VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

VIT Chennai

Vandalur - Kelambakkam Road, Chennai - 600 127 **Final**

Review Report

Program	**:**	Integrated Mtech
Course	**:**	Natural Langauge Processing
Slot	**:**	A2
Faculty	**:**	Mrs. PremLatha .M
Component	**:**	J
Title	**:**	Text summarization using Spacy

Team Member: -

Atul Patel (20MIA1134)

SHASHANK Pandey (20MIA1147)

Gaurav Sharan (20MIA1081)

ACKNOWLEDGEMENT

We wish to express our sincere thanks and a deep sense of gratitude to our project guide, Dr. M Premlatha, for her consistent encouragement and valuable guidance pleasantly offered to us throughout the course of the project work. We also take this opportunity to thank all the faculty of the School for their support and the wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

CONTENTS:

- Abstract
- Introduction
- Problem Statement & Objective
- Proposed Model/Diagram
- Result Analysis
- Conclusion and Future Scope
- References

ABSTRACT:

Text summarization is the process of generating a shorter version of a longer text while preserving its key information. Spacy is a popular open-source NLP library that provides several built-in tools for text summarization. One of the most common approaches to text summarization using Spacy is extractive summarization, where important sentences or phrases are extracted from the original text and combined to form a summary.

Spacy's built-in statistical models and algorithms can help identify key sentences and phrases by analysing word frequency, sentence structure, and semantic meaning. Additionally, Spacy allows for the customization and finetuning of models to improve the accuracy and relevance of generated summaries. Overall, Spacy is a useful tool for text summarization that can assist in automating the process of extracting important information from large texts.

INTRODUCTION:

The text summary is a way of selecting important points from the provided article or a document that can be reduced by a program. As the data overload problem increased, so did the interest in capturing the text as the amount of data increased. Summarizing a large document manually is challenging since it requires a lot of human effort and is time-consuming.

There are mainly two methods for summarizing the text document that can be done by using extractive and abstractive techniques. Extractive summaries concentrate on selecting important passages, sentences, words, etc. from the primary text and connecting them into a concise form. The importance of critical sentences is concluded based on the analytical and semantic features of the sentences.

Summary systems are usually based on sentence delivery methods and for understanding the whole document properly as well as for extracting the important sentences from the document. The technique of generating a brief description that comprises a few phrases that describe the key concepts of an article or section is known as abstractive summarization. This function is also included to naturally map the input order of words in a source document to the target sequence of words called the summary

The goal of text summarization is to make large volumes of information more manageable and accessible, particularly in situations where time and attention are limited. Text summarization techniques can be applied to a variety of text types, including news articles, scientific papers, legal documents, and social media posts. With the increasing amount of digital content being generated every day, text summarization is becoming an increasingly important tool for improving information retrieval and knowledge management.

Problem Statement & Objectives:

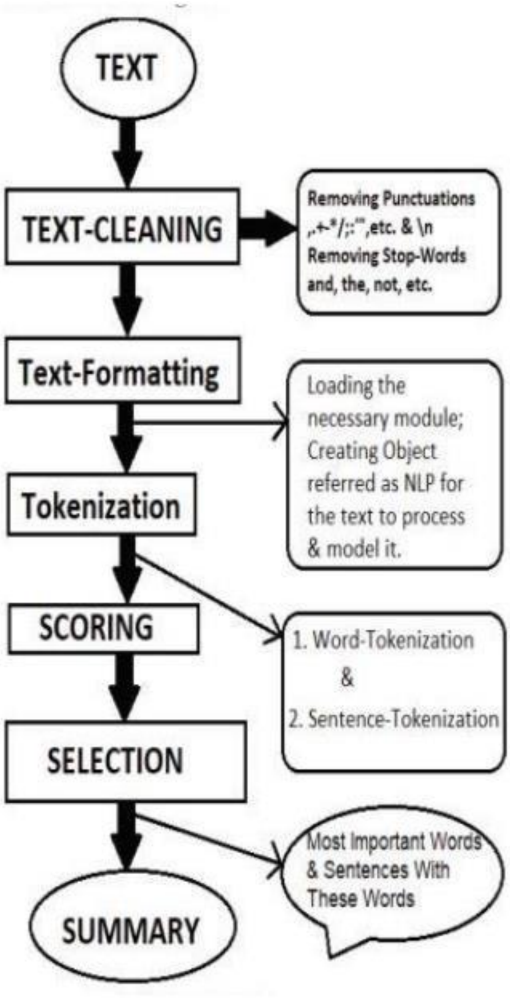
Problem statement-Text summarization is the process of condensing a text document into a shorter version while retaining important information and the text's overall meaning. With the abundance of information available on the internet, it is becoming increasingly difficult to consume large volumes of text data in a short amount of time. Text summarization can help in various applications, such as news article summarization, document summarization, and social media post summarization.

Objectives:

- To develop an NLP-based system for text summarization using the Spacy library.
- To extract important sentences from the input document and condense them into a shorter version.
- To retain the overall meaning of the text while removing redundant information.
- Summaries reduce reading time.
- When researching documents, summaries make the selection process easier.
- Automatic summarization improves the effectiveness of indexing.
- Automatic summarization algorithms are less biased than human summarizers.
- Personalized summaries are useful in question-answering systems as they provide personalized information.
- Using automatic or semi-automatic summarization systems enables commercial abstract services to increase the number of text documents they can process.

Proposed Model/Diagram:

Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken and written referred to as natural language. It is a component of artificial intelligence. NLP the abbreviation of Natural Language Processing is the branch of artificial intelligence that is it intersection of Computational Learning & Linguistics (Natural Languages) or the communicating tool of humans. Natural Language Processing is part of advanced technology used to give insights into natural languages to the machine. The objective list of NLP extends from simple interpretation to complex comprehension i.e., to reading, comprehending, interpreting, deciphering, and making sense of human freedom, and languages in a manner that is meaningful to the machines Now, in this segment we are about to review numerous works that we will accomplish on Text Summarization as shown in architecture diagram and the points are given below. We are essentially going to represent their approach and workflows


TEXT
TEXT-CLEANING
Removing Punctuations ,.+-*/;:'",etc. & \n Removing Stop-Words and, the, not, etc.
Text-Formatting
Loading the necessary module; Creating Object referred as NLP for the text to process & model it.
Tokenization
1. Word-Tokenization & 2. Sentence-Tokenization
SCORING
SELECTION
Most Important Words & Sentences With These Words
SUMMARY

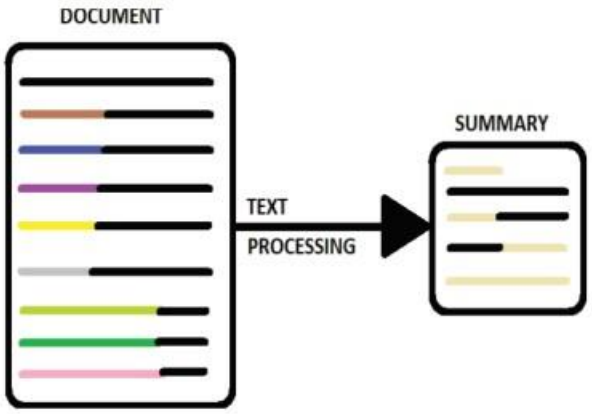
1. **Text processing:** Text processing includes Text Cleaning and Text Formatting in which text cleaning will be where we will be removing punctuations like &; . : etc and removing stop words and, the, not, etc. Text formatting is where we will be loading the necessary module, creating an object refers to ed as NLP for the text to process & model it. The automated process of analysis & manipulation of the text is known as text processing. It takes the text as input, processes it & finally provides the required outcome; it could be widely used within different areas of an organization, such as product teams could get insights from customer feedback to automate customer services. Here, words/tokens of the text represent discrete, categorical features.
2. 2. **Tokenization:** Splitting into tokens. Tokens refer to any individual unit in the program which is meant for either the machine or the human.
   And it contains word tokenization and sentence tokenization.

   a) **Word-Tokenization:** When the entire text is divided into individual words and a word score is generated for every word according to its count.

   b) **Sentence-Tokenization:** When the entire text is divided. into individual sentences and each sentence is provided its sentence score according to the occurrence of the high-scored.

