CANDLE APP

Graduation Project v2

June 24, 2021

PROJECT ADVISORS:

- DR/MOHAMED ATTA KHAFAGY
- DR/SAFYNAZ ABDELFATTAH

BY:

- MUHAMMAD REFAAY ALALLAMY
- ABDELRHMAN AHMED ALI
- AHMED MOHAMMED HASSAAN
- MAHMOUD RAGAB HEZAYEN
- TAHA MAHMOUD FARG

Contents

Abstract

Reading has always instilled ideas in the heads of creative people and developed into new inventions, from the "Earth" program in Neil Stephenson's novel "The Avalanche", from which the "Google Earth" application was inspired.

Of course, the importance of reading is not the subject of the document. We all understand the importance of reading, and our project is very important for everyone who reads because it will simply change the concept of regular reading and will help extract important information in more than one form and will help to understand the texts better and thus get many inspirational ideas .

We provide three services through our App:
1-Automatically generating Word cloud a word cloud is a collection, or cluster, of words depicted in different sizes. The bigger and bolder the word appears, the more often it's mentioned within a given text and the more important it is.
2-Automatically generating summary is a shortened version of a text.
Given long documents to read, our natural preference is to not read, or at least, to scan just the main points. So having a summary would always be great to save us time ⏳ and brain processing power.
3-Automatically generating mind maps out of pure text and that requires many stages of text processing

In addition, in order to be able to achieve all of this we are focusing on Natural Language Processing or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text by software and of course, we will using python.

1- Introduction

When a Falcon rocket blasted into space in February 2018, it was carrying an unexpected payload. Unlike the red "Tesla Roadstar" car and the doll in the astronaut suit behind the steering wheel, SpaceX founder Elon Musk put in the front glove box of the car the "foundation" series of novels by Isaac Asimov recorded on an optical disk. Its events took place in a future era after nearly 50,000 years, igniting his passion for space travel as a teenager. It has now been flying in space around our solar system for about 10 million years or more.

Reading has always instilled ideas in the heads of creative people and developed into new inventions, from the "Earth" program in Neil Stephenson's novel "The Avalanche", from which the "Google Earth" application was inspired, to phones that have the ability to sense in Arthur C. Clarks' novel, which She led Tim Berners-Lee to invent the World Wide Web. Barack Obama says that reading helped him get to know himself and shaped his beliefs.

I have mentioned all these examples to demonstrate the importance of reading and that it was the first inspiration for all the great people who changed the course of history.

The question now is what is the relationship of all this to our project?

Our project aims to launch many services that facilitate the task of reading, and we can mention, for example, the word cloud, the mind map, and the summary of texts. All these services will help attract many readers and help them to reach the purpose of every article, report or scientific content that they read easily and without effort, our project will

help many creative people to continue achievements, change the concept of reading for some, and continue working until reading becomes an enjoyable activity.

2- Domain model

What is mind map?

Mind mapping is a popular brainstorming tool and thinking technique of visually arranging ideas and their interconnections. It is a way of representing related thoughts with symbols rather than with inessential words. The human mind forms associations almost promptly, and "mapping" allows capturing these ideas quicker than expressing them using only words or phrases. Originated in the late 1960s by Tony Buzan (1). It is now used by millions of people around the world.

Manually creating mind maps requires comprehensive reading and good understanding the text which takes much time and effort. In addition to that not all people are inventive enough to draw elegant and meaningful mind maps.

Therefore, automatically generating mind maps spares much time and exertion and serves way better and speedier several applications .

The applications that make the mind map is Xmind , FreeMind, Edraw MindMap, and MindMeister.

But this applications are manual, because the user manually determines the mind map element These softwares help the user in drawing the mind map and have some ready designs and diagrams which can be used. But the user must read, understand the text well and come up with a design for the mind map himself.

We provide Automatically generating mind maps out of pure text and that requires many stages of text processing.

What Is Word Cloud?

Word clouds (also known as text clouds or tag clouds) work in a simple way: the more a specific word appears in a source of textual data (such as a speech, blog post, or database), the bigger and bolder it appears in the word cloud.

A word cloud is a collection, or cluster, of words depicted in different sizes. The bigger and bolder the word appears, the more often it's mentioned within a given text and the more important it is.

Also known as tag clouds or text clouds, these are ideal ways to pull out the most pertinent parts of textual data, from blog posts to databases. They can also help business users compare and contrast two different pieces of text to find the wording similarities between the two.

Perhaps you're already leveraging advanced data visualization techniques to turn your important analytics into charts, graphs, and infographics. This is an excellent first step, as our brains prefer visual information over any other format.

Yet, what do you do if your raw data is text-based in nature?

Much of the research your organization conducts will include at least some form of an open-ended inquiry that prompts respondents to give a textual answer.

For instance, you might ask current customers what they like or don't like about your new product line. Or, you could ask them to give suggestions on how your organization could improve. They could also have the chance to elaborate on any pain points they're experiencing.

There are industry tools that allow you to code such open-ended data so users can understand it quantitatively. Yet, these don't come cheap. Word clouds offer a cost-effective, yet powerful, alternative.

With these, you can still quantify your text-based insights into measurable analytics. The only difference? You won't create a chart or graph as you would with a set of numbers. Instead, you'll create a word cloud generator to transform the most critical information into a word cloud.

Here's an example from USA Today using U.S. President Barack Obama's State of the Union Speech 2012

Advice for CHOOSING Word cloud (tips and traps)

Take time to clean up the data before importing it into the word cloud generator. For example, interview transcripts, may have common conversational fillers such as "you know" or "like" that need to be removed.

Ensure that the data collection team documents key words consistently– with consistent capitalization – since some software packages will not recognize capitalized and uncapitalized words as the same.

So, What about Summarization?

The navigation through long documents in order to find the interesting information is a tough job and waste of the time and effort. Automatic text summarization is a technique concerning the creation of a compressed form for single document for tackling such

problem. we introduced a solution for automatic text summarization problem; we tried to exploit different resources advantages in building of our app like advantage of diversity based method which can filter the similar sentences and select the most diverse ones and advantage of the non diversity method .

Moved by the cutting edge advancement and Innovation, Data is to this century what oil was to the last one. Today, our reality is parachuted by the gathering and dissemination of huge amounts of data.

In fact, the International Data Corporation that the total amount of digital data circulating annually around the world would sprout from 4.4 zettabytes in 2013 to hit 180 zettabytes in 2025. That's a lot of data!

With such a huge amount of data circulating in the digital space, there is a need to develop algorithms that can automatically shorten large huge texts and summaries that information that can fluently pass the intended messages.

However, auto-summarization used to be an impossible task. Specifically, abstractive summarization is very challenging. Differing from extractive summarization (which extracts important sentences from a document and combines them to form a "summary"), abstractive summarization involves paraphrasing words and hence, is more difficult but can potentially give a more coherent and polished summary.

Approaches

The summary should retain the most important points of the original text document. A good text summarizer should also take into account variables such as length, writing style and syntax of the original document.

Current approaches to automatic summarization fall into two broad categories: extraction and abstraction. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary. In contrast, abstractive methods first build an internal semantic representation and then use natural language

generation techniques to create a summary. Such a summary might contain words not explicitly present in the original document.
Key challenges in text summarization include topic identification, interpretation, summary generation, and evaluation of the generated summary. Most practical text summarization systems are based on some form of extractive summarization. Abstraction based summarization is inherently more difficult and is an active area of research.

