with a difference of almost 10%. Thus, our model is superior to the pre trained model by showing an improvement of 10% in blind test cases.

8.2. Testing the Extractive summarizer

This section presents all the ROGUE scores of the abstractive summaries that result from the second stage of the system.

Source	R-1
Followed Paper Score	44.43
Nutshell Score	44.52

Table 8.3 R-1 Scores for the extractive summarizer [12]

Table 8.3 above shows that the scores of the extractive summarizer, when tested on a sample of the DUC dataset, were almost identical in both the paper implementation and our implementation. That is also expected since the LexRank algorithm works in both cases because it processes the sentences separately, not the piece of text as a bulk.

8.3. Testing System

This section assesses the final results of the overall system. Interpretations and comments on the results are given at the end.

8.3.1. Test Case 1

This test case was done using the following scraped articles about the keyword “corona-virus”:

1. https://www.abc.net.au/news/health/2021-07-06/what-we-know-about-the-lambda-variant/100267978
2. https://edition.cnn.com/2021/06/29/health/us-coronavirus-tuesday/index.html
3. https://www.bbc.com/news/world-middle-east-57594155
4. https://www.nytimes.com/2021/06/25/opinion/coronavirus-lab.html
5. https://www.channelnewsasia.com/news/world/egypt-eases-guest-limits-in-hospitality-sector-as-covid-19-15155542

The total count of words in these articles were 3782, which is later condensed into a summary of 450 words.

8.3.1.1. The abstractive summary

Figure 8.1 below shows a screenshot of a single summary of the multiple summaries generated by the seq2seq model. The size of the summary is 127 words, whereas the size of the input article was 781 words. As we can notice some post processing is still needed and will be done in the following module (LexRank). This will be shown in the following subsection.

```
"summary0": "tourists take lunch on a mountain restaurant amid
the coronavirus disease (covid-19) pandemic in the red sea resort
of sharm el-sheikh,south of cairo, egypt february 4 (reuters) . </s>
<s> egypt's cabinet on sunday eased guest limits for hotels, restaurants,
cinemas and theatres to 70% of their capacity from 50 percent at present
as coronavirus infections slow, a cabinet statement said. egypt has been
gradually easing pandemic restrictions since june 1. official figures
showed 181 new covid-19 cases were recorded on saturday, with 27 deaths
from the disease. reporting by moamen said attalah, writing by alaa swilamm
editing . </s> <s> moamen said attalah, said attalah, writing by alaa swilamm
editing by alaa swilamm editing by catherine evans our standards:\\s editing .
</s> <stop>\n",
```

Figure 8.1 Test Case 1 generated abstractive summary

8.3.1.2 The Final summary

staff 2 min read cairo, june 24 (reuters) - egypt will allow travellers who have taken full doses of approved novel coronavirus vaccines to enter without taking a pcr test, the health ministry said on thursday. travellers must present qr-coded certificates that they have received their full doses of one of six covid-19 vaccines approved by egypt . the who at least two weeks before their arrival. those from countries impacted by coronavirus variants will be subject to a rapid test upon arrival, . all non-vaccinated travellers must present a pcr test. on thursday, egypt reported 466 new coronavirus cases, bringing its total to 278,761. however, officials and experts say the real number of infections is far higher . tourists take lunch on a mountain restaurant amid the coronavirus disease (covid-19) pandemic in the red sea resort of sharm el-sheikh, south of cairo, egypt february 4 (reuters) . egypt's cabinet on sunday eased guest limits for hotels, restaurants, cinemas and theatres to 70% of their capacity from 50 percent at present as coronavirus infections slow, a cabinet statement said. egypt has been gradually easing pandemic restrictions since june 1. official figures showed 181 new covid-19 cases were recorded on saturday, with 27 deaths from the disease. reporting by moamen said attalah, writing by alaa swilamm editing . moamen said attalah, said attalah, writing by alaa swilamm editing by alaa swilamm editing by catherine evans our standards:\s editing .