3. **Scoring and Selection:** TF-IDF (Term Frequency — Inverse Document Frequency) is one of the Word frequency techniques which we will be used for scoring and selection in our project. TF-IDF stands for the Term Frequency Inverse Document Frequency of records. It can be defined as the calculation of how relevant a word in a series or corpus is to a text. The meaning increases proportionally to the number of times in the text a word appears but is compensated by the word frequency in the corpus (dataset). TF-IDF (Term Frequency — Inverse Document Frequency) gives weights to individual words based on their uniqueness compared to the document's overall vocabulary. Words with higher weights (more unique) often have more importance or provide more meaning to the document.

4. **Final Summarization:**



After all these steps and the process of making a summary of any text. A summary is a crisp statement or restatement of major points, especially as a conclusion to a work, it is comprehension and usually a brief extract, abstract, or recapitulation of previously stated facts or statements. To summarize means, to sum up, t, the main points of something- a summarization is the kind of summation of a large document or huge amount of text. And Text summarization is the process in which a long piece of text gets a crisp format with a lesser number of words than the actual text still reflecting the same meaning as the original doc/text. Finally, now we will get our final summary of the input we gave after all the steps are followed

Result Analysis:

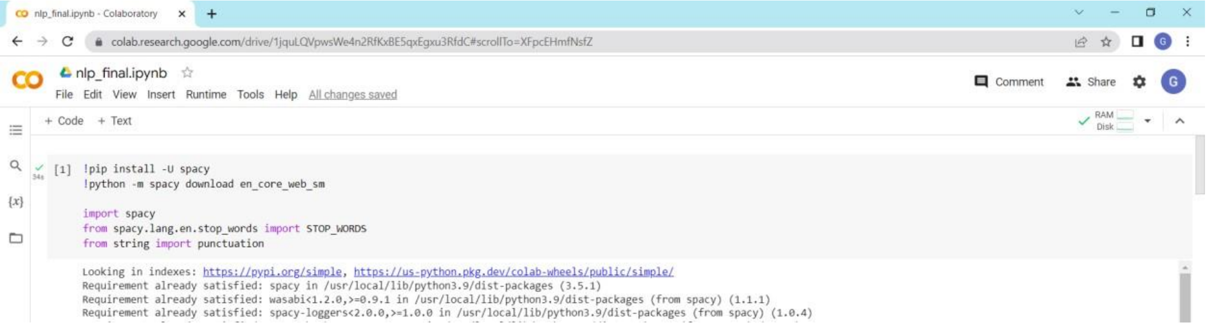
We have Used two approaches

1.Word frequency

2. Text rank algorithm using genism

1) Approach -1

Importing all the required libraries



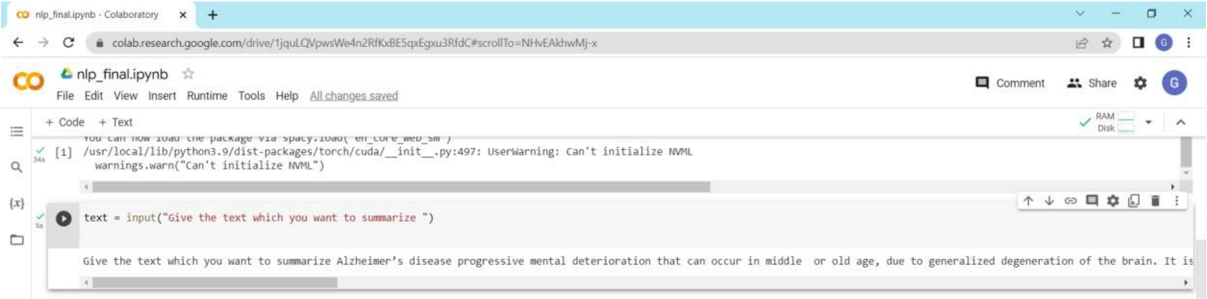
nlp_final.ipynb - Colaboratory
colab.research.google.com/drive/1jquLQVpwsWe4n2RfKxBE5qxEgxu3RfdC#scrollTo=XFpcEHmfNsfZ
nlp_final.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk

```
[1] !pip install -U spacy
    !python -m spacy download en_core_web_sm

    import spacy
    from spacy.lang.en.stop_words import STOP_WORDS
    from string import punctuation
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: spacy in /usr/local/lib/python3.9/dist-packages (3.5.1)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.9/dist-packages (from spacy) (1.1.1)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.9/dist-packages (from spacy) (1.0.4)
```

Taking the user input of the text

Input text-Alzheimer’s disease is progressive mental deterioration that can occur in middle or old age, due to generalized degeneration of the brain. It is the commonest cause f premature senility. It is currently ranked as the sixth leading cause of death in the United States, but recent estimates indicate that the disorder may rank third, just behind heart disease and cancer, as a cause of death for older people. The causes of dementia can vary, depending on the types of brain changes at may be taking place. Other dementias include Lewy body dementia, front temporal disorders, and vascular dementia. It is common for people to have mixed dementia- a combination of two or more types of dementia. Alzheimer’s is the most common cause of dementia among older adults

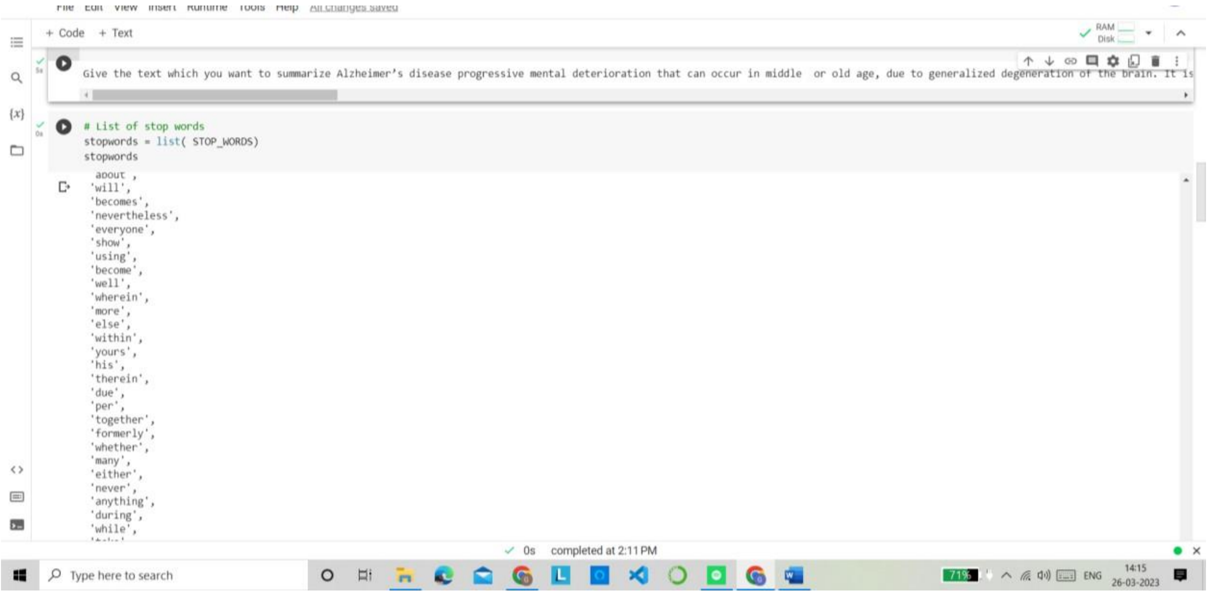

nlp_final.ipynb - Colaboratory
colab.research.google.com/drive/1jquLQVpwsWe4n2RfKxBE5qxEgxu3RfdC#scrollTo=NHvEAkhwMj-x
nlp_final.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk

```
[1] /usr/local/lib/python3.9/dist-packages/torch/cuda/__init__.py:497: UserWarning: Can't initialize NVML
      warnings.warn("Can't initialize NVML")
```

```
text = input("Give the text which you want to summarize ")
```

```
Give the text which you want to summarize Alzheimer’s disease progressive mental deterioration that can occur in middle  or old age, due to generalized degeneration of the brain. It is
```

