2022

IOMP Project Report on

AI Based Intelligent Insight Extractor Using IBM Watson

SUBMITTED BY

N.AMULYA	**19UK1A0512**
V.RAHUL	**19UK1A0511**
M.MAHESH	**19UK1A0502**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)

Bollikunta, Warangal – 506005

2019– 2023

CONTENTS

CHAPTER 1

INTRODUCTION

Manually generating a summary can be time-consuming and tedious. Automatictext summarization promises to overcome such difficulties and allow you to generate the key ideas in a piece of writing easily.Text summarization in NLP means telling a long story in short with a limited number of words and conveying an important message in brief. There can be many strategies to make the large message short and give the most important information forward, one of them is calculating word frequencies and then normalizing the word frequencies by dividing by the maximum frequency. After that find the sentences withhigh frequencies and take the most important sentences to convey the message.

1.1 OVERVIEW

This AI based intelligent insight extractor is a text summarization using NLP. Text summarization in NLP means telling a long story in short with a limited number of words andconveying an important message in brief. There can be many strategies to make the large message short and give the most important information forward, one of them is calculating word frequencies and then normalizing the word frequencies by dividing by the maximum frequency. After that find the sentences with high frequencies and take the most important sentences to convey the message.The strategies used to make long story into a short story is text pre-processing whichincludes stopwords, load spacy language pipeline, word tokenization, word frequency, normalization, sentence tokenization, sentence score, and finally summary.

Finally text summarization of long stories which contains more than 3 paragraphs aresummarized into short story less than a paragraph using NLP

1.2 PURPOSE

The main objectives of the project is:

- Know fundamental concepts and techniques used for NLP.
- Gain a broad understanding of spacy package.
- Gain knowledge on pre-processing the text data.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

To complete this project, you must require the following software, concepts, andpackages.

1.Anaconda navigator and pycharm.

Anaconda Navigator is a free and open-source distribution of the Pythonand R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS. Conda is an open-source, cross-platform, package management system. Anaconda comes with tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code.

2.2 PURPOSED SOLUTION

The method or solution is Jupiter notebook and spyder we used to complete this project. and you will use this jupiter notebook for you recommended.

To build Machine learning models you must require the following packages

Sklearn: Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms.

NumPy: NumPy is a Python package that stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object

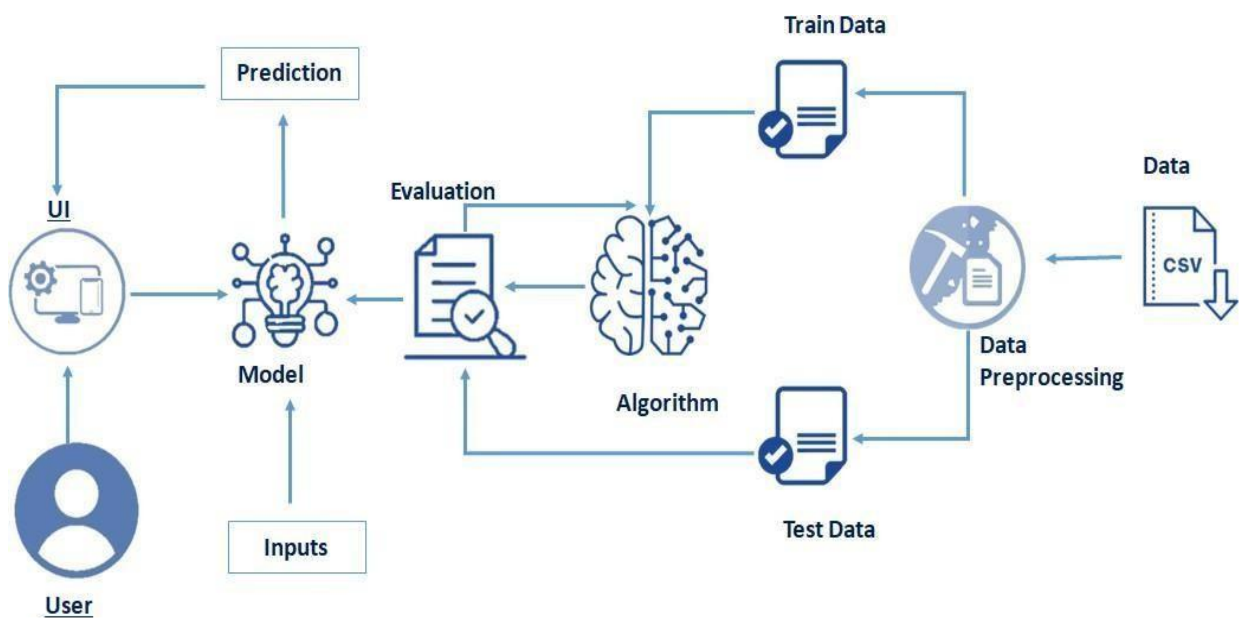
Pandas: pandas is a fast, powerful, flexible, and easy to use open source data analysis and manipulation tool,built on top of the Python programming language.