All extraction based summarizers, irrespective of the differences in approaches, perform the following three relatively independent tasks:
(a) capturing key aspects of text and storing it using an intermediate representation.
(b) scoring sentences in the text based on that representation.
(c) and composing a summary by selecting several sentences.
Abstractive Summarization
task in Natural Language Processing (NLP) that aims to generate a concise summary of a source text. Unlike extractive summarization, abstractive summarization does not simply copy important phrases from the source text but also potentially come up with new phrases that are relevant, which can be seen as paraphrasing. Abstractive summarization yields a number of applications in different domains, from books and literature, to science and R&D, to financial research and legal documents analysis.
To date, the most recent and effective approach toward abstractive summarization is using transformer models fine-tuned specifically on a summarization dataset. we demonstrate how you can easily summarize a text using a powerful model within a few simple steps.
On a high level, PEGASUS uses an encoder-decoder model for sequence-to-sequence learning. In such a model, the encoder will first take into consideration the context of the whole input text and encode the input text into something called context vector, which is basically a numerical representation of the input text. This numerical representation will

then be fed to the decoder whose job is decode the context vector to produce the summary.

2.1 Related Work

2.1.1 M2Gen In 2009 (2)

The concept of M2Gen is to generate the Mind Map object from semantic model taken from a given text. The English text is transformed into a semantic model or meaning representation using numerous natural language understanding tools such as morphological analysis, parsing and Semantic Analyzer.

The complete procedure is shown in Figure 1.

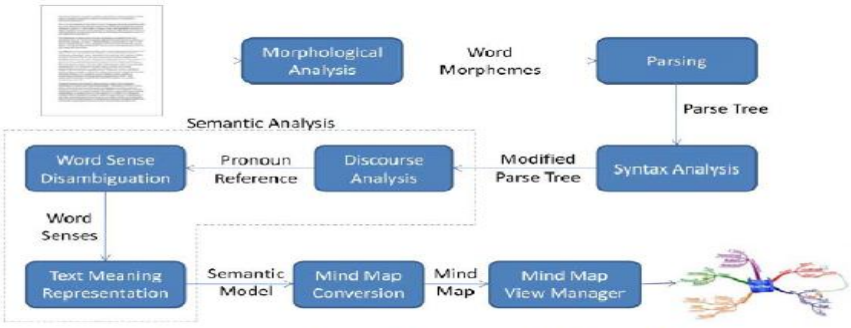


Figure 1. The Architecture of M^{2}Gen [2]

First, the English text is processed by morphological analysis in order to analyze each word into its lemma and affix along with its POS tag. The result of morphological analysis is then processed by parsing component by using CFG and top down chart parsing. Since not all parts of parse tree are used, then there is a parse tree alteration process in the syntax analysis. The result is then used by semantic analysis to produce the semantic model. The semantic analysis contains of several sub components such as discourse analysis, word sense disambiguation and text meaning representation. The discourse analysis aims to solve the pronoun reference in sentences .

The word sense disambiguation aims to select the best sense of a single word, and the text meaning representation aims to transform the parse tree result into the semantic model. The resulted semantic model is then converted into Mind Map figure. The example of generated Mind Map figure is shown in Figure 2.

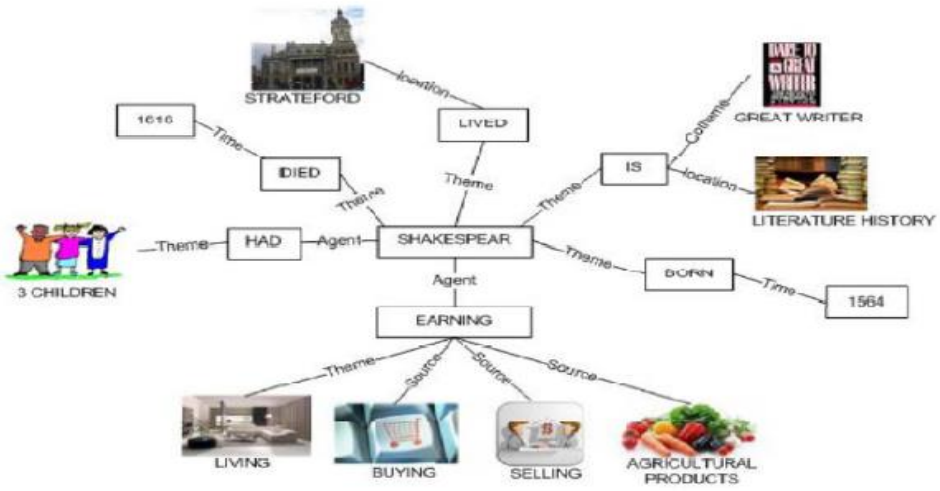


Figure 2. Example of Mind Map Resulted by M^{2}Gen [2]

On the other hand:

- this version only supports single layer mind map and does not produce a hierarchical (multilevel)
- the images assigned to the nodes are not accurately representing them

2.1.2 Generating Mind Map from Indonesian Text using Natural Language Processing (3)

This make the process easier by generating mind map from text (here is Indonesian text) and providing mind map editor to manipulate the object and relation set.

To build such tool, they utilize obtainable Indonesian NLP (Natural Language Processing) tools.

There are three constituents needed: semantic net generator, mind map visualization and interaction handler. In the semantic net generator, the resulted first order logic (FOL) resulted by the semantic analyzer is changed into semantic net which is represented by roll of objects and roll of relations. The resulted semantic net is then imagined by using combination method of radial and layering drawing. The interaction is obtainable for editing the object and the relation. The tool was then assessed by 2-experiment set: testing the semantic net generation and testing the resulted imagining.

The semantic net generation was assessed by using the valid input text, while the visualization was assessed by user acceptance test.

As the result, although the semantic net generation (from FOL) is an acceptable one, but the whole semantic analyzer for Indonesian text still has a soft accuracy specifically for composite sentence. As for the user acceptance test, the automatic generation still gives inconsequential object which should be altered by the interaction.

2.1.3 Generating mind map from an article using machine learning (4)

Data Collection: The data used in this study contains of two types, namely:

a. Article: article contains of numerous paragraphs, data in given to the expert for validation.

b. Validation Results: articles that have been stamped topic sentence in each paragraph.

Model design: At this stage there are several things that are improved that is Information Retrieval Approach where in it there is a pre-processing text procedure, Core NLP, Part of Speech Tagger. At the procedure of text processing technique used is segmentation, tokenization. Data is processed to distinct sentences into words per word text pre-processing, then done core NLP and part of speech tagger to determine the word type per word in the sentence. There are three types of words specified, among others, Noun, Verb, and Adjective. The following handle is the highlight extraction in which there's a prepare: frequency, sentence position value, and similarity with title and sub-title. After that, sentence scoring is done to sum up the value obtained from the feature extraction. Then the last phase is the sentence ranking and summary extraction after the calculation is done it will be selected the highest value to be the topic sentence of the paragraph .

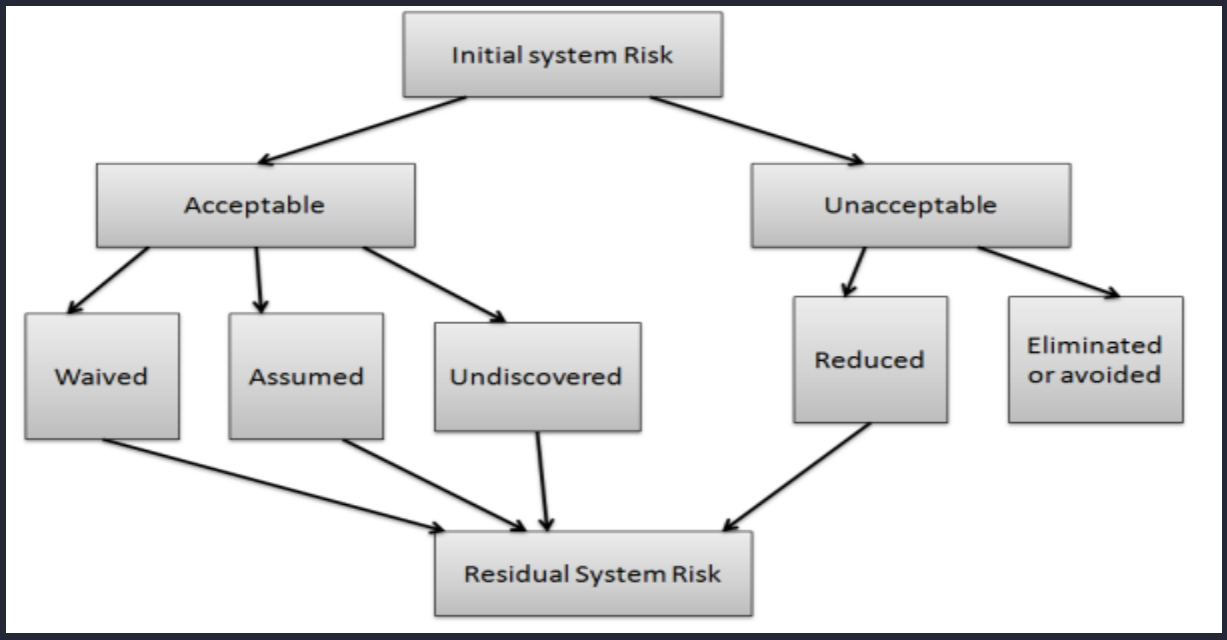
Testing is done by contrasting the concept map of automatic generation with concept map made by experts , The results showed 52.5% accuracy the accuracy of the system is average.