Listing stop words

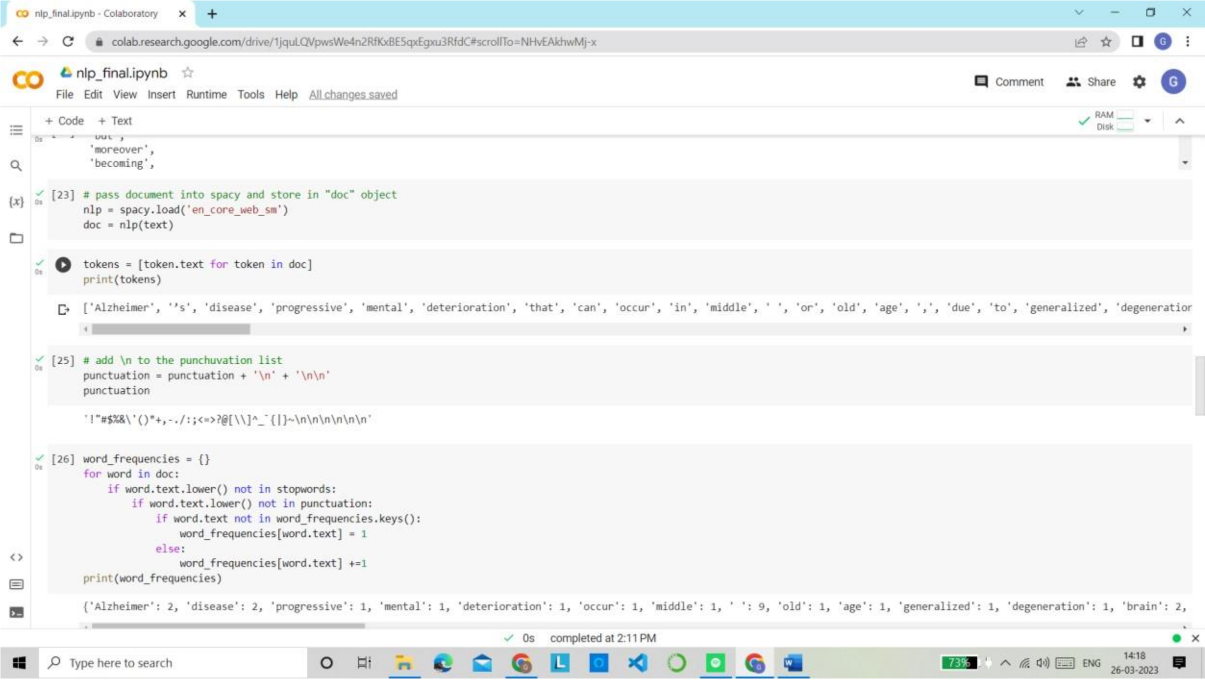

File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
RAM
Disk
✓ 0s completed at 2:11 PM
Type here to search
71%
ENG
14:15
26-03-2023

```
Give the text which you want to summarize Alzheimer’s disease progressive mental deterioration that can occur in middle  or old age, due to generalized degeneration of the brain. It is
```

```
# List of stop words
stopwords = list( STOP_WORDS)
stopwords
```

```
 'about',
 'will',
 'becomes',
 'nevertheless',
 'everyone',
 'show',
 'using',
 'become',
 'well',
 'wherein',
 'more',
 'else',
 'within',
 'yours',
 'his',
 'therein',
 'due',
 'per',
 'together',
 'formerly',
 'whether',
 'many',
 'either',
 'never',
 'anything',
 'during',
 'while',
```

Passing the document into Spacy and storing it in the ‘doc’ object and adding \n to the punctuation list



nlp_final.ipynb - Colaboratory
colab.research.google.com/drive/1jquLQVpwsWe4n2RfKxBE5qxEgxu3RfdC#scrollTo=NHvEAkhwMj-x
nlp_final.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk

```
    'moreover',
    'becoming',
```

```
[23] # pass document into spacy and store in "doc" object
     nlp = spacy.load('en_core_web_sm')
     doc = nlp(text)
```

```
tokens = [token.text for token in doc]
print(tokens)
```

```
['Alzheimer', '’s', 'disease', 'progressive', 'mental', 'deterioration', 'that', 'can', 'occur', 'in', 'middle', ' ', 'or', 'old', 'age', ',', 'due', 'to', 'generalized', 'degeneratio
```

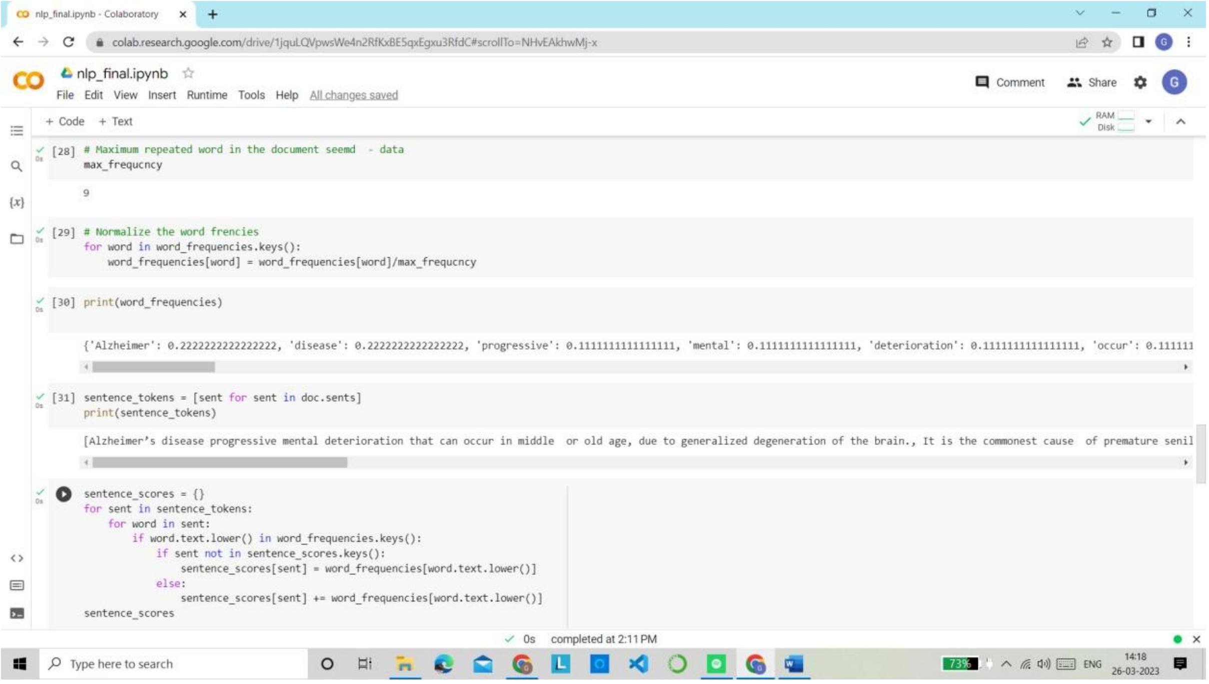
```
[25] # add \n to the punchuvation list
     punctuation = punctuation + '\n' + '\n\n'
     punctuation
```

```
'!"#$%&\'()"+,-./:;<=>?@[\\]^_`{|}~\n\n\n\n\n\n'
```

```
[26] word_frequencies = {}
     for word in doc:
         if word.text.lower() not in stopwords:
             if word.text.lower() not in punctuation:
                 if word.text not in word_frequencies.keys():
                     word_frequencies[word.text] = 1
                 else:
                     word_frequencies[word.text] +=1
     print(word_frequencies)
```

```
{'Alzheimer': 2, 'disease': 2, 'progressive': 1, 'mental': 1, 'deterioration': 1, 'occur': 1, 'middle': 1, ' ': 9, 'old': 1, 'age': 1, 'generalized': 1, 'degeneration': 1, 'brain': 2,
```