Matplotlib: It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits

CHAPTER 3

THEORETICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2 HARDWARE / SOFTWARE DESIGNING

The hardware required for the development of this project is:

Processor : Intel CoreTM i5-9300H
Processor speed : 2.4GHz
RAM Size : 8 GB DDR
System Type : X64-based processor

SOFTWARE DESIGNING:

The software required for the development of this project is:

Desktop GUI : Anaconda Navigator
Operating system : Windows 10
Front end : HTML, CSS,JAVASCRIPT
Programming : PYTHON
Cloud Computing Service : IBM Cloud Services

CHAPTER 4

EXPERIMENTAL INVESTIGATION

IMPORT REQUIRED LIBRARIES:

Import the necessary libraries as shown in the figure.

- Spacy is an open-source software library for advanced natural languageprocessing.
- Package string implements simple functions to manipulate UTF-8 encodedstrings.

READ DATASET:

For this project, we make use of three different datasets (Books_Ratings, Books, Users). In this project, our input is text data. Text data are called unstructured data.A variable 'doc' is created and the text is passed to that variable as the data.We can use Any article text as the data and give it to the variable as an input.

CHAPTER 5

FLOWCHART

PROJECT FLOW:

- The user interacts with the UI to enter the input.
- Entered input is analyzed by the model which is integrated.
- Once the model analyses the input the summary is showcased on the UI.

To accomplish this, we have to complete all the activities listed below,

- Import required libraries
  1.Read dataset

- Text pre-processing
  1.Stopwords
  2.Load spacy language pipeline
  3.Word tokenization
  4.Word frequency
  5.Normalization
  6.Sentence tokenization
  7.Sentence score
  8.Summary
- Application building
  1.Create an html file
  2.Build python code

TEXT PREPROCESSING:

In this milestone, we will be pre-processing the text that is collected. That includes

1. STOPWORDS:

Stop words are a set of commonly used words in all languages. Stop Words are used to eliminate unimportant words from data. So, the application/model can focus onlyon important text data. The list of stopwords in the English language is shown below.

2.LOADSPACY LANGUAGE PIPELINE:

The en_core_web_sm is a small English pipeline trained on written web text (blogs, news, comments), that includes vocabulary, syntax, and entities. There arefour different pipelines available in spacy in our project we are using the small pipeline.

The load() method from spacy is used to load the pipeline.

3.WORD TOKENIZATION:

The process of splitting the paragraph, sentence, etc from text docs is called wordtokenization

With the help of a list comprehension word, tokenization is done.(refer below image)

4.WORD FREQUENCY:

Word frequency is used to find the recurring/repeated words in the text data.The empty dictionary is created.

Words are passed as keys for thedictionary and the counts of the words are passed as values with the help of for loop and condition

5.NORMALIZATION:

Normalization refers to rescaling real numerical value attributes into 0 to 1. Data normalization in this project is used to make model training less sensitive to the scale of features.For text data normalization we need the maximum repeating value counts. So, word frequency max values are saved in a variable called max frequency.Then we are dividing the word frequency dictionary values by max frequencies.

6.SENTENCE TOKENIZATION:

Sentence tokenization is the process of splitting text into sentences.In this project, text summarization is done with the help of a sentence score. To calculate the sentence score, sentence tokenization is doneby list comprehension

7. SENTENCE SCORE:

An empty dictionary sentence score is created. With the help of for loop and if conditions the normalized word frequencies values are summed up for the sentence and the score is passed to the sentence score values. To visualize the output refer to the below image.