3- Risk/Constraints

3.1-Risk

A possibility of suffering from loss . Loss can be anything, can be anything, increment in production taken a toll, improvement of destitute software, not being able to total the project on time. Software risk exists because the future is ambiguous and there are many known and unknown things that cannot be combined in the plan.

In order to identify the risks that our project may be subjected to, it is vital to begin with think about the issues confronted by past ventures. Study the project plan accurately and check for all the conceivable regions that are helpless to a few or the other sort of dangers. The most excellent ways of analyzing a project is by changing over it to a flowchart and assay all fundamental areas. It is important to make few brainstorming sessions and mind map to identify the known unknowns that can influence the project. Any decision taken related to technical, operational, ,internal or external factors should be evaluated correctly.



Risk Planning

Consequently, we will provide problems that faced previous projects and potential risks:

- The accuracy of the mind maps produce by the model may be low in accuracy compared to the mind maps that humans produce for the same articles or reports
- Complex sentences that need to focus and understand the context of the whole sentence. The model may not define its meaning precisely or it may show undesirable results for these sentences.
- If we try to add more than one language that the model can deal with, we will face many difficulties, most notably languages that contain many words and there is a difficulty in understanding their grammatical rules, and among these languages of course is the Arabic language, and accordingly we think that we will focus on the English language.
- The M2Gen version only supports single-layer mind map and does not produce a hierarchy (multi-level), and we believe that we should solve this problem in order for our model to produce mind maps whose efficiency is higher than that of the mind maps produced by M2Gen.

3.2-Consraints

To minimize constraints, then, it is first helpful to understand their definition. A constraint is anything that slows a system down or prevents it from achieving its goal. We could think of a constraint as a bottleneck in our processes that impedes our progress. There are numerous, numerous distinctive sorts of constraint. Be that as it may, it's vital to note that a framework cannot have hundreds or thousands of limitations at one time. The most extreme is capped. This can be since constraints are the variables that most restrain your production.

- people constraints are caused by not having enough skill for a project or having skills not related to our project but we distribute tasks that suitable to each one as we can . While this can be the case, in software development, having as well

numerous talented individuals on a venture can moreover cause a people constraint (Known as and explained by Brooks' law.)

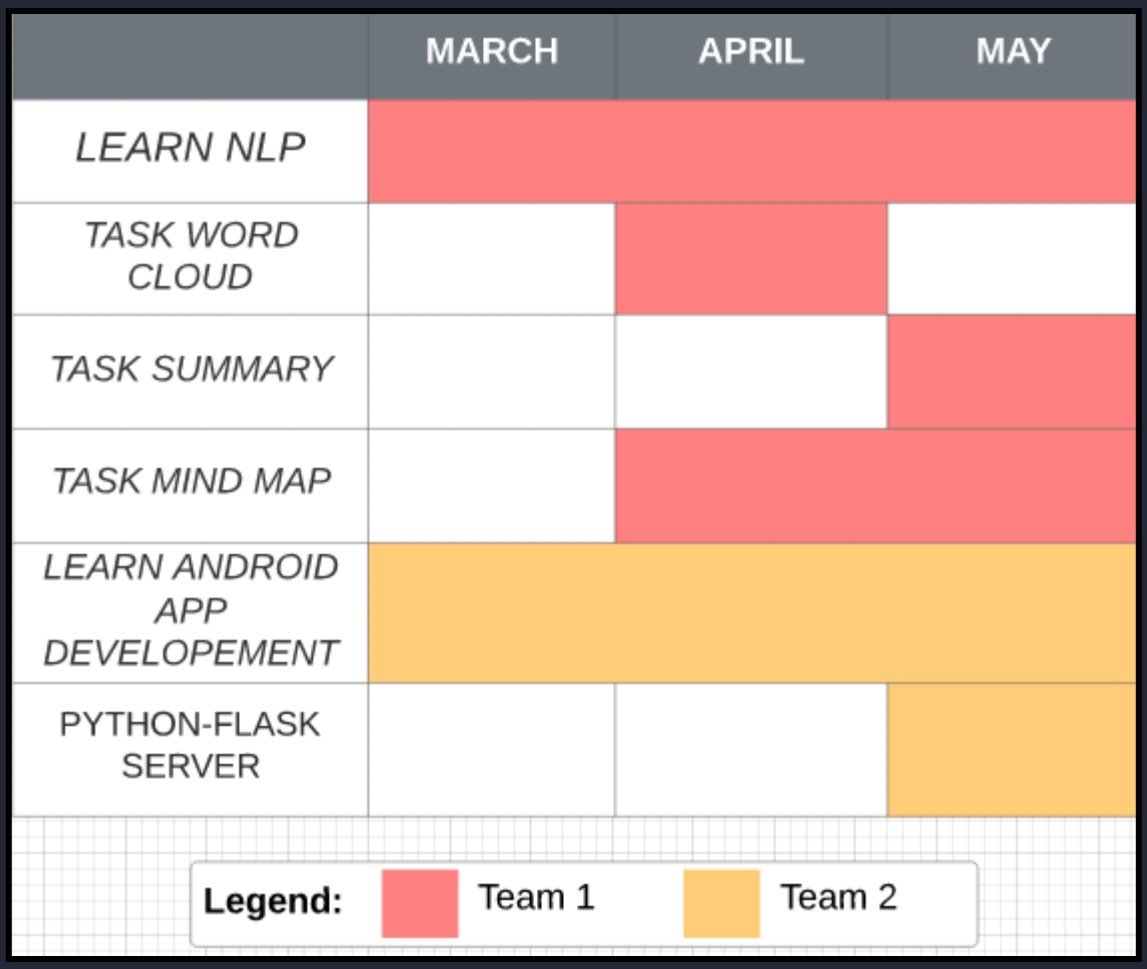
- The idea of the project should be purposeful and its implementation solves an existing problem or facilitates doing work that would have required much effort and time to complete it . And we believe that our project meets these conditions. **otherwise** You're spending time creating things that consumers don't want or need.
- Equipment constraints are those delays caused by defective, moderate, or out of date gear or a need of adequate space. In our project it might be a lack of devices And in this case we may go to the web because it does not require devices with high capabilities, all we need are devices that can run the browser, and unlike that in Android, which requires devices with specific capabilities to run Android Studio

4- Project plan

Project planning is at the heart of the project life cycle, and tells everyone involved where we are going and how we are going to get there.