Calculating the sentence scores



nlp_final.ipynb - Colaboratory
colab.research.google.com/drive/1jquLQVpwsWe4n2RfKxBE5qxEgxu3RfdC#scrollTo=NHvEAkhwMj-x
nlp_final.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk

```
[28] # Maximum repeated word in the document seemd  - data
     max_frequcncy
```

```
9
```

```
[29] # Normalize the word frencies
     for word in word_frequencies.keys():
         word_frequencies[word] = word_frequencies[word]/max_frequcncy
```

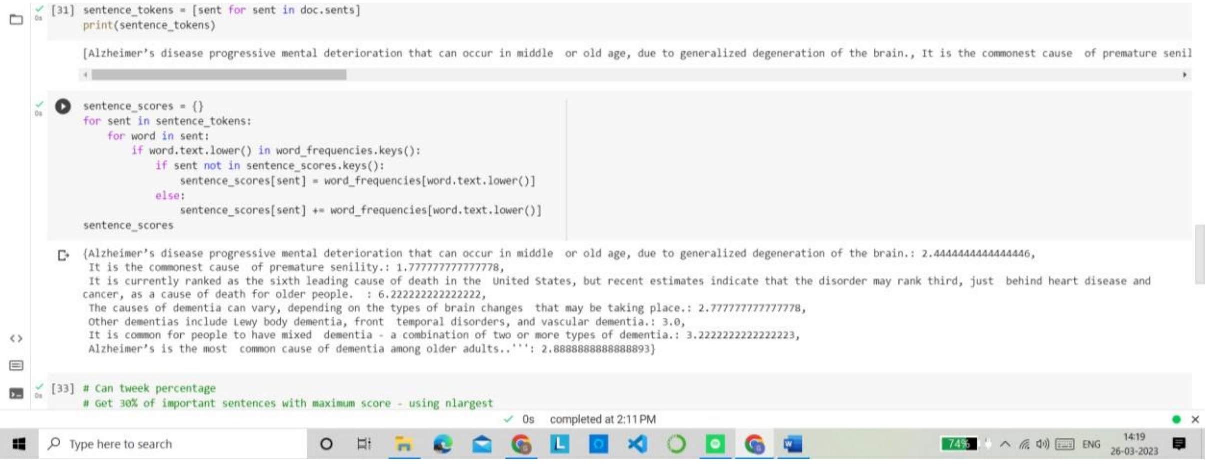
```
[30] print(word_frequencies)
```

```
{'Alzheimer': 0.2222222222222222, 'disease': 0.2222222222222222, 'progressive': 0.1111111111111111, 'mental': 0.1111111111111111, 'deterioration': 0.1111111111111111, 'occur': 0.111111
```

```
[31] sentence_tokens = [sent for sent in doc.sents]
     print(sentence_tokens)
```

```
[Alzheimer’s disease progressive mental deterioration that can occur in middle  or old age, due to generalized degeneration of the brain., It is the commonest cause  of premature senil
```

```
sentence_scores = {}
for sent in sentence_tokens:
    for word in sent:
        if word.text.lower() in word_frequencies.keys():
            if sent not in sentence_scores.keys():
                sentence_scores[sent] = word_frequencies[word.text.lower()]
            else:
                sentence_scores[sent] += word_frequencies[word.text.lower()]
sentence_scores
```



```
[31] sentence_tokens = [sent for sent in doc.sents]
     print(sentence_tokens)

     [Alzheimer’s disease progressive mental deterioration that can occur in middle  or old age, due to generalized degeneration of the brain., It is the commonest cause  of premature senil
```

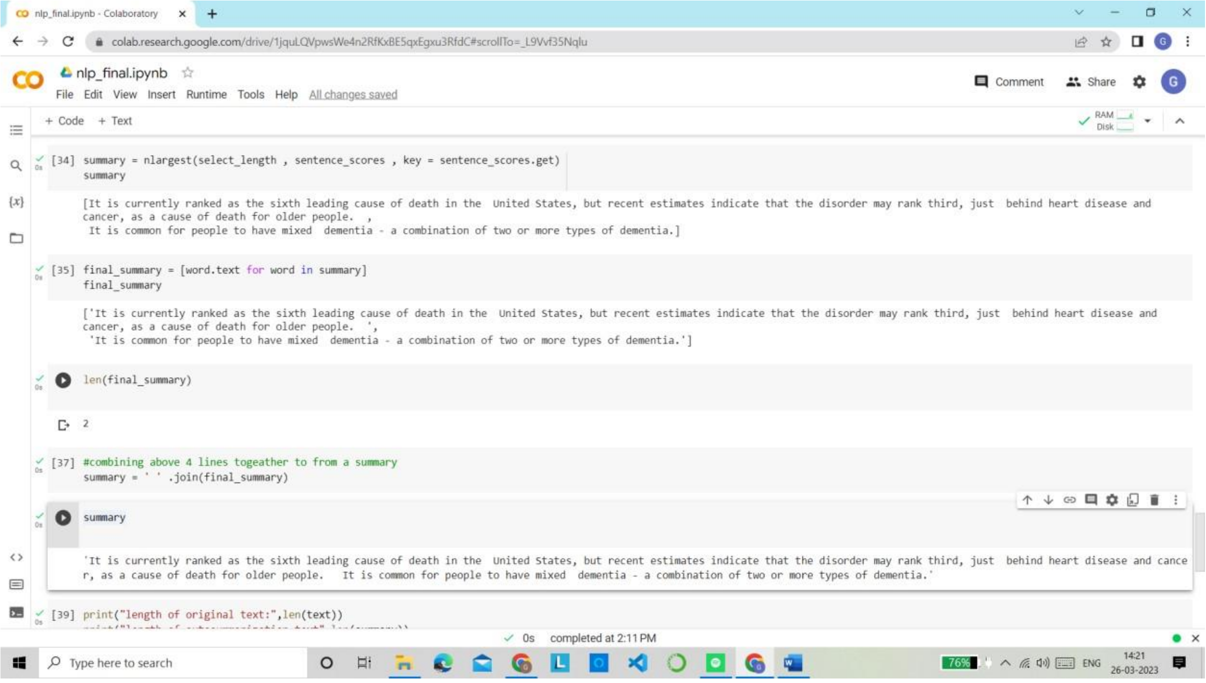
```
sentence_scores = {}
for sent in sentence_tokens:
    for word in sent:
        if word.text.lower() in word_frequencies.keys():
            if sent not in sentence_scores.keys():
                sentence_scores[sent] = word_frequencies[word.text.lower()]
            else:
                sentence_scores[sent] += word_frequencies[word.text.lower()]
sentence_scores
```

```
{Alzheimer’s disease progressive mental deterioration that can occur in middle  or old age, due to generalized degeneration of the brain.: 2.4444444444444446,
 It is the commonest cause  of premature senility.: 1.7777777777777778,
 It is currently ranked as the sixth leading cause of death in the  United States, but recent estimates indicate that the disorder may rank third, just  behind heart disease and
cancer, as a cause of death for older people.  : 6.222222222222222,
 The causes of dementia can vary, depending on the types of brain changes  that may be taking place.: 2.7777777777777778,
 Other dementias include Lewy body dementia, front  temporal disorders, and vascular dementia.: 3.0,
 It is common for people to have mixed  dementia - a combination of two or more types of dementia.: 3.2222222222222223,
 Alzheimer’s is the most  common cause of dementia among older adults..''': 2.8888888888888893}
```

```
[33] # Can tweek percentage
     # Get 30% of important sentences with maximum score - using nlargest
```