Figure 8.2 Test Case 1 Generated final summary

The summary above after reading the articles from the sites actually did get the main points about the topic and concatenated the summaries into a single piece of text that gives all the needed information with about one sixth of the length of the original articles. The summary was cut short at the end because of the max length that was set previously by us. But with some minor tweaks to make the max length a little dynamic the issue can be solved. Also, we can notice that no redundancy is found in the summary which shows that LexRank has selected the correct summaries.

8.3.2. Test Case 2

This test case was done using the following scraped articles about the keyword “Trump”:

1. https://www.dexerto.com/apex-legends/apex-legends-leak-hints-worlds-edge-map-update-coming-in-season-10-1607167/
2. https://www.pcgamer.com/apex-legends-hacked-to-protest-titanfalls-server-situation/
3. https://attackofthefanboy.com/news/apex-legends-and-fifa-21-online-servers-keep-going-down-today-july-4/
4. https://www.dexerto.com/apex-legends/apex-legends-players-want-simple-fortnite-feature-to-make-finding-games-easier-1604961/
5. https://www.givemesport.com/1714078-apex-legends-genesis-event-new-valkyrie-skin-revealed
6. https://piunikaweb.com/2021/07/06/apex-legends-not-loading-after-genesis-event-update-gets-acknowledged/

The total count of words in these articles were 3457, which is later condensed into a summary of 650 words.

8.3.2.1. The abstractive summary

Figure 8.3 below shows a screenshot of a single summary of the multiple summaries generated by the seq2seq model. The size of the summary is 116 words, whereas the size of the input article was 625 words.


```
"summary1": "game news update: apex legends and fifa 21 online servers
keep going down today (july 4) servers are down july 4th, 2021 by it
appears the online servers for some ea games are being inconsistent
today. the games affected today . </s> <s> for fifa 21, it looks like
people playing the ps4 version cannot play the fut mode of the game.
the servers aren\u2019t allowing people to connect to the game today.
you can read an announcement below from the . </s> <s> for fifa 21,
it looks like people playing the ps4 version cannot play the fut mode
of the game. the servers aren\u2019t allowing people to connect to the
game today. you can read an announcement below from the . </s> <s>\n",
```


Figure 8.3 Test Case 2 generated abstractive summary

8.3.1.2 The Final summary

the season 10 legend is still in the midst of the current ninth season of content with legacy, but that does not stop us from finding out more about what is to come in the next season. it looks like we might have some information about the season 10 legend . this new limited-time event included releasing new content and gameplay changes for players to check out. of course, with the release of a new voice lines and other files . apex legends data miner humansas on twitter, they were able to discover some rather intriguing apex legends seer voice lines that may point to what's to come for the game. the data miner . the data miner posted game news update: apex legends and fifa 21 online servers keep going down today (july 4) servers are down july 4th, 2021 by it appears the online servers for some ea games are being inconsistent today. the games affected today . for fifa 21, it looks like people playing the ps4 version cannot play the fut mode of the game. the servers aren't allowing people to connect to the game today. you can read an announcement below from the . for fifa 21, it looks like people playing the ps4 version cannot play the fut mode of the game. the servers aren't allowing people to connect to the game today. you can read an announcement below from the . pc players. the high cpu usage rates have caused the game to have major problems, which will likely end in a game crash. cpu usage rates for some players have been reaching abnormal highs of 90-100% . developers issue a fix: related articles there has been a thread on the ea comment boards that players are talking about on this issue, and some players are suggesting that it does fix the problem, . it is not guaranteed that it will completely get rid of the bug nor that it will work. community method as of lately: users on reddit are also speculating that there is "spaghetti code" - unmaintainable strings of code - that is launching in the lobby which is

Figure 8.4 Test Case 2 Generated final summary

Same as test case 1, this test case have resulted in a meaningful summary but with the same issues found in the final summary of the first test case in Figure 8.2., Nutshell in it is current version is not perfect but with some minor tweaks, given enough time, all these issues can be handled .