8.SUMMARY:

30% of the input sentence length is considered as summary length. So the length of theinput is multiplied with 0.3.The nlargest() function of the python module heapq returns the specified number oflargest elements from a python iterable like a list, tuple, and others.The function nlargest() can also be passed as a key function that returns acomparisionkey to be used in the sorting.
Refer to the below image to visualize the output.
To join the sentence list comprehension and join() method is used.

APPLICATION BUILDING:

In this section, we will be building a web application that is integrated into the model webuilt. A UI is provided for the uses where he/she has to enter the text for a summary. Then the summary is showcased on the UI.

This section has the following tasks

- Building HTML pages
- Building server-side script

BUILDING HTML PAGES:

For this project create 3 HTML files namely

- Home.html
- Summarize.html
- Submit.html

And save them in template folder.

BUILD PYTHON CODE:

Import the libraries

Importing the flask module into the project is mandatory. An object of the Flaskclass is ourWSGI application.

Flask constructor takes the name of the current module(--name--) as anargument.Render HTML page:Here we will be using the declared constructor to route to the HTML page that wehave created. In the above example, the'/' URL is bound with the home.html function. Hence,when the home page of the web server is opened in the browser, the HTML page will be rendered. Whenever you Enter the values from the HTML page the values can be retrieved using the POST method.

Retrieves the summary from UI:

Copy and paste the code of the model building.

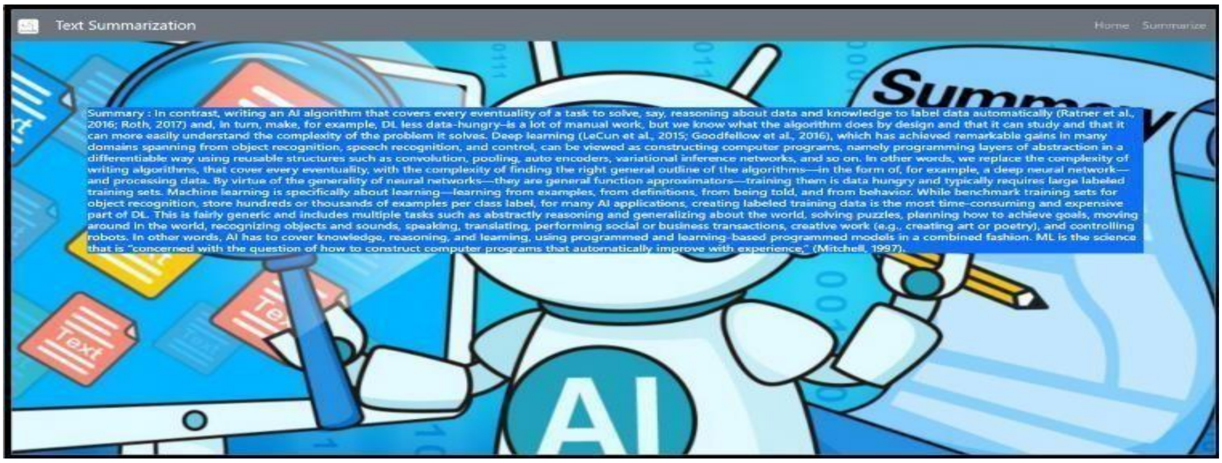
Main function.

RUN THE APPLICATION:

- Open the anaconda prompt from the start menu.
- Navigate to the folder where your python script is.
- Now type the "python app.py" command.
- Navigate to the localhost where you can view your web page.
- Click on the predict button from the top right corner, enter the inputs,click on thesubmit button, and see the result/prediction on the web.

CHAPTER 6

RESULT



CHAPTER 7

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

Easy to Understand: One major advantage of text summarization is that it makes it easier to understand large amounts of content.

Makes Information Easier to Find: Another benefit of text summarization is its ability to make information easier to find. By reducing the number of words needed to convey important ideas, text summarization helps reduce the amount of time spent searching through long documents or articles. Summaries designed to help convey information more easily are sometimes referred to as TLDR, or too long didn't read.

Helps Keep Up-to-Date: Another advantage of text summarization comes from the fact that it keeps users up-to-date with current events. This means that people who rely on news sources will be able to quickly access relevant information when they want it.