Printing the summary: -

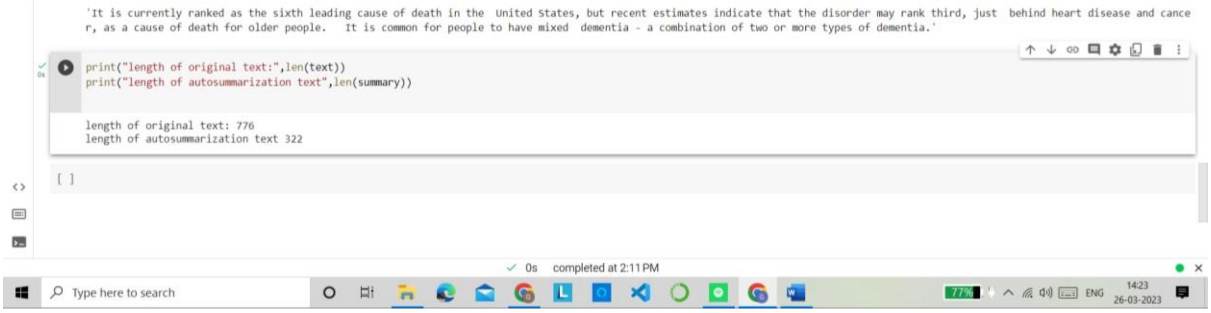
Summary-It is currently ranked as the sixth leading cause of death in the United States, but recent estimates indicate that the disorder may rank third, just behind heart disease and cancer, as a cause of death for older people.   It is common for people to have mixed dementia - a combination of two or more types of dementia.



```
[34] summary = nlargest(select_length , sentence_scores , key = sentence_scores.get)
     summary

     [It is currently ranked as the sixth leading cause of death in the  United States, but recent estimates indicate that the disorder may rank third, just  behind heart disease and
     cancer, as a cause of death for older people.  ,
      It is common for people to have mixed  dementia - a combination of two or more types of dementia.]

[35] final_summary = [word.text for word in summary]
     final_summary

     ['It is currently ranked as the sixth leading cause of death in the  United States, but recent estimates indicate that the disorder may rank third, just  behind heart disease and
     cancer, as a cause of death for older people.  ',
      'It is common for people to have mixed  dementia - a combination of two or more types of dementia.']

     len(final_summary)

     2

[37] #combining above 4 lines togeather to from a summary
     summary = ' ' .join(final_summary)

     summary

     'It is currently ranked as the sixth leading cause of death in the  United States, but recent estimates indicate that the disorder may rank third, just  behind heart disease and cance
     r, as a cause of death for older people.   It is common for people to have mixed  dementia - a combination of two or more types of dementia.'

[39] print("length of original text:",len(text))
```

nlp_final.ipynb - Colaboratory
colab.research.google.com/drive/1jquLQVpwsWe4n2RfKxBE5qxEgxu3RfdC#scrollTo=_L9Vvf35Nqlu
nlp_final.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
0s completed at 2:11 PM
Type here to search
76%
ENG
14:21
26-03-2023

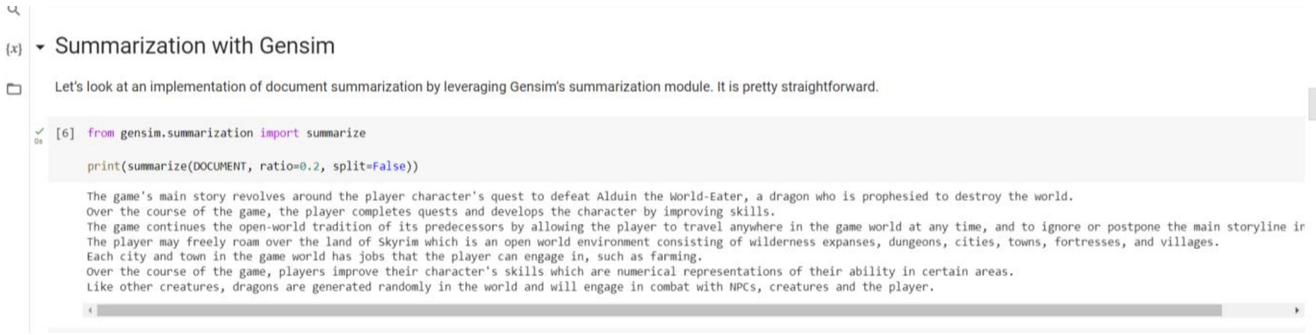
Now we can compare the length of text before auto summarization and after auto summarization



```
'It is currently ranked as the sixth leading cause of death in the  United States, but recent estimates indicate that the disorder may rank third, just  behind heart disease and cance
r, as a cause of death for older people.   It is common for people to have mixed  dementia - a combination of two or more types of dementia.'

print("length of original text:",len(text))
print("length of autosummarization text",len(summary))

length of original text: 776
length of autosummarization text 322

[ ]
```

2) Approach-2(The text Rank Algorithm)

Importing the library gensim



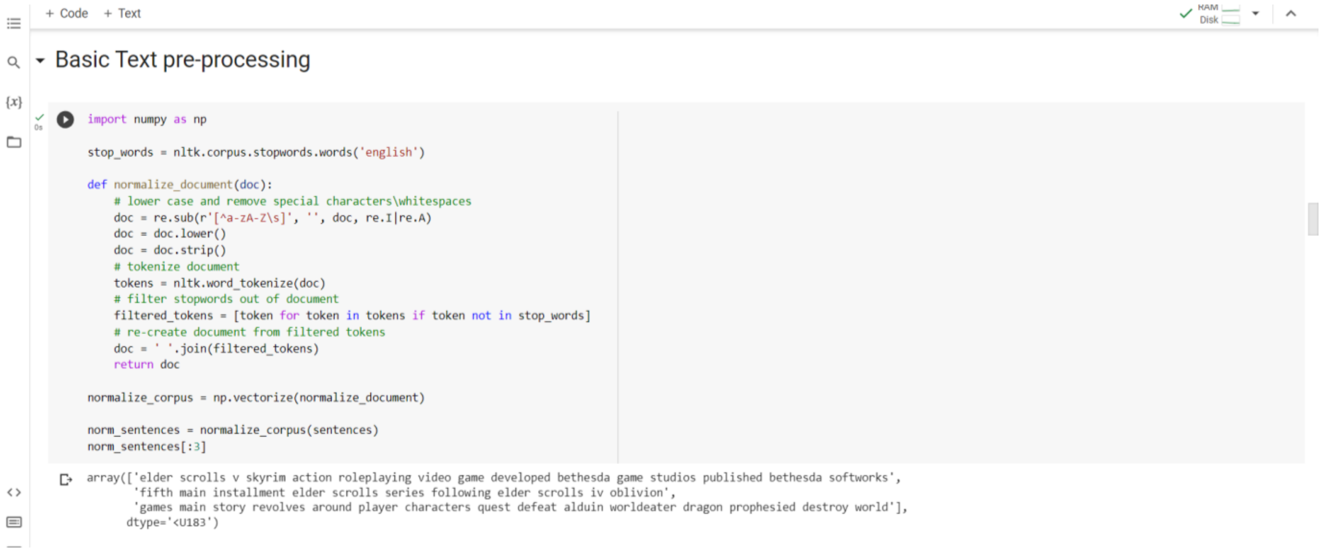
Summarization with Gensim

Let's look at an implementation of document summarization by leveraging Gensim's summarization module. It is pretty straightforward.

```
[6] from gensim.summarization import summarize

    print(summarize(DOCUMENT, ratio=0.2, split=False))
```

```
The game's main story revolves around the player character's quest to defeat Alduin the World-Eater, a dragon who is prophesied to destroy the world.
Over the course of the game, the player completes quests and develops the character by improving skills.
The game continues the open-world tradition of its predecessors by allowing the player to travel anywhere in the game world at any time, and to ignore or postpone the main storyline ir
The player may freely roam over the land of Skyrim which is an open world environment consisting of wilderness expanses, dungeons, cities, towns, fortresses, and villages.
Each city and town in the game world has jobs that the player can engage in, such as farming.
Over the course of the game, players improve their character's skills which are numerical representations of their ability in certain areas.
Like other creatures, dragons are generated randomly in the world and will engage in combat with NPCs, creatures and the player.
```