8.4 Bad test case scenario

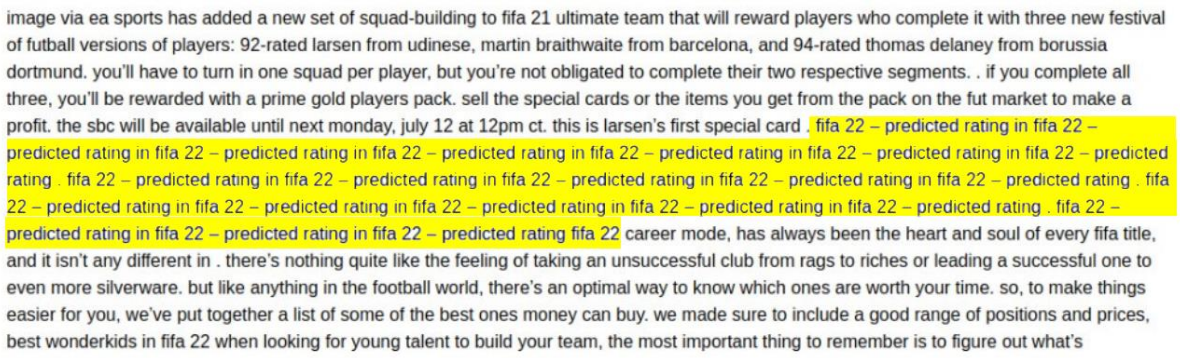


image via ea sports has added a new set of squad-building to fifa 21 ultimate team that will reward players who complete it with three new festival of futball versions of players: 92-rated larsen from udinese, martin braithwaite from barcelona, and 94-rated thomas delaney from borussia dortmund. you'll have to turn in one squad per player, but you're not obligated to complete their two respective segments. . if you complete all three, you'll be rewarded with a prime gold players pack. sell the special cards or the items you get from the pack on the fut market to make a profit. the sbc will be available until next monday, july 12 at 12pm ct. this is larsen's first special card . fifa 22 – predicted rating in fifa 22 – predicted rating in fifa 22 – predicted rating in fifa 22 – predicted rating in fifa 22 – predicted rating in fifa 22 – predicted rating in fifa 22 – predicted rating . fifa 22 – predicted rating in fifa 22 – predicted rating in fifa 22 – predicted rating in fifa 22 – predicted rating in fifa 22 – predicted rating in fifa 22 – predicted rating . fifa 22 – predicted rating in fifa 22 – predicted rating in fifa 22 – predicted rating in fifa 22 – predicted rating in fifa 22 – predicted rating in fifa 22 – predicted rating . fifa 22 – predicted rating in fifa 22 – predicted rating in fifa 22 – predicted rating fifa 22 career mode, has always been the heart and soul of every fifa title, and it isn't any different in . there's nothing quite like the feeling of taking an unsuccessful club from rags to riches or leading a successful one to even more silverware. but like anything in the football world, there's an optimal way to know which ones are worth your time. so, to make things easier for you, we've put together a list of some of the best ones money can buy. we made sure to include a good range of positions and prices, best wonderkids in fifa 22 when looking for young talent to build your team, the most important thing to remember is to figure out what's

Figure 8.5 Bad test case scenario

In this test case queries "Fifa" the summary used the following links:

- https://dotesports.com/fifa/news/how-to-complete-denmark-nation-players-sbc-in-fifa-21-ultimate-team
- https://www.dexerto.com/fifa/best-fifa-22-young-players-to-sign-on-career-mode-1606674/

Since the model has a maximum length to generate and the scraper did not provide enough articles to generate from, the model started embedding parts of the same test to fill the remaining text. This is because the abstractive summarizer is actually generating text with the given parameters and information about the data supplied. To solve this issue we started excluding the articles that had no enough sequence for the seq2seq model to start generating meaningful summaries therefore the result would look like the summary shown in Figure 8.6.