Increases User Engagement:A final advantage of text summarization lies in its ability to increase user engagement. When people read short summaries instead of lengthy ones, they tend to spend less time reading each piece and will typically read more as a result. This leads to higher levels of engagement.

DISADVANTAGES

May Miss Relevant Information:A disadvantage of text summarization is the possibility of missing relevant information. If the original piece of content was written by someone else, then the summary generator might omit key information the writer meant the reader to see. As a result, the generated summary may not be useful.

Can't Always Tell What Is Important: Another drawback of text summarization is its inability to tell what is most important about a particular article. For example, if a person wants to know more about specific examples given in the full article, he or she would have to go back and look at the full article.

Doesn't Work Well With Long Documents:A final disadvantage of summary generators are the lack of support for longer documents. Machine learning algorithms haven't been able to produce high-quality summaries for longer pieces of text because of hardware limitations. This is improving all the time and some summary generators are capable of producing high-quality text summaries for pieces as long as 5000-6000 words.

CHAPTER 8

APPLICATIONS

NLP based on epidemiological study

During the coronavirus outbreak in China, Alibaba's Damo Academy developed the StructBERT NLP model. This model, implemented in the Alibaba ecosystem, used not only the search engine on Alibaba's retail platforms, but also the analysis of anonymous medical data. Natural Language Processing application StructBERT has been also actively used to fight the coronavirus in many Chinese cities.

NLP based competitive analysis

There are many tools available to assist entrepreneurs in tracking down their competitors. Such an example is Zirra, an NLP-based engine that simplifies the process of automatically creating a competitive environment. The Zirra NLP application makes a list of companies and ranks them from zero to one. This list and rating are created by Zirra by scanning the Internet for articles and placing data in the natural language programming module.

NLP application tool – SignAll

SignAll has developed a technology that is able to recognize and translate sign language. It can be used in both business and education. Using advanced natural language processing and machine translation techniques, visual input is transformed into meaningful data for efficient sign language detection and translation. In fact, this NLP application tool can help individuals who are deaf communicate with those who don't know sign language.

NLP solution for optimization of contract processes

Natural language processing applications use the Python programming language to help businesses process contracts quickly. COIN (short for Contract Intelligence) is a text mining program created by JPMorgan Chase that can read and analyze commercial credit contracts. COIN can do document analysis by highlighting and extracting specific words or phrases. According to reports, this NLP application saves the company 360,000 hours each year.

NLP implementation in personal assistant

Digital personal assistants, such as Alexa, have grown in popularity in recent years. The success of these bots largely depends on the use of natural language programming and generation tools. Alexa can perform more than 70,000 skills. NLP and machine learning have played a key role in this rapid evolution. Currently, there are over 28,000 smart home devices that can be connected with Alexa. You can watch a short video about Alexa:

CHAPTER 9

CONCLUSION AND FUTURESCOPE

CONCLUSION

Automatic text summarization is an old challenge but the current research direction diverts towards emerging trends in biomedicine, product review, education domains, emails and blogs. This is due to the fact that there is information overload in these areas, especially on the World Wide Web.Automated summarization is an important area in NLP (Natural Language Processing) research. It consists of automatically creating a summary of one or more texts. The purpose of extractive document summarization is to automatically select a number of indicative sentences, passages, or paragraphs from the original document .Text summarization approaches based on Neural Network, Graph Theoretic, Fuzzy and Cluster have, to an extent, succeeded in making an effective summary of a document.Both extractive and abstractive methods have been researched. Most summarization techniques are based on extractive methods. Abstractive method is similar to summaries made by humans. Abstractive summarization as of now requires heavy machinery for language generation and is difficult to replicate into the domain specific areas.