Applying basic text pre processing

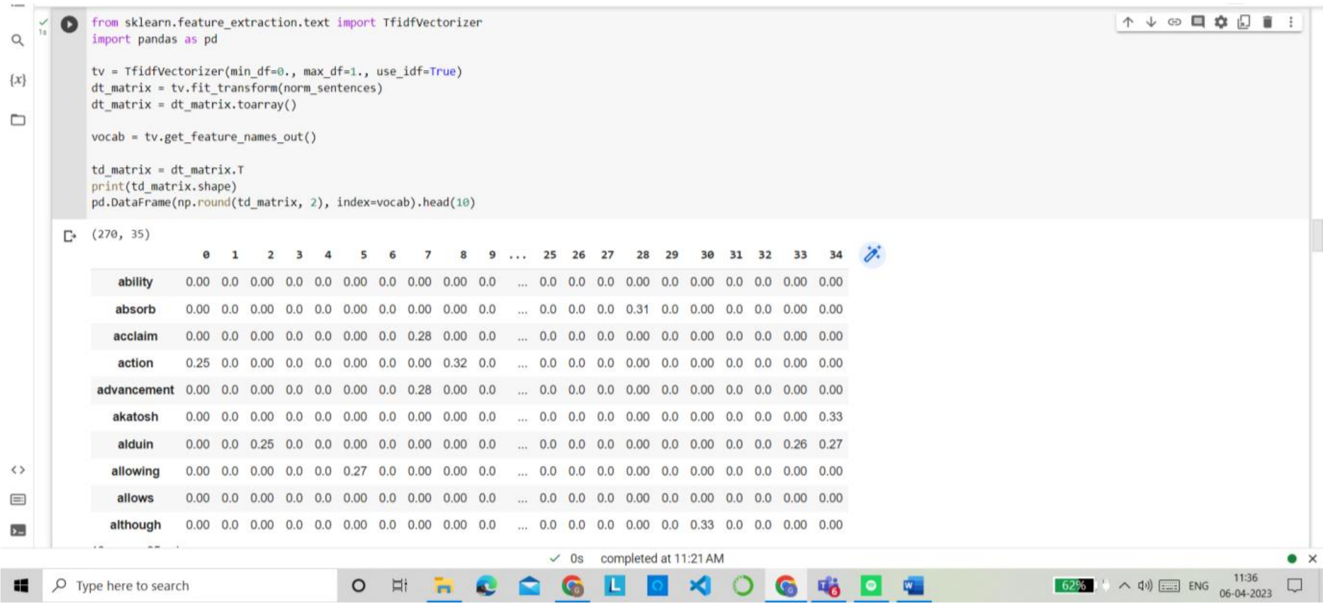


Basic Text pre-processing

```
import numpy as np

stop_words = nltk.corpus.stopwords.words('english')

def normalize_document(doc):
    # lower case and remove special characters\whitespaces
    doc = re.sub(r'[^a-zA-Z\s]', '', doc, re.I|re.A)
    doc = doc.lower()
    doc = doc.strip()
    # tokenize document
    tokens = nltk.word_tokenize(doc)
    # filter stopwords out of document
    filtered_tokens = [token for token in tokens if token not in stop_words]
    # re-create document from filtered tokens
    doc = ' '.join(filtered_tokens)
    return doc

normalize_corpus = np.vectorize(normalize_document)

norm_sentences = normalize_corpus(sentences)
norm_sentences[:3]
```

```
array(['elder scrolls v skyrim action roleplaying video game developed bethesda game studios published bethesda softworks',
       'fifth main installment elder scrolls series following elder scrolls iv oblivion',
       'games main story revolves around player characters quest defeat alduin worldeater dragon prophesied destroy world'],
      dtype='<U183')
```

We will be vectorizing our normalized sentences using the TF-IDF feature engineering scheme. We keep things simple and don't filter out any words based on document frequency. But feel free to try that out and maybe even leverage n-grams as features.



```
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd

tv = TfidfVectorizer(min_df=0., max_df=1., use_idf=True)
dt_matrix = tv.fit_transform(norm_sentences)
dt_matrix = dt_matrix.toarray()

vocab = tv.get_feature_names_out()

td_matrix = dt_matrix.T
print(td_matrix.shape)
pd.DataFrame(np.round(td_matrix, 2), index=vocab).head(10)
```

```
(270, 35)
```

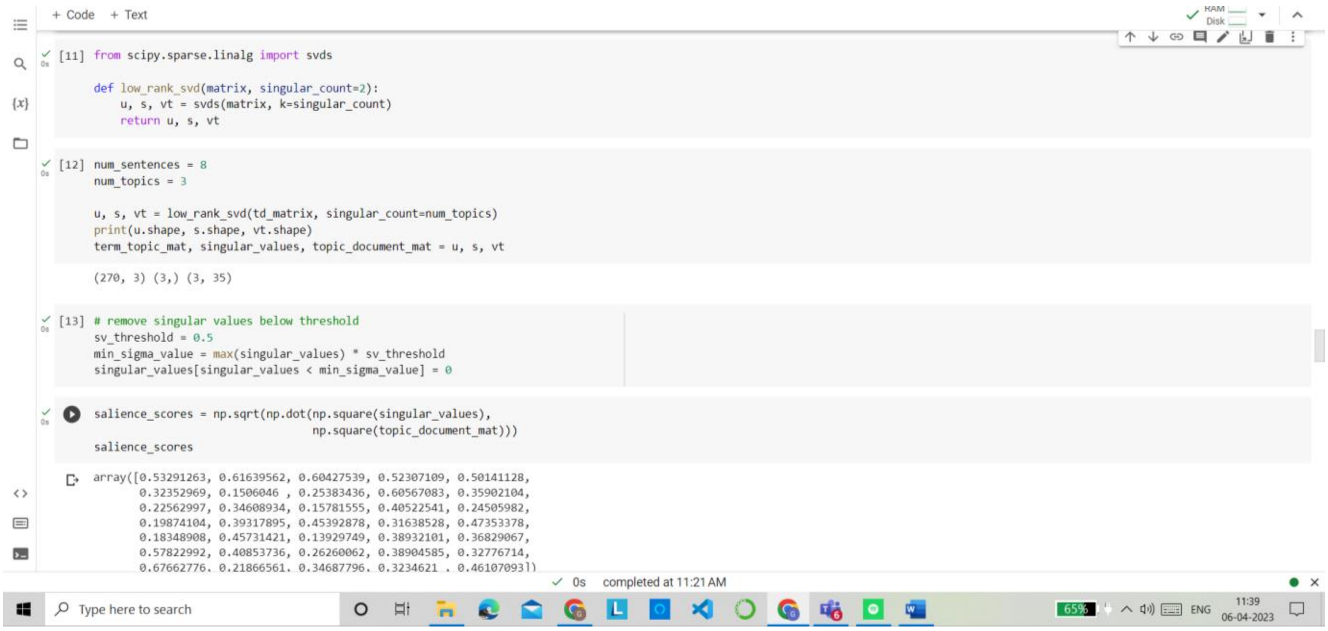
	0	1	2	3	4	5	6	7	8	9	...	25	26	27	28	29	30	31	32	33	34
ability	0.00	0.0	0.00	0.0	0.0	0.00	0.0	0.00	0.00	0.0	...	0.0	0.0	0.0	0.00	0.0	0.00	0.0	0.0	0.00	0.00
absorb	0.00	0.0	0.00	0.0	0.0	0.00	0.0	0.00	0.00	0.0	...	0.0	0.0	0.0	0.31	0.0	0.00	0.0	0.0	0.00	0.00
acclaim	0.00	0.0	0.00	0.0	0.0	0.00	0.0	0.28	0.00	0.0	...	0.0	0.0	0.0	0.00	0.0	0.00	0.0	0.0	0.00	0.00
action	0.25	0.0	0.00	0.0	0.0	0.00	0.0	0.00	0.32	0.0	...	0.0	0.0	0.0	0.00	0.0	0.00	0.0	0.0	0.00	0.00
advancement	0.00	0.0	0.00	0.0	0.0	0.00	0.0	0.28	0.00	0.0	...	0.0	0.0	0.0	0.00	0.0	0.00	0.0	0.0	0.00	0.00
akatosh	0.00	0.0	0.00	0.0	0.0	0.00	0.0	0.00	0.00	0.0	...	0.0	0.0	0.0	0.00	0.0	0.00	0.0	0.0	0.00	0.33
alduin	0.00	0.0	0.25	0.0	0.0	0.00	0.0	0.00	0.00	0.0	...	0.0	0.0	0.0	0.00	0.0	0.00	0.0	0.0	0.26	0.27
allowing	0.00	0.0	0.00	0.0	0.0	0.27	0.0	0.00	0.00	0.0	...	0.0	0.0	0.0	0.00	0.0	0.00	0.0	0.0	0.00	0.00
allows	0.00	0.0	0.00	0.0	0.0	0.00	0.0	0.00	0.00	0.0	...	0.0	0.0	0.0	0.00	0.0	0.00	0.0	0.0	0.00	0.00
although	0.00	0.0	0.00	0.0	0.0	0.00	0.0	0.00	0.00	0.0	...	0.0	0.0	0.0	0.00	0.0	0.33	0.0	0.0	0.00	0.00