Generate word cloud, summary & mind map automatically from articles using Natural language processing(NLP) is idea for our project and this is what we want to go to achieve **But How?**

the project work plan can be shown using Gantt chart to illustrate phases and tasks:



Gantt Chart

5- Software Quality Assurance (SQA)

(SQA) is easily a way to ensure quality in the software. It is the group of activities which ensure processes, procedures as well as standards suitable for the project and implemented accurately.

list of SQA activities	Description for each activity
Executing Formal Technical Reviews:	An FTR is done to evaluate the quality and design of the prototype. • In this process, a meeting is conducted with the team to discuss concerning the genuine quality prerequisites of the software and the design quality of the prototype. • This activity helps in discovering errors in the early stage of SDLC and reduces rework effort in the later stages.
Controlling Change:	In this activity, we use a combination of manual procedures and automated tools to have a mechanism for change control.

	• By validating the change requests evaluating the nature of change and controlling the change effect, • it is confirmed that the software quality is maintained during the development and maintenance stages.
Having a Multi-Testing Strategy:	By multi-testing strategy, we mean that one should not rely on any single testing tactic, instead, multiple types of testing should be performed so that the software product can be tested well from all viewpoints to ensure better quality
Enforcing Process Adherence:	This activity require for handle adherence amid the software development process. The advancement prepare ought to moreover adhere to the characterized methods. This activity is a mix of two sub-activities which are explained below in detail: • Product Evaluation: This activity confirms that the software product is meeting the requirements that were discovered in the project management plan. It ensures that the set standards for the project are followed correctly. • Process Monitoring: This activity verifies if the correct steps were taken during software development. This is done by matching the actually taken steps against the documented steps.

Table1: QSA

6. Limitation of the System

We could not implement the Abstractive summary. The model is a bit large and needs to be trained on large data in order to produce perfect results. There are also some defects in the model, for example

Most current state-of-the-art techniques revolves around encoder-decoder models, that may not be scalable enough for long document summarization.

From our experience working directly in abstractive summarization

The results are often small and almost give a line or several lines of summary, but this summary is strong and highly efficient, as we tried the model on our Abstract and the result was pretty good

```
input_ids = tokenizer.encode(ARTICLE, return_tensors='pt').to(torch_device)
output = model.generate(input_ids, max_length=200, num_beams=5, early_stopping=True).to(torch_device)
summary = tokenizer.decode(output[0], skip_special_tokens=True)

summary
```

```
'The goal of this project is to create an application that will help people to read better.'
```

We encountered problems including the lack of content related to this type of summary and in the end we failed to integrate the model with the application

7. System requirement specification

7.1 Function Requirements

This section **Function Requirements** contains the specification of function requirements of the system, below tables are the lists of function requirements

7.1.1 Generating Mind-map [GM]:

Requirement ID	Requirement Description
GM1	System **generating mind map automatically** to the uploaded Articles
GM2	The system can save the mind map image in the device locally

Table 5 Mind Map

7.1.2 Generating Word Cloud [GWC]:

Requirement ID	Requirement Description
GWC1	System **generating word cloud automatically** to the uploaded Articles
GWC2	System can save the word cloud locally in the device

Table 6 Word Cloud

7.1.3 Text Summarization [TS]:

Requirement ID	Requirement Description
TS1	System **generating the summary automatically** for the uploaded Articles
TS2	The system can save the summary in the device locally as a file

Table 8 Text Summarization

7.2 Non- Functional Requirement

7.2.1: Performance Requirement [PR]

Requirement ID	Requirement Description
PR1	System must respond The Operations in less 7 seconds for user
PR2	The System should be **compatible** with all android versions and browsers
PR3	**"Accessibility/direct access"** System should be accessible for all
PR4	**"Maintainability"** Make the system capable to use in order to achieve a particular goal
PR5	**"Extensibility"** The ability to make extend a system and this is by can be through the modification of previously existing functionality or through addition of new functionality in the system
PR6	**"Usability"** the system provide for user capability to use in order to achieve a particular goal
PR7	**"Privacy"** the system must be able to protect user information and articles from unauthorized person
PR8	**"Security"** the ability to provide protection to the system from attack and intrusions.

Table 6: performance Requirement

7.2.2: Safety and Security Requirement [SSR]:

Requirement ID	Requirement Description
SSR1	The System checks if the uploaded data is real data not empty (50 char as minimum).
SSR2	Hashing technology and Algorithms should be used to handle the secure login for users.
SSR3	The system could use SSL (Secured Socket Layer) certificates to secure the data transmitted.

Table 7 Safety and security Requirement

7.2.3: Reliability [R]:

System should be designed in modular manner to ease in software maintenance. By designing modularly, we are able to reduce coupling allowing each module to perform a specific function.

Requirement ID	Requirement Description
R1	The Program should be **reliable** and provide catching of exceptions so that unintended result do not occur such as system crashes or data validation failure.

Table 8: Reliability

7.2.4 Other Software Quality Attributes:

7.2.4.1 Usability:

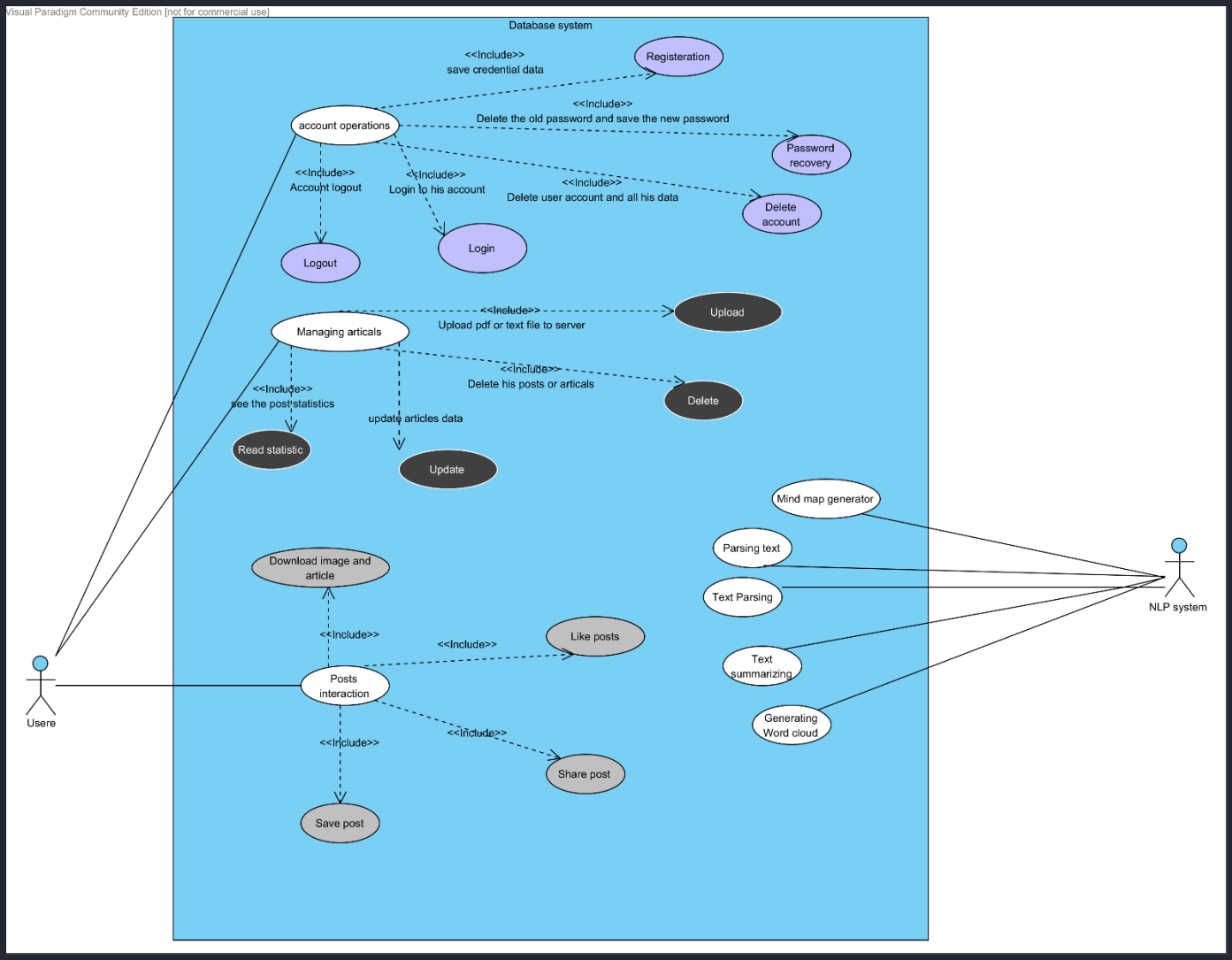
Usability the system provide for user capability to use in order to achieve a particular goal.