FUTURESCOPE

In this thesis, we focused on summarization of news articles belonging to sports and technical domain. The techniques proposed here are adaptable across other domains. One of the future plans may be to apply the topic-focused summarization framework to news articles or blogs and to extend the work in the machine leaning approaches. Topic-focused summaries of news articles would be lot more accurate and valuable to users. It would be more interesting to work on topic modeling and summarization in the domain of social media in future.The rate at which the information is growing is tremendous. Hence it is very important to build a multilingual summarization system and this research could be a stepping stone towards achieving that goal provided there is availability of online lexical databases in other languages. The work presented by the thesis can also be applicable to multi document summarization by using minimal extensions. The thesis has used evaluation metrics Precision, Recall and F-measure to measure performance gain over existing systems with ROUGE tool. In future work, new metrics can be investigated which can be used in automatic evaluation environment to measure the overall quality such as grammar, readability, prominence and relativeness. The state of the art summarization systems are all extractive in nature, but the community is gradually progressing towards abstractive summarization. Although a complete abstractive summarization would require deeper natural language understanding and processing, a hybrid or shallow abstractive summarization can be achieved through sentence compression and textual entailment techniques. Textual entailment helps in detecting shorter versions of text that entail with same meaning as original text. With textual entailment we can produce more concise and shorter summaries.The Implemented system in this thesis can work as framework for the research community to understand and extend the applicability of cognitive and symbolic approach in various domains of business needs. Research in summarization continues to enhance the diversity and information richness, and strive to produce coherent and focused answers to users information need.

CHAPTER 10

BIBILOGRAPHY

1. **Wenbin Wei** and **Mark Hansen** used a logarithmic function for estimating thedemand of passenger air traffic based on the independent variables: flight frequencies, number of seats in the aircraft, ticket price, flight distance.etc.
   https://www.researchgate.net/publication/257424511_A_comment_on_W_Wei_and_M_Hansen's_benefit_estimation_in_An_aggregate_demand_model_for_air_passenger_traffic_in_the_hub-and-spoke_network

2. The **mixed multinomial logit model (MMNL) by Stephane Hess, John W.Polak, 2005** has also been used with good results to analyze air travel behaviour in airport choice.
   https://www.researchgate.net/publication/23731362_An_analysis_of_airport-choice_behaviour_using_the_Mixed_Multinomial_Logit_model

3. The multivariate regression model is the **ARIMAX** model (**Wai Hong Kan Tsui,Hatice Ozer Balli, Andrew Gilbey, Hamish Gow, 2014**) can improve the forecastaccuracy.
   https://www.researchgate.net/publication/259121979_Forecasting_of_Hong_Kong_airport's_passenger_throughput