Here, we summarize our game description by utilizing document sentences. The terms in each sentence of the document have been extracted to form the term-document matrix, which we observed in the previous cell.

We apply low-rank Singular Value Decomposition to this matrix. The core principle behind Latent Semantic Analysis (LSA) is that in any document, there exists a latent structure among terms that are related contextually and hence should also be correlated in the same singular space.

The main idea in our implementation is to use SVD (recall M = USVT) so that U and V are the orthogonal matrices and S is the diagonal matrix, which can also be represented as a vector of the singular values.

The original matrix can be represented as a term-document matrix where the rows are terms and each column is a document, i.e., a sentence from our document in this case. The values can be any type of weighting like Bag of Words model-based frequencies, TF-IDFs, or binary occurrences.



```
[11] from scipy.sparse.linalg import svds

     def low_rank_svd(matrix, singular_count=2):
         u, s, vt = svds(matrix, k=singular_count)
         return u, s, vt
```

```
[12] num_sentences = 8
     num_topics = 3

     u, s, vt = low_rank_svd(td_matrix, singular_count=num_topics)
     print(u.shape, s.shape, vt.shape)
     term_topic_mat, singular_values, topic_document_mat = u, s, vt
```

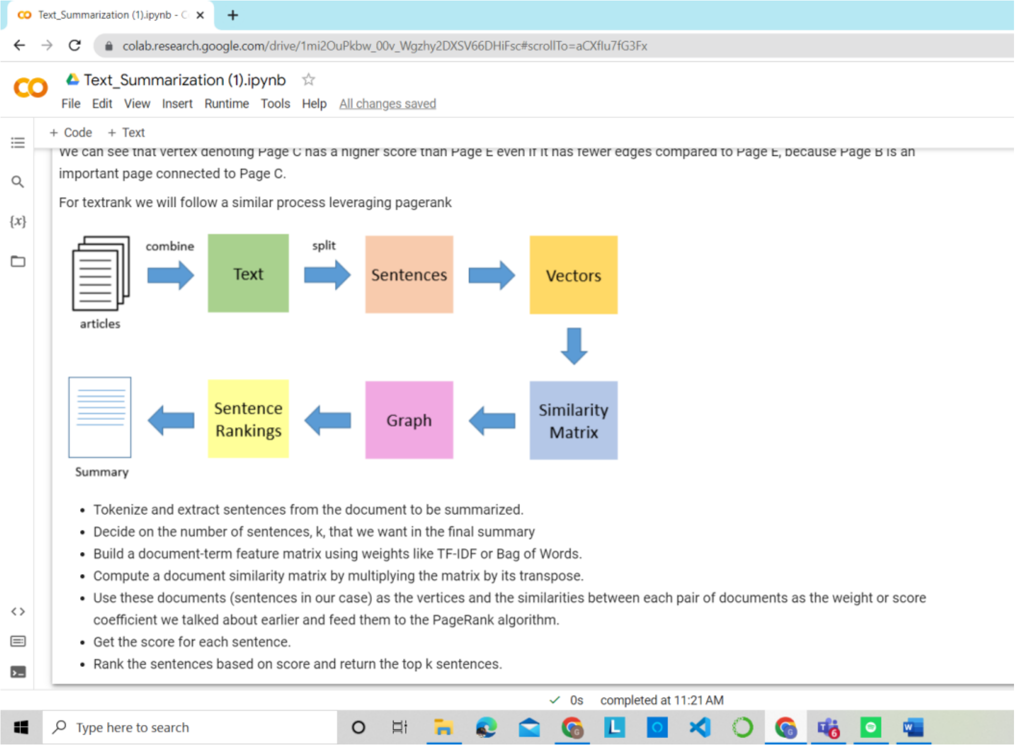
```
(270, 3) (3,) (3, 35)
```

```
[13] # remove singular values below threshold
     sv_threshold = 0.5
     min_sigma_value = max(singular_values) * sv_threshold
     singular_values[singular_values < min_sigma_value] = 0
```

```
salience_scores = np.sqrt(np.dot(np.square(singular_values),
                                 np.square(topic_document_mat)))
salience_scores
```

```
array([0.53291263, 0.61639562, 0.60427539, 0.52307109, 0.50141128,
       0.32352969, 0.1506046 , 0.25383436, 0.60567083, 0.35902104,
       0.22562997, 0.34608934, 0.15781555, 0.40522541, 0.24505982,
       0.19874104, 0.39317895, 0.45392878, 0.31638528, 0.47353378,
       0.18348908, 0.45731421, 0.13929749, 0.38932101, 0.36829067,
       0.57822992, 0.40853736, 0.26260062, 0.38904585, 0.32776714,
       0.67662776, 0.21866561, 0.34687796, 0.3234621 , 0.46107093])
```