The system should have user-friendly interface

7.2.4.2 Availability:

The system must be available 24/7, with no more than 20 minutes down time per day

8- Use-case

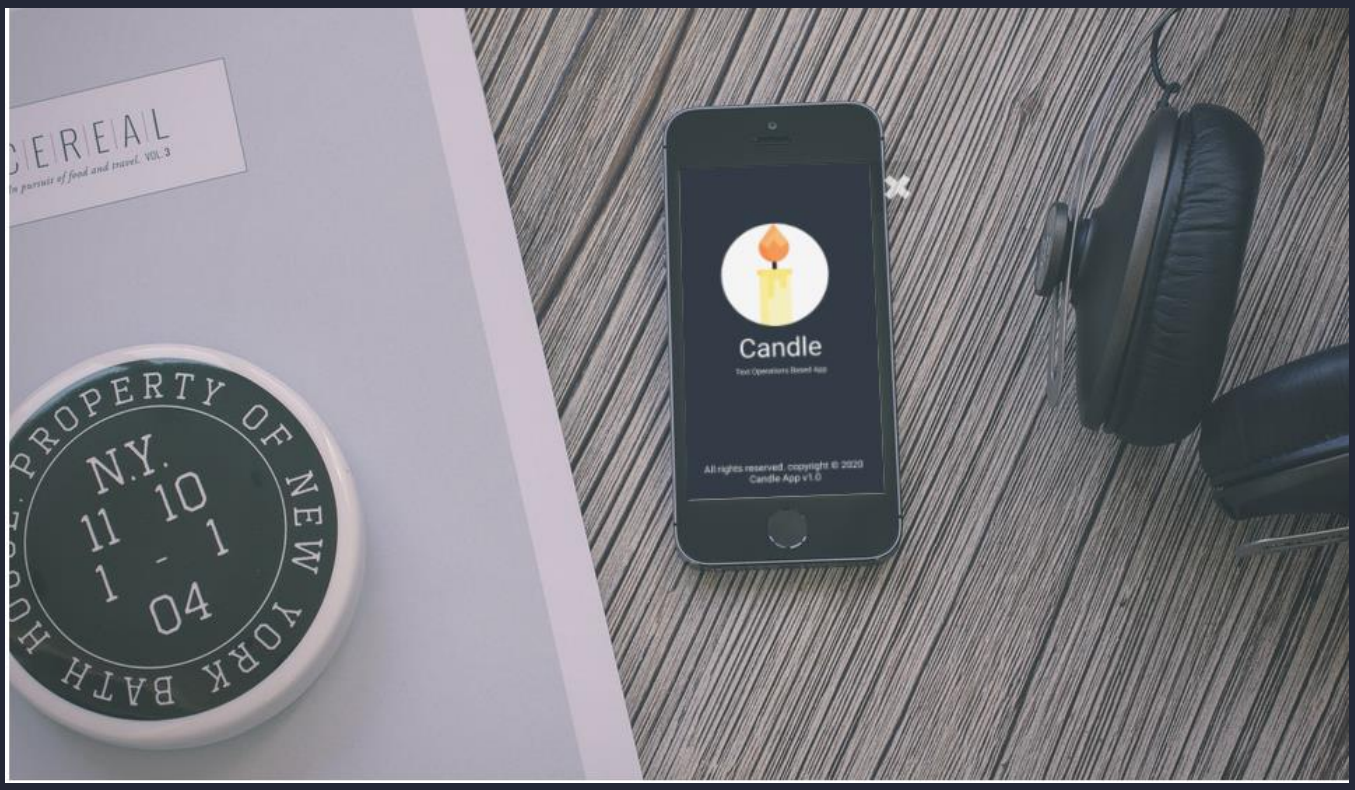

Visual Paradigm Community Edition [not for commercial use]
Database system
Registeration
<<Include>>
save credential data
account operations
<<Include>>
Delete the old password and save the new password
Password recovery
<<Include>>
Account logout
<<Include>>
Login to his account
<<Include>>
Delete user account and all his data
Delete account
Logout
Login
Managing articals
<<Include>>
Upload pdf or text file to server
Upload
<<Include>>
Delete his posts or articals
Delete
<<Include>>
see the post statistics
update articles data
Read statistic
Update
Mind map generator
Parsing text
Text Parsing
Text summarizing
Generating Word cloud
NLP system
Download image and article
<<Include>>
<<Include>>
Like posts
Posts interaction
<<Include>>
<<Include>>
Share post
Save post
Usere