APPENDIX

A Source Code of Flask:

```
# Importing the required packages

import spacy
from spacy.lang.en.stop_words import STOP_WORDS
from string import punctuation
import numpy as np
```

```
# Passing the input data

doc = """Machine learning (ML) and artificial intelligence (AI) are becoming dominant problem-solving techniques in many areas of

Big Data is no fad. The world is growing at an exponential rate and so is the size of the data collected across the globe. Data i

However, is AI, ML, and DL really synonymous, as recently suggested in the news, blogs, and media? For example, when AlphaGo (Si

print(doc)
```

```
Machine learning (ML) and artificial intelligence (AI) are becoming dominant problem-solving techniques in many areas of resear
ch and industry, not least because of the recent successes of deep learning (DL). However, the equation AI=ML=DL, as recently s
uggested in the news, blogs, and media, falls too short. These fields share the same fundamental hypotheses: computation is a u
seful way to model intelligent behavior in machines. What kind of computation and how to program it? This is not the right ques
tion. Computation neither rules out search, logical, and probabilistic techniques, nor (deep) (un)supervised and reinforcement
learning methods, among others, as computational models do include all of them. They complement each other, and the next breakt
hrough lies not only in pushing each of them but also in combining them.

Big Data is no fad. The world is growing at an exponential rate and so is the size of the data collected across the globe. Data
is becoming more meaningful and contextually relevant, breaking new grounds for machine learning (ML), in particular for deep l
earning (DL) and artificial intelligence (AI), moving them out of research labs into production (Jordan and Mitchell, 2015). Th
e problem has shifted from collecting massive amounts of data to understanding it—turning it into knowledge, conclusions, and a
ctions. Multiple research disciplines, from cognitive sciences to biology, finance, physics, and social sciences, as well as ma
ny companies believe that data-driven and “intelligent” solutions are necessary to solve many of their key problems. High-throu
ghput genomic and proteomic experiments can be used to enable personalized medicine. Large data sets of search queries can be u
sed to improve information retrieval. Historical climate data can be used to understand global warming and to better predict we
ather. Large amounts of sensor readings and hyperspectral images of plants can be used to identify drought conditions and to ga
in insights into when and how stress impacts plant growth and development and in turn how to counterattack the problem of world
hunger. Game data can turn pixels into actions within video games, while observational data can help enable robots to understan
d complex and unstructured environments and to learn manipulation skills.

However, is AI, ML, and DL really synonymous, as recently suggested in the news, blogs, and media? For example, when AlphaGo (S
ilver et al., 2016) defeated South Korean Master Lee Se-dol in the board game Go in 2016, the terms AI, ML, and DL were used by
the media to describe how AlphaGo won. In addition to this, even Gartner's list (Panetta, 2017) of top 10 Strategic Trends for
2018 places (narrow) AI at the very top, specifying it as “consisting of highly scoped machine-learning solutions that target a
specific task.
```

```
# To visualize the stop words

stopWords = list(STOP_WORDS)
print(stopWords)
```

```
['other', 'not', 'show', 'whoever', 'thereby', 'indeed', 'another', 'fifty', 'hence', 'below', 'whether', 'where', 'hereafter',
'should', 'your', 'own', 'formerly', 'go', "'ve", 'some', 'rather', '‘ll', 'myself', 'top', 'might', 'others', 'get', 'above',
'eleven', 'nine', 'sixty', 'beyond', 'otherwise', 'down', 'all', 'during', 'along', 'here', 'therefore', 'us', 'give', 'his',
'an', 'few', 'to', 'former', 'thus', 'twenty', 'whatever', 'but', 'fifteen', 'throughout', 'via', 'ten', 'the', 'just', 'with',
'around', 'yourselves', 'you', 'often', "'d", 'without', 'which', 'amongst', 'about', 'what', 'wherever', 'thereupon', 'than',
'’m', 'those', 'two', 'sometimes', 'nobody', 'between', 'have', 'therein', 'onto', 'serious', 'already', 'everywhere', 'too',
'for', 'thence', 'by', 'they', 'perhaps', 'side', 'would', 'of', 'amount', 'how', 'themselves', 'less', 'am', 'there', 'up', 's
ometime', 'anyway', 'six', 'this', 'is', 'three', 'yours', 'out', 'towards', 'becomes', 'most', 'could', 'forty', "'re", 'excep