- **The text rank algorithm**



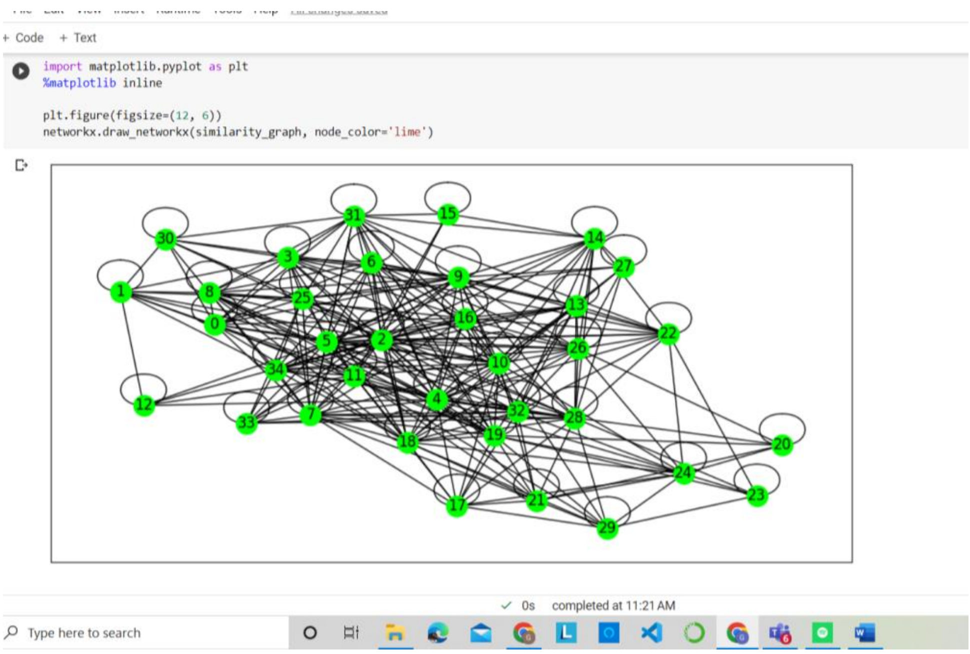
We can see that vertex denoting Page C has a higher score than Page E even if it has fewer edges compared to Page E, because Page B is an important page connected to Page C.

For textrank we will follow a similar process leveraging pagerank

Text_Summarization (1).ipynb
colab.research.google.com/drive/1mi2OuPkbw_00v_Wgzhy2DXSV66DHiFsc#scrollTo=aCXflu7fG3Fx
File Edit View Insert Runtime Tools Help All changes saved
+ Code
+ Text
combine
split
articles
Text
Sentences
Vectors
Similarity Matrix
Graph
Sentence Rankings
Summary
✓ 0s completed at 11:21 AM
Type here to search

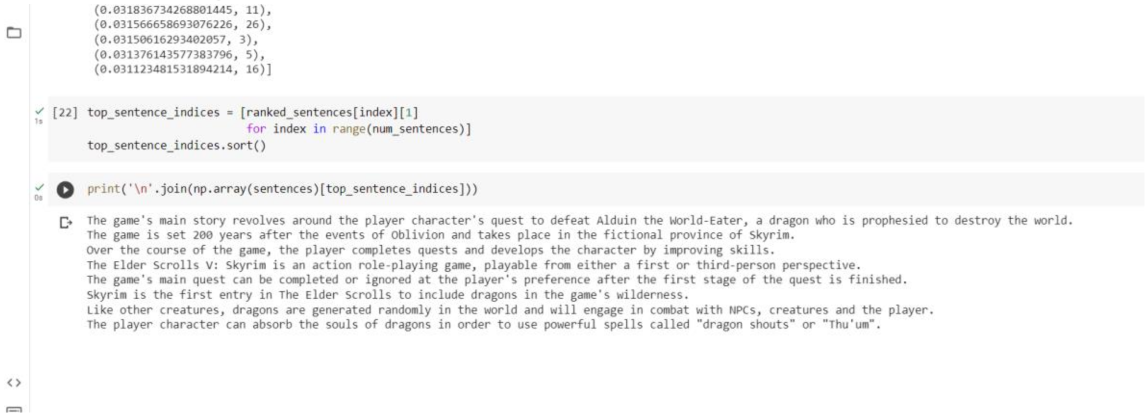
- Tokenize and extract sentences from the document to be summarized.
- Decide on the number of sentences, k, that we want in the final summary
- Build a document-term feature matrix using weights like TF-IDF or Bag of Words.
- Compute a document similarity matrix by multiplying the matrix by its transpose.
- Use these documents (sentences in our case) as the vertices and the similarities between each pair of documents as the weight or score coefficient we talked about earlier and feed them to the PageRank algorithm.
- Get the score for each sentence.
- Rank the sentences based on score and return the top k sentences.

- **Building the similarity graph**



```
import matplotlib.pyplot as plt
%matplotlib inline

plt.figure(figsize=(12, 6))
networkx.draw_networkx(similarity_graph, node_color='lime')
```

- **Getting the final summary**



```
 (0.031836734268801445, 11),
 (0.031566658693076226, 26),
 (0.03150616293402057, 3),
 (0.031376143577383796, 5),
 (0.031123481531894214, 16)]
```

```
[22] top_sentence_indices = [ranked_sentences[index][1]
                            for index in range(num_sentences)]
     top_sentence_indices.sort()
```

```
print('\n'.join(np.array(sentences)[top_sentence_indices]))
```

```
The game's main story revolves around the player character's quest to defeat Alduin the World-Eater, a dragon who is prophesied to destroy the world.
The game is set 200 years after the events of Oblivion and takes place in the fictional province of Skyrim.
Over the course of the game, the player completes quests and develops the character by improving skills.
The Elder Scrolls V: Skyrim is an action role-playing game, playable from either a first or third-person perspective.
The game's main quest can be completed or ignored at the player's preference after the first stage of the quest is finished.
Skyrim is the first entry in The Elder Scrolls to include dragons in the game's wilderness.
Like other creatures, dragons are generated randomly in the world and will engage in combat with NPCs, creatures and the player.
The player character can absorb the souls of dragons in order to use powerful spells called "dragon shouts" or "Thu'um".
```

Conclusion-

Spacy is a powerful natural language processing (NLP) tool that can be used for text summarization. With its efficient tokenization, part-of-speech tagging, and named entity recognition capabilities, Spacy can identify key sentences and phrases in a text, making it ideal for summarization tasks. Spacy also has a wide range of pre-trained models, which can be fine-tuned for specific summarization tasks. Overall, Spacy is a valuable tool for anyone looking to automate the process of summarizing large amounts of text, whether it be for research, journalism, or other applications.

Text summaries are useful for natural languages processing tasks such as question and answer or other related fields of computer science such as text classification and data retrieval. And access time for information search will be improved. At the same time, sequencing enhances the effect and its algorithms are less biased than human creams. Using a text summary system, commercial capture services allow users to increase the number of texts they can process.

Future Scope: -

potential future scopes of text summarization in NLP using Spacy:

1. **Multi-document summarization:** Currently, most text summarization models are designed to summarize single documents. However, as the amount of information available online continues to grow, there will be an increasing need for summarization models that can handle multiple documents at once.

2. **Customization of summaries:** Currently, most summarization models produce generic summaries that may not be suitable for all users or contexts. In the future, there may be more focus on developing models that can produce customized summaries that are tailored to the specific needs of individual users or organizations.

3. **Domain-specific summarization:** There is potential for developing domain-specific summarization models that are trained on text from specific domains, such as medical or legal documents. These models could produce more accurate and relevant summaries for users working in these fields.

4. **Summarization of audio and video content:** While text summarization is currently the most common form of summarization, there may be increasing demand for models that can summarize audio and video content as well. This would require the development of new techniques and models that can extract key information from audio and video sources.

5. **Multilingual summarization**: As the world becomes increasingly globalized, there will be a growing need for summarization models that can handle text in multiple languages. Spacy is already capable of handling multiple languages, but there is potential for further improvements in this area.

6. **Interpretability and explainability:** As summarization models become more complex, there will be a growing need for models that are interpretable and explainable. This will be important for ensuring that users can understand

how the models are making their decisions and for building trust in the technology.

References:

https://www.topcoder.com/thrive/articles/text-summarization-in-nlp

https://www.analyticsvidhya.com/blog/2021/11/a-beginners-guide-to-understanding-textsummarization-with-nlp/

https://www.machinelearningplus.com/nlp/text-summarization-approaches-nlp-example/

https://www.numpyninja.com/post/text-summarization-through-use-of-spacy-library

https://ieeexplore.ieee.org/document/9404712

https://jcharistech.wordpress.com/2018/12/31/text-summarization-using-spacy-and-python/

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

https://www.researchgate.net/publication/362581063_Implementation_of_NLP_based_automatic_text_summarization_using_spacy

https://ieeexplore.ieee.org/document/9404712