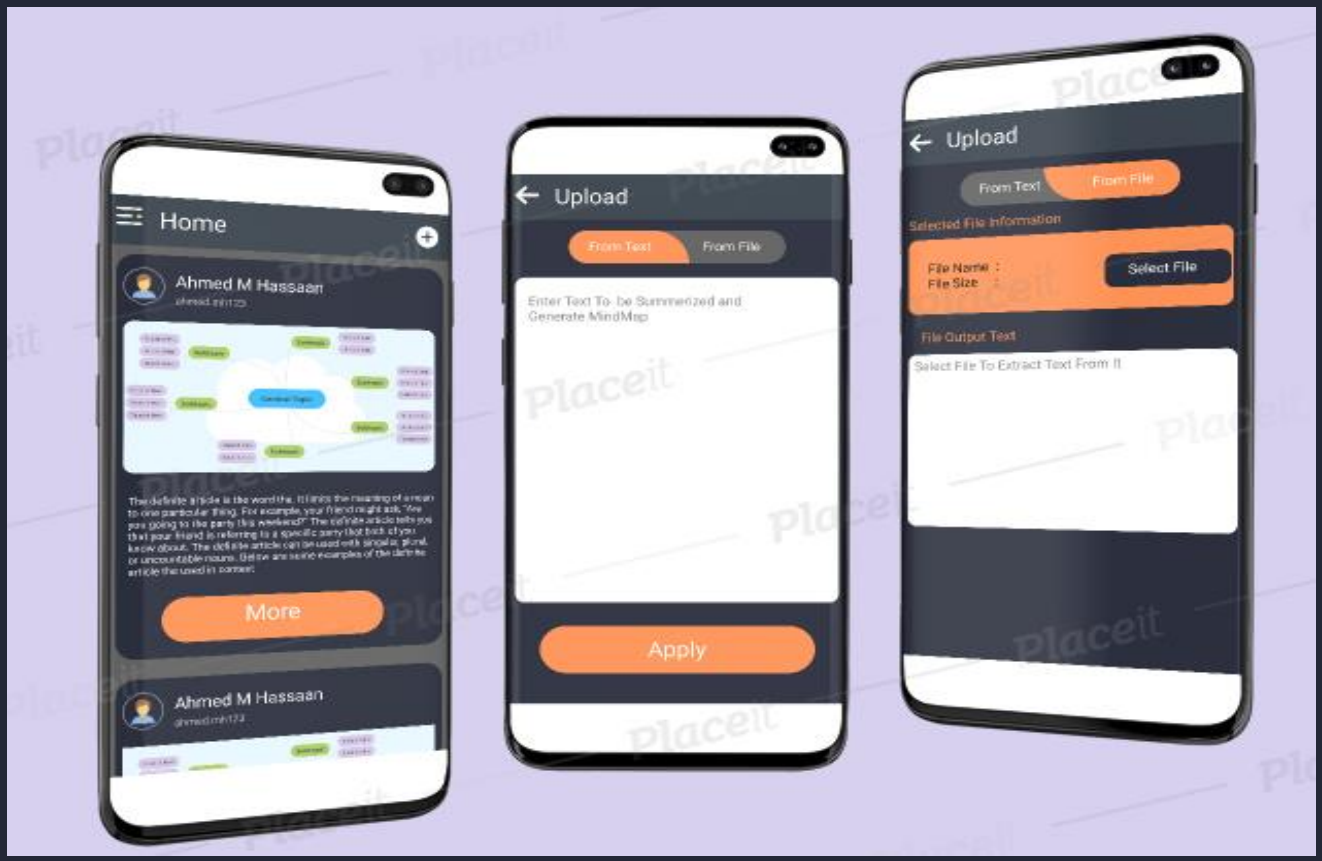
8.1-Use-case description

Table 1: NLP System

Use case Name	NLP System
Scope	Product
Level	Task
Description	This use case does the all the NLP tasks (text preprocessing, Text parsing, Text summarization, Mind map generation and word cloud generation)
Step performed	This use case starts when file uploaded 1. Text preprocessing • The system does some operations on the text • The system executes a standard template from text 2. Morphological analysis • In this phase the analysis for each word is done • The result of the analysis is each word with its lemma and affix along with its pos tag 3. Parsing • A CFG (top down) grammar is used 4. Syntax analysis • Generates a modified parse tree 5. Sematic analysis • Discourse analysis This step generates a pronoun reference • Word sense disambiguation This step generates a word sense • Text meaning representation After this step we will have a semantic model 6. Mind map conversion This phase uses the semantic model to generate the mind map 7. Mind map generation
Alternative path	The uploaded file is too large.
Precondition	A text must be uploaded.

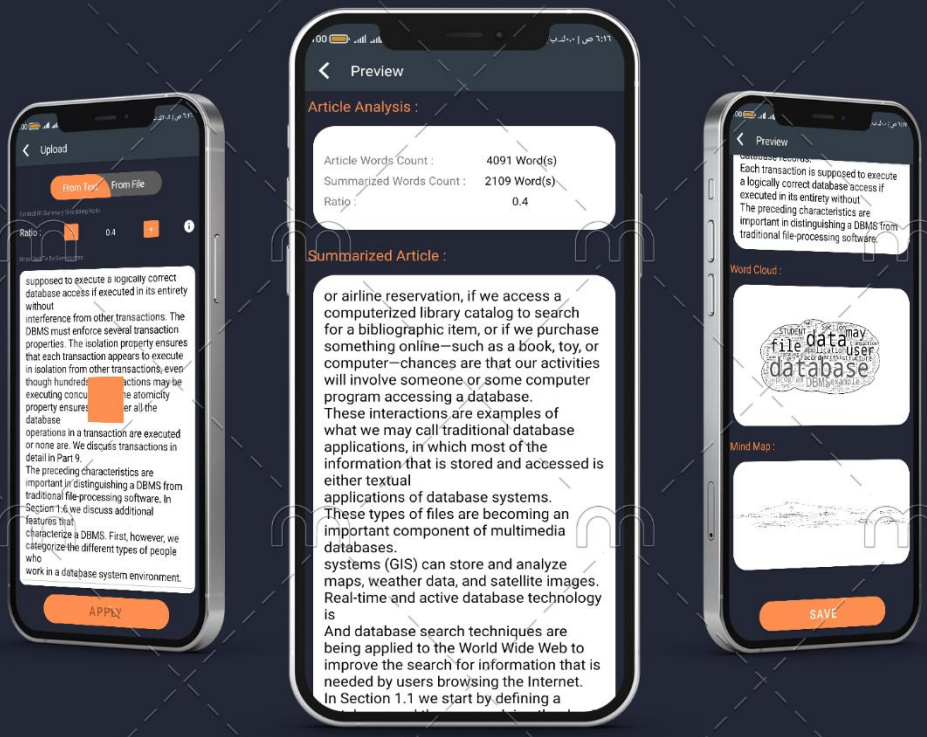
9- Graphical User Interface:



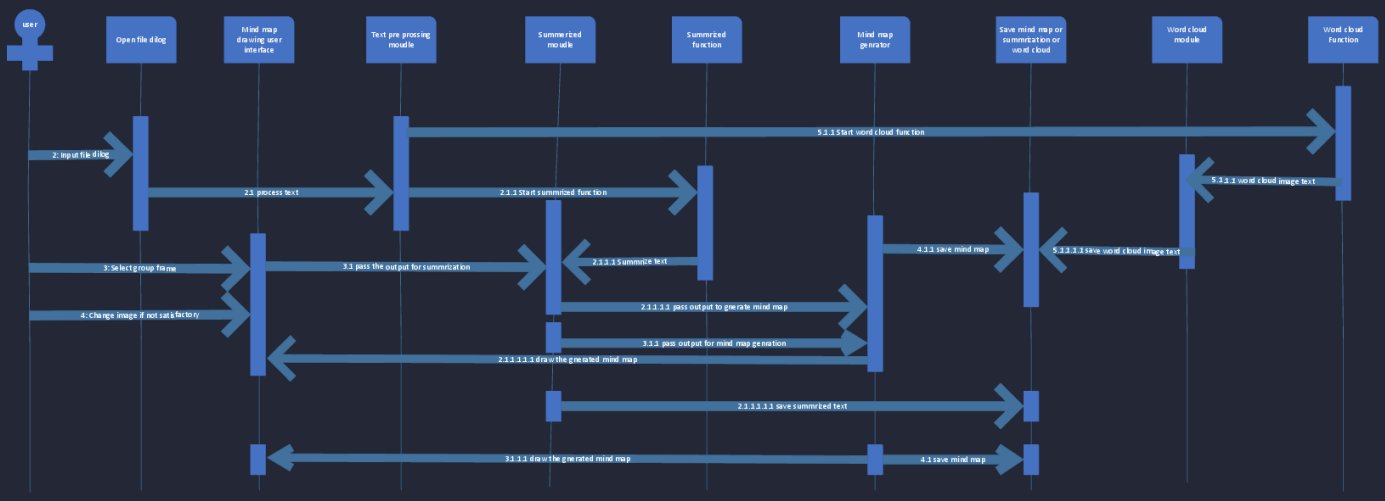



Mind Map Generator
Generating Mind Map From Articles Automatically
Skip
Next
Text Summarization
Summarize Articles Automatically For Any Uploaded Article
Skip
Next
Candle App
Text Operation Based App.
START
All Rights Reserved To Candle App © 2021


START
Upload
From Text
From File
Ratio :
0.4
APPLY
Preview
Article Analysis :
Article Words Count :
Summarized Words Count :
Ratio :
4091 Word(s)
2109 Word(s)
0.4
Summarized Article :
Word Cloud :
Mind Map :
Saved Successfully
SAVE


Upload
From Text
From File
Ratio :
0.4
APPLY
Preview
Article Analysis :
Article Words Count : 4091 Word(s)
Summarized Words Count : 2109 Word(s)
Ratio : 0.4
Summarized Article :
or airline reservation, if we access a computerized library catalog to search for a bibliographic item, or if we purchase something online—such as a book, toy, or computer—chances are that our activities will involve someone or some computer program accessing a database.
These interactions are examples of what we may call traditional database applications, in which most of the information that is stored and accessed is either textual
applications of database systems.
These types of files are becoming an important component of multimedia databases.
systems (GIS) can store and analyze maps, weather data, and satellite images.
Real-time and active database technology is
And database search techniques are being applied to the World Wide Web to improve the search for information that is needed by users browsing the Internet.
In Section 1.1 we start by defining a
Word Cloud :
Mind Map :
SAVE

10. Sequence diagram



11. Prototype Description

We use android studio to create a system and we save the result {mind map , summarization ,word cloud } local in the device in any android devices (tv , phone , tab …….) and use python to generate the mind map and make summarization and word cloud

Make a relation between android studio an muddle m5 in python let's get the result from our application

12. Implementation Platform:

12.1 Flask framework

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application

12.2 Post man

Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs—faster.