t', 'used', 'enough', '’ve', '’ll', 'her', 'behind', 'put', 'though', 'only', 'was', 'before', 'hereupon', 'why', 'until', 'nam
ely', 'herself', 'then', 'various', 'elsewhere', 'please', 'ca', 'itself', 'quite', 'after', 'one', 'cannot', 'several', 'befor
ehand', 'yourself', 'from', 'has', 'me', 'and', 'thereafter', 'wherein', 'are', 'a', 'since', 'when', 'anyhow', 'or', 'being',
'even', 'nothing', 'neither', 'part', 'regarding', 'four', 'beside', 'per', 'very', 'upon', 'at', 'although', 'still', 'eithe
r', 'ours', 'over', 'whose', 'on', 'five', 'can', 'full', 'in', 'thru', 'afterwards', 'doing', 'last', 'whereupon', "'ll", 'bac
k', 'first', 'eight', 'name', 'always', 'take', 'well', 'herein', 'them', 'their', 'now', 'due', 'become', 'ourselves', 'acros
s', '‘d', 'move', 'alone', 'hundred', 'whither', 'off', 'more', 'any', 'both', 'whereas', 'hers', "n't", '‘s', 'really', 'nowhe
re', 'somewhere', 'while', 'keep', 'somehow', '‘ve', 'else', 'however', 'may', '‘m', 'third', 'anyone', 'many', 'that', 'whereb
y', 'if', 'every', 'these', 'nevertheless', 'again', 'mine', 'as', 'were', 're', 'who', 'i', 'does', 'together', 'whenever', 's
ee', "'m", 'had', 'meanwhile', 'within', 'seemed', 'front', 'whom', 'bottom', 'something', 'further', 'must', 'whence', 'into',
'besides', '‘re', 'toward', 'under', 'do', 'each', 'once', '‘d', 'moreover', 'seem', 'been', 'against', 'did', 'someone', 'als
o', 'twelve', 'whole', 'everyone', 'much', 'n‘t', 'latterly', 'unless', 'yet', 'n’t', 'done', 'ever', 'we', 'whereafter', 'so',
'’s', 'latter', 'least', 'be', 'empty', 'anything', 'made', 'becoming', 'our', 'none', 'will', 'himself', 'such', 'through', 'h
ereby', 'everything', 'she', 'it', 'next', 'among', 'make', 'noone', 'no', 'same', '’re', 'him', 'almost', 'using', 'nor', 'see
ming', 'mostly', 'never', 'he', 'say', 'anywhere', 'because', "'s", 'my', 'became', 'its', 'call', 'seems']
```

```python
# Loading english language... (3 different packages are available --small--medium--large). We are loading small packages.

nlp = spacy.load('en_core_web_sm')
```

```python
# We are passing the input data to spacy

docs = nlp(doc)
print(doc)
```

```
Machine learning (ML) and artificial intelligence (AI) are becoming dominant problem-solving techniques in many areas of resear
ch and industry, not least because of the recent successes of deep learning (DL). However, the equation AI=ML=DL, as recently s
uggested in the news, blogs, and media, falls too short. These fields share the same fundamental hypotheses: computation is a u
seful way to model intelligent behavior in machines. What kind of computation and how to program it? This is not the right ques
tion. Computation neither rules out search, logical, and probabilistic techniques, nor (deep) (un)supervised and reinforcement
learning methods, among others, as computational models do include all of them. They complement each other, and the next breakt
hrough lies not only in pushing each of them but also in combining them.

Big Data is no fad. The world is growing at an exponential rate and so is the size of the data collected across the globe. Data
is becoming more meaningful and contextually relevant, breaking new grounds for machine learning (ML), in particular for deep l
earning (DL) and artificial intelligence (AI), moving them out of research labs into production (Jordan and Mitchell, 2015). Th
e problem has shifted from collecting massive amounts of data to understanding it—turning it into knowledge, conclusions, and a
ctions. Multiple research disciplines, from cognitive sciences to biology, finance, physics, and social sciences, as well as ma
ny companies believe that data-driven and “intelligent” solutions are necessary to solve many of their key problems. High-throu
ghput genomic and proteomic experiments can be used to enable personalized medicine. Large data sets of search queries can be u
sed to improve information retrieval. Historical climate data can be used to understand global warming and to better predict we
ather. Large amounts of sensor readings and hyperspectral images of plants can be used to identify drought conditions and to ga
in insights into when and how stress impacts plant growth and development and in turn how to counterattack the problem of world
hunger. Game data can turn pixels into actions within video games, while observational data can help enable robots to understan
d complex and unstructured environments and to learn manipulation skills.