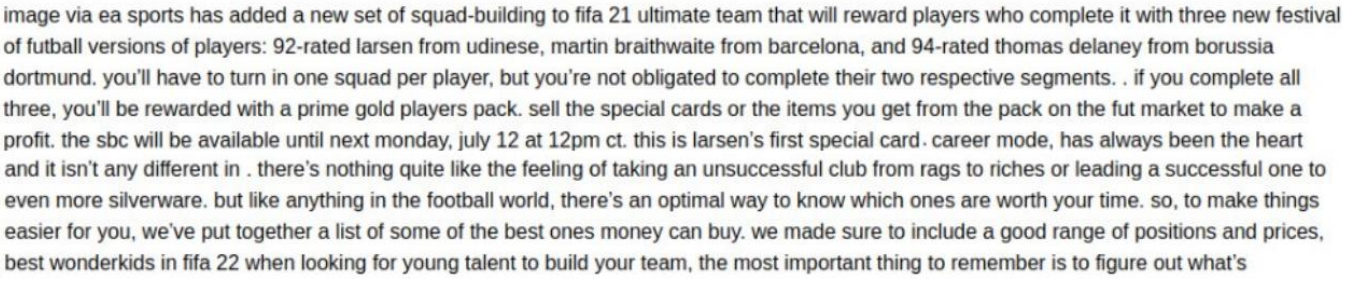


image via ea sports has added a new set of squad-building to fifa 21 ultimate team that will reward players who complete it with three new festival of futball versions of players: 92-rated larsen from udinese, martin braithwaite from barcelona, and 94-rated thomas delaney from borussia dortmund. you'll have to turn in one squad per player, but you're not obligated to complete their two respective segments. . if you complete all three, you'll be rewarded with a prime gold players pack. sell the special cards or the items you get from the pack on the fut market to make a profit. the sbc will be available until next monday, july 12 at 12pm ct. this is larsen's first special card . career mode, has always been the heart and it isn't any different in . there's nothing quite like the feeling of taking an unsuccessful club from rags to riches or leading a successful one to even more silverware. but like anything in the football world, there's an optimal way to know which ones are worth your time. so, to make things easier for you, we've put together a list of some of the best ones money can buy. we made sure to include a good range of positions and prices, best wonderkids in fifa 22 when looking for young talent to build your team, the most important thing to remember is to figure out what's

Figure 8.6 Modified Bad test case scenario

Chapter 9: Conclusion and Future Work

When we chose the topic of multi document summarization as the core for our graduation system, the Nutshell we were aware of its difficulty and the challenges involved. Moreover, document summarization is still in the preliminary phases of development in the community, thus the data to start with and complete upon is so rare and in many cases protected such as Facebook's summarization algorithms which are considered the best nowadays. However, we still selected it because we knew its importance and we were eager to learn more dig into the topic to finalize our academic study with an emanant work that would be of use to the community.

9.1. Gained Experience

The system has taught us a lot, from the research phase to implementation phase. We have learned how to investigate a certain research area and extract needed info from scholarly articles and papers, we have learned how to assess the quality of a research by the way it is drafted.

The implementation phase has been quite difficult and demanding, but rewarding at the same time, we have faced a lot of obstacles that prevented us from getting the expected results, and while overcoming these obstacles we have learned a lot about machine learning, and different technologies that we believe will be of great help in our future projects.

9.2. Conclusion

Text summarization, especially multi document, is not yet in a phase that may allow a full dependency by users but so was language translation and speech recognition/synthesis at some point of time, and now they are used by a lot of people and have almost perfect results. We believe that if the multi document summarization is supplied with enough data and enough research it will become as good as human's ability to summarize but with better performance in time thus saving humans a lot of their precious time.

From the data provided throughout this paper, and from the testing results. We can conclude that our system was successful in implementing abstractive and extractive summarization systems, this is evident from the comparison results with the followed paper system. But this is not to say that our system is perfect, as it has its drawbacks that were showcased in section 8.4, and 9.3.

9.3. Future Work

This paper has dealt with a specific type of text summarization problems which deals with long pieces of text. This is considered to be the hardest subfield of text summarization.

Thus, we plan to improve the performance and quality of the system by adding more filtration and preprocessing to the input text and also some post processing to the output summary.

These add-ons should drastically improve the quality of the output and help in overcoming any of the shortcomings of the current version. Such as, Seq2Seq model filling the summary with redundant or seemingly random words due to there not being enough articles on a certain topic, or when the output summary gets cut short due to the provided summary size limit.

The filtration and preprocessing that could be applied to the current version are removal of unrelated date which is not relevant to the topic of the majority of documents. This can be achieved using clustering the text based on the topic which is actually another field that is being investigated and researched a lot nowadays.

During our research we saw some outputs that were hugely affected by these outliers, but could not implement them due to time limitations. Postprocessing would vary from removing incomplete sentences and fine tuning the output thus creating a more readable version of the summary.

9.4. Work Division

Samy Saeed	**In Model:** ● Research paper about abstractive text summarization. ● (Bi-directional LSTM) Encoder ● (Uni-directional LSTM) Decoder ● Attention mechanism
Kamel Mohsen	**In Model:** ● Research paper about abstractive text summarization. ● (Bi-directional GRU) Encoder ● (Uni-directional GRU) Decoder ● Beam search mechanism ● Pointer generator mechanism
Ziad AbdelHamid	**In LexRank:** ● Research ● calculating centrality scores ● ranks sentences ● calculates idf and idf cosine matrices **In Backend:** ● implemented the scraper
Ahmad Nader	**In LexRank:** ● Research ● Implemented Utility functions, implementing power method for calculating eigenvectors, in addition to functions calculating Markov matrices and stationary distribution. Also, additional helper functions for tokenizing words and removing punctuation **In Frontend:** ● Implemented website frontend using NextJS

Table 9.1 Workload distribution

References

[1] Kumari, Anita & Shashi, M.. (2019). Deep Learning Architecture for Multi-Document Summarization as a cascade of Abstractive and Extractive Summarization approaches. https://www.ijcseonline.org/pdf_paper_view.php?paper_id=3945&159-IJCSE-06050.pdf

[2] Mac, R. (2020, December 23). Facebook Is Developing A Tool To Summarize News Articles.BuzzFeedNews.https://www.buzzfeednews.com/article/ryanmac/facebook-news-article-summary-tools-brain-reader

[3] SummarizeBot - Get to Know More by Reading Less! (2019). SummarizeBot. https://www.summarizebot.com/

[4] R. (2020). Resoomer | Summarizer to make an automatic text summary online. Resoomer. https://resoomer.com/en/

[5] Amazon Web Services (AWS) - Cloud Computing Services. (2013). Amazon Web Services, Inc. https://aws.amazon.com/

[6] Roul, Rajendra & Sahoo, Jajati & Goel, Rohan. (2017). Deep Learning in the Domain of Multi-Document Text Summarization. 575-581. 10.1007/978-3-319-69900-4_73.

[7] Welcome to Flask — Flask Documentation (2.0.x). (2010). Flask. https://flask.palletsprojects.com/en/2.0.x/

[8] PyTorch. (2010). PyTorch. https://pytorch.org/

[9] Google Colaboratory. (2015). Google Colaboratory. https://colab.research.google.com/

[10] cnn_dailymail · Datasets at Hugging Face. (2021, February 2). CNN/Dailymail Dataset. https://huggingface.co/datasets/cnn_dailymail

[11] Shi, T. (2018, December 5). Neural Abstractive Text Summarization with Sequence-to-Sequence Models. ArXiv.Org. https://arxiv.org/abs/1812.02303

[12] Erkan, G. (2004). LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. ArXiv.Org. https://arxiv.org/abs/1109.2128

[13]See, A. (2017, April 14). *Get To The Point: Summarization with Pointer-Generator Networks*. ArXiv.Org. https://arxiv.org/abs/1704.04368

[14] T. (2017). thunlp/TensorFlow-Summarization. GitHub. https://github.com/thunlp/TensorFlow-Summarization

WEE Water Engineering and Environment

STE Structural Engineering

PPC Petro Chemical Engineering

MDE Mechanical Design Engineering

CEM Construction Engineering and Management

CCE Communication and Computer Engineering

AET Architectural Engineering and Technology

Sponsor