12.3 Android studio (java)

You write Android apps in the Java programming language using an IDE called Android Studio. Based on JetBrains' IntelliJ IDEA software, Android Studio is an IDE designed specifically for Android development.

12.4 Hosting

We use **pythonanywhere.com** to host flask files and images in backend to send and receive requests from and to it.

13.Implementation Details

To implement summarization that sum up the text and make easier for user to read and under stand the text:

```python
@app.route('/articles/summary', methods = ['POST'] )
def summarizeArticleRequestHandling():

    articleParam= 'article'
    ratioParam  = 'ratio'

    if  articleParam  in request.form :
        text  = str(request.form.get(articleParam))

    else:
        return returnError("Article Not Found In  Request")

    if  ratioParam in request.form:
        r = str(request.form.get(ratioParam, type=float))
        try:
            ratio = float(r)
        except Exception as e :
            return returnError(f'{r} Not Float Value\n{e}' )

    else:
        return returnError("Ratio Not Found In  Request")

    if ratio > 1 or ratio <0:
        return returnError("Ratio Must Be > 0 And <1 ")
    else:
        # text = text.replace('.', '.\n')
        summary=summarize(text ,ratio)
        # summary= ''   #temp
        # return 'The Summary is : '+summary
        # return 'The Summary is : '+text
        wordCount = len(text.split())
        summWordsCount = len(summary.split())
        jStr = {'mainArticleWordsCount' : wordCount  , 'summarizedWordsCount':  summWordsCount,'summarizedArticle': summary }


        responsStr = {'code':1, 'message': 'The Text Summarized Successfuly', 'response' : jStr}
        return json.dumps(responsStr)
```

Generate mind map from article:

```
groucho_grammar = nltk.CFG.fromstring("""
S -> NP VP
PP -> P NP
NP -> Det N | Det N PP | 'I'
VP -> V NP | VP PP
Det -> 'an' | 'my'
N -> 'elephant' | 'pajamas'
V -> 'shot'
P -> ''in'
""")

parser = nltk.ChartParser(groucho_grammar)
for tree in parser.parse(sent):
print(tree)
```

```python
@app.route('/articles/mindmap', methods = ['POST'] )
def summarizeArticleRequestHandling():

    articleParam= 'article'


    if  articleParam  in request.form :
        text  = str(request.form.get(articleParam))

    else:
        return returnError("Article Not Found In  Request")


    text1=word_tokenize(text)
    pos=pos_tag(text1)
    N=VB=DT=IN=JJ=''
    print(pos)
    for i in pos:
        if(i[1]=='CD' or i[1]=='CC'):
            print(i)
    for i in pos:
        if( i[1]=='NNS' or i[1]=='NN' or i[1]=='NNP'):
            N=N+i[0]+' | '
        elif(i[1]=='VBP' or i[1]=='VBZ' or i[1]=='VBN' or i[1]=='VB' or i[1]=='VBG'):
            VB=VB+i[0]+' | '
        elif(i[1]=='DT' or i[1]=='WDT' or i[1]=='PDT'):
            DT=DT+i[0]+' | '
        elif(i[1]=='IN'):
            IN=IN+i[0]+' | '
        elif(i[1]=='JJ' or i[1]=='JJS'):
            JJ=JJ+i[0]+' | '
    N=N.split()
    N.pop(-1)
    nns=''
    for i in N:
        nns=nns+i
```

To generate word-cloud we use:

```python
@app.route('/articles/generate/wordcloud', methods = ['GET','POST'] )
def wordCloudRequest():

    articleParam = 'article'
    if articleParam in request.form :
        text = str(request.form.get(articleParam))
    else :
        return returnError('No Text Founded')

    if text == 'None':
        return returnError("Send Article To Start Processing")
    else:
        img=Image.open('/home/aabojana09/mysite/cloud.png')
        mask =np.array(img)
        stopwords=set(STOPWORDS)
        base =WordCloud(background_color="white",mask =mask,height=1500,width=1700,contour_width=3,contour_color='black',stopwords=stopwords).generate(text)
        #base =WordCloud(background_color="orange",mask =mask,height=820,width=820,contour_width=3,contour_color='black',stopwords=stopwords).generate(text)
        image_colors = ImageColorGenerator(mask)
        base.recolor(color_func = image_colors)
        plt.title("Document")
        plt.imshow(base)
        plt.axis("off")
        plt.show()
        base.to_file('/home/aabojana09/mysite/wordcloud.png') #save world cloud pic in same dir
        # base.to_file('/home/aabojana09/mysite/__pycache__/wordcloud.png') #save world cloud pic in same dir

        response = {} # create empty Dictionary
        response['code'] =1
        response['message'] = 'The Output Generated With No Error '

        imgResponse = {}
        # imgResponse['imageLink'] = "https://www.pythonanywhere.com/user/aabojana09/files/home/aabojana09/mysite/wordcloud.png"
        imgResponse['imageLink'] = "https://www.pythonanywhere.com/api/v0/user/aabojana09/files/path/home/aabojana09/mysite/wordcloud.png"
        # imgResponse['imageLink'] = "https://www.pythonanywhere.com/user/aabojana09/files/home/aabojana09/mysite/__pycache__/wordcloud.png"
```

Http Request Client :

```java
public class RetrofitClient {

    private static RetrofitClient mInstance;
    private static Retrofit retrofit;
    static final String BASE_URL = "https://aabojana09.pythonanywhere.com/";

    private RetrofitClient() {
        retrofit = new Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .client(
                        new OkHttpClient
                                .Builder()
                                .callTimeout( timeout: 30, TimeUnit.SECONDS)
                                .connectTimeout( timeout: 30, TimeUnit.SECONDS).build()
                )
                .build();
    }

    public static synchronized RetrofitClient getInstance() {
        if (mInstance == null)
            mInstance = new RetrofitClient();
        return mInstance;
    }

    public ArticlesOperations getArticlesOperation() {
        return retrofit.create(ArticlesOperations.class);
    }
```

Sending Summarized request to the host(server)

```java
    public void summarize(String article, double ratio) {

        Log.d(App.TAG,  msg: "summarize: Ratio = " + ratio);
        RetrofitClient
                .getInstance() RetrofitClient
                .getArticlesOperation() ArticlesOperations
                .summarizeArticle(article, ratio) Call<ResponseSchema<SummarizedArticle>>
                .enqueue(new Callback<ResponseSchema<SummarizedArticle>>() {
                    @Override
                    public void onResponse(@NonNull Call<ResponseSchema<SummarizedArticle>> call, @NonNull Response<ResponseScher
                        if (response.body() != null && response.isSuccessful()) {
                            summarizedLiveData.setValue(response.body());
                        } else {

                            try {
                                summarizedLiveData.setValue(new ResponseSchema<>( resultCode: 0,  message: "Summarization Says : \n"

                            } catch (IOException e) {
                                summarizedLiveData.setValue(new ResponseSchema<>( resultCode: 0,  message: "Summarization Page Not F
                            }

                        }
                    }

                    @Override
                    public void onFailure(@NonNull Call<ResponseSchema<SummarizedArticle>> call, @NonNull Throwable t) {
                        summarizedLiveData.setValue(new ResponseSchema<>(t));
                    }
                });
    }
```