However, is AI, ML, and DL really synonymous, as recently suggested in the news, blogs, and media? For example, when AlphaGo (S
ilver et al., 2016) defeated South Korean Master Lee Se-dol in the board game Go in 2016, the terms AI, ML, and DL were used by
```

```python
# Word tokenization is performed

tokens = [i.text for i in docs]
print(tokens)
```

```
['Machine', 'learning', '(', 'ML', ')', 'and', 'artificial', 'intelligence', '(', 'AI', ')', 'are', 'becoming', 'dominant', 'pr
oblem', '-', 'solving', 'techniques', 'in', 'many', 'areas', 'of', 'research', 'and', 'industry', ',', 'not', 'least', 'becaus
e', 'of', 'the', 'recent', 'successes', 'of', 'deep', 'learning', '(', 'DL', ')', '.', 'However', ',', 'the', 'equation', 'AI',
'=', 'ML', '=', 'DL', ',', 'as', 'recently', 'suggested', 'in', 'the', 'news', ',', 'blogs', ',', 'and', 'media', ',', 'falls',
'too', 'short', '.', 'These', 'fields', 'share', 'the', 'same', 'fundamental', 'hypotheses', ':', 'computation', 'is', 'a', 'us
eful', 'way', 'to', 'model', 'intelligent', 'behavior', 'in', 'machines', '.', 'What', 'kind', 'of', 'computation', 'and', 'ho
w', 'to', 'program', 'it', '?', 'This', 'is', 'not', 'the', 'right', 'question', '.', 'Computation', 'neither', 'rules', 'out',
'search', ',', 'logical', ',', 'and', 'probabilistic', 'techniques', ',', 'nor', '(', 'deep', ')', '(', 'un)supervised', 'and',
'reinforcement', 'learning', 'methods', ',', 'among', 'others', ',', 'as', 'computational', 'models', 'do', 'include', 'all',
'of', 'them', '.', 'They', 'complement', 'each', 'other', ',', 'and', 'the', 'next', 'breakthrough', 'lies', 'not', 'only', 'i
n', 'pushing', 'each', 'of', 'them', 'but', 'also', 'in', 'combining', 'them', '.', '\n\n', 'Big', 'Data', 'is', 'no', 'fad',
'.', 'The', 'world', 'is', 'growing', 'at', 'an', 'exponential', 'rate', 'and', 'so', 'is', 'the', 'size', 'of', 'the', 'data',
'collected', 'across', 'the', 'globe', '.', 'Data', 'is', 'becoming', 'more', 'meaningful', 'and', 'contextually', 'relevant',
',', 'breaking', 'new', 'grounds', 'for', 'machine', 'learning', '(', 'ML', ')', ',', 'in', 'particular', 'for', 'deep', 'learn
ing', '(', 'DL', ')', 'and', 'artificial', 'intelligence', '(', 'AI', ')', ',', 'moving', 'them', 'out', 'of', 'research', 'lab
s', 'into', 'production', '(', 'Jordan', 'and', 'Mitchell', ',', '2015', ')', '.', 'The', 'problem', 'has', 'shifted', 'from',
'collecting', 'massive', 'amounts', 'of', 'data', 'to', 'understanding', 'it', '—', 'turning', 'it', 'into', 'knowledge', ',',
'conclusions', ',', 'and', 'actions', '.', 'Multiple', 'research', 'disciplines', ',', 'from', 'cognitive', 'sciences', 'to',
'biology', ',', 'finance', ',', 'physics', ',', 'and', 'social', 'sciences', ',', 'as', 'well', 'as', 'many', 'companies', 'bel
ieve', 'that', 'data', '-', 'driven', 'and', '“', 'intelligent', '”', 'solutions', 'are', 'necessary', 'to', 'solve', 'many',
'of', 'their', 'key', 'problems', '.', 'High', '-', 'throughput', 'genomic', 'and', 'proteomic', 'experiments', 'can', 'be', 'u
sed', 'to', 'enable', 'personalized', 'medicine', '.', 'Large', 'data', 'sets', 'of', 'search', 'queries', 'can', 'be', 'used',
'to', 'improve', 'information', 'retrieval', '.', 'Historical', 'climate', 'data', 'can', 'be', 'used', 'to', 'understand', 'gl
obal', 'warming', 'and', 'to', 'better', 'predict', 'weather', '.', 'Large', 'amounts', 'of', 'sensor', 'readings', 'and', 'hyp
erspectral', 'images', 'of', 'plants', 'can', 'be', 'used', 'to', 'identify', 'drought', 'conditions', 'and', 'to', 'gain', 'in
sights', 'into', 'when', 'and', 'how', 'stress', 'impacts', 'plant', 'growth', 'and', 'development', 'and', 'in', 'turn', 'ho
w', 'to', 'counterattack', 'the', 'problem', 'of', 'world', 'hunger', '.', 'Game', 'data', 'can', 'turn', 'pixels', 'into', 'ac
tions', 'within', 'video', 'games', ',', 'while', 'observational', 'data', 'can', 'help', 'enable', 'robots', 'to', 'understan
```

```python
word_frequencies = {}

for word in docs:
    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)
```

```
{'Machine': 1, 'learning': 6, 'ML': 5, 'artificial': 2, 'intelligence': 2, 'AI': 6, 'dominant': 1, 'problem': 3, 'solving': 1,
'techniques': 2, 'areas': 1, 'research': 3, 'industry': 1, 'recent': 1, 'successes': 1, 'deep': 3, 'DL': 5, 'equation': 1, 'rec
ently': 2, 'suggested': 2, 'news': 2, 'blogs': 2, 'media': 3, 'falls': 1, 'short': 1, 'fields': 1, 'share': 1, 'fundamental':
1, 'hypotheses': 1, 'computation': 2, 'useful': 1, 'way': 1, 'model': 1, 'intelligent': 2, 'behavior': 1, 'machines': 1, 'kin
d': 1, 'program': 1, 'right': 1, 'question': 1, 'Computation