Handle response of summarize request

```
articlesViewModel.getSummarizedLiveData().observe(getViewLifecycleOwner(), schema -> {

    enableView( enalbe: true);

    if (schema.isSuccessful()) {
        SummarizedArticle response = schema.getResponse();
        article.setSummarizedArticle(response.getSummarizedArticle());
        article.setMainArticleWordsCount(response.getMainArticleWordsCount());
        article.setSummarizedWordsCount(response.getSummarizedWordsCount());
        Log.d(App.TAG,  msg: "observeViewModels:  Count = "
                + response.getSummarizedWordsCount() + "\nSummarizedText is : " + response.getSummarizedArticle());
          msg(schema.getMessage());
        enableView( enalbe: false);
        articlesViewModel.generateWordCloud(article.getArticle());

    } else {

        errorOccurred(schema.getMessage(), dialog -> {...},
                dia -> {...});
    }
});
```

Sending word-cloud request to the host(server)

```
public void generateWordCloud(String article) {
    RetrofitClient
            .getInstance() RetrofitClient
            .getArticlesOperation() ArticlesOperations
            .generateWordCloud(article) Call<ResponseSchema<WordCloud>>
            .enqueue(new Callback<ResponseSchema<WordCloud>>() {
                @Override
                public void onResponse(@NonNull Call<ResponseSchema<WordCloud>> call, @NonNull Response<Resp
                    if (response.isSuccessful() && response.body() != null) {
                        wordCloudLiveData.setValue(response.body());
                    } else {
                        wordCloudLiveData.setValue(new ResponseSchema<>( resultCode: 0,  message: "WordCloud Pa
                    }
                }

                @Override
                public void onFailure(@NonNull Call<ResponseSchema<WordCloud>> call, @NonNull Throwable t) {
                    wordCloudLiveData.setValue(new ResponseSchema<>(t));
                }
            });
}
```

Handle response of word-cloud request

```
articlesViewModel.getWordCloudLiveData().observe(getViewLifecycleOwner(), schema -> {
    enableView( enalbe: true);
    if (schema.isSuccessful()) {
        article.setWordCloud(schema.getResponse().getImageLink());
        Log.d(App.TAG,  msg: "observeViewModels:  WordCloud is : " + schema.getResponse().getImageLink());

          msg(schema.getMessage());
        enableView( enalbe: false);
        articlesViewModel.generateMindMap(article.getSummarizedArticle());


    } else {

        // to avoid null word cloud in result page we set default value for wordCloud as empty string
        errorOccurred(schema.getMessage(), dialog -> {...}, dialog -> article.setWordCloud(""));

    }
});
```

Sending mind map request to the host(server)

```
public void generateMindMap(String summarizedArticle) {

    RetrofitClient
            .getInstance() RetrofitClient
            .getArticlesOperation() ArticlesOperations
            .generateMindMap(summarizedArticle) Call<ResponseSchema<MindMap>>
            .enqueue(new Callback<ResponseSchema<MindMap>>() {
                @Override
                public void onResponse(@NonNull Call<ResponseSchema<MindMap>> call, @NonNull Response<ResponseSchema<
                    if (response.isSuccessful() && response.body() != null) {
                        mindMapLiveData.setValue(response.body());
                    } else {
                        mindMapLiveData.setValue(new ResponseSchema<>( resultCode: 0,  message: "MindMap Page Not Found
                    }
                }

                @Override
                public void onFailure(@NonNull Call<ResponseSchema<MindMap>> call, @NonNull Throwable t) {
                    mindMapLiveData.setValue(new ResponseSchema<>(t));
                }
            });
}
```

Handle response of Mind map request

```
articlesViewModel.getMindMapLiveData().observe(getViewLifecycleOwner(), schema -> {
    enableView( enalbe: true);
    if (schema.isSuccessful()) {

        msg(schema.getMessage()); // show message from server side
        article.setMindMap(schema.getResponse().getImageLink());  // assign mindMap to article

        Tools.msg( msg: "Process Finished Successfully", ToastTags.TOAST_SUCCESS);
        prepareArticleForShown();

    } else {
        // show the error message and ask for retry
        tools.customMessage(new CustomDialogModel(
                "New Message",
                 message: schema.getMessage() + ".\n" + "Result Not Completely Successful  Show AnyWay" + "\nOr Retry Aga:
                "Retry",
                "Show Result",
                "Cancel",
                dialog -> {...},
                dialog -> {...},
                dialog -> {...},
                 isCancelable: false,
                 imageRes: 0

        ));
```

14. Testing

Software Testing is the process of identifying the accuracy and quality of the software product and service under test.At the end of the day, testing executes a framework or application with a specific end goal to point out bugs, errors or defects At the end of the day, testing executes a framework or application with a specific end goal to point out bugs, errors or defects. The responsibility of testing is to point out the issues of bugs and give (Developers) a clue to help them fix it right following the requirements

And we use Android Studio Profiling Tool to make tasting for our application and the result become :

- Max RAM : 128 MB * Max CPU : 8% * Max Network : 130 kb/s

14.1 Unit Testing

Unit tests are the fundamental tests in your app testing strategy. By creating and running unit tests against your code, you can easily verify that the logic of individual units is correct. Running unit tests after every build helps you to quickly catch and fix software regressions introduced by code changes to your app.

A unit test generally exercises the functionality of the smallest possible unit of code (which could be a method, class, or component) in a repeatable way. You should build unit tests when you need to verify the logic of specific code in your app. For example, if you are unit testing a class, your test might check that the class is in the right state. Typically, the unit of code is tested in isolation; your test affects and monitors changes to that unit only.

```
@Test
public void testHost() {

    RetrofitClient
            .getInstance() RetrofitClient
            .getArticlesOperation() ArticlesOperations
            .testHost() Call<ResponseBody>
            .enqueue(new Callback<ResponseBody>() {
                @Override
                public void onResponse(@NonNull Call<ResponseBody> call, @NonNull Response<ResponseBody> response) {
                    if (response.isSuccessful() && response.body() != null){
                        try {
                            assertEquals( expected: "test",response.body().string());
                        } catch (IOException e) {
                            e.printStackTrace();
                        }
                    }
                }

                @Override
                public void onFailure(Call<ResponseBody> call, Throwable t) {

                }
            });

}
```

15- Conclusion and Future Work

When we talk about future work, we can talk about a way to integrate this type of Abstractive summary into our application, which will undoubtedly be a big plus especially if we could find a way to make the summary a little bit longer.

We have discussed in this document more than one service that can help people to read and extract important information from texts because we believe in the importance of reading and it is the first inspiring activity for all creative people who changed the course of history, and this is precisely a point that we referred to in the Abstract. 0

It is clear that we have focused on the mind map because of its advantages, and if it is extracted automatically from the texts, it will be an easy way to understand the content and there is consensus on its importance and the ease of remembering the mind map.

We face many difficulties, including, for example, the production of two versions of the application, one in Arabic and the other in English, and the lack of educational content for natural language processing (NLP)compared to other common topics.

All these challenges motivate us more for success and for the benefit to prevail and for the largest number of people to benefit from the application, and life is full of success and failure.

We hope that our application will help a lot of people, impress readers, and make reading a fun activity.

with our cooperation, we will be able to achieve the achievement.

References

- Word Clouds & the Value of Simple Visualizations

https://boostlabs.com/blog/what-are-word-clouds-value-simple-visualizations/

- Automatic summarization

https://en.wikipedia.org/wiki/Automatic_summarization

- Automatic Text Summarization

https://scialert.net/fulltext/?doi=itj.2009.1249.1255

- Text Summarization with Python

https://medium.com/@umerfarooq_26378/text-summarization-in-python-76c0a41f0dc4

- How to Perform Abstractive Summarization with PEGASUS

https://towardsdatascience.com/how-to-perform-abstractive-summarization-with-pegasus-3dd74e48bafb

- (M. Abdeen, R. El-Sahan, A. Ismaeil, S. El-Harouny, M. Shalaby)

The Faculty of Computers and Information Sciences | Ain-Shams University Cairo, Egypt.

https://www.researchgate.net/publication/224129737_Direct_automatic_generation_of_mind_maps_from_text_with_M2Gen

- Generating mind map from an article using machine learning: Nurrokhim in 2019

https://iopscience.iop.org/article/10.1088/1742-6596/1280/3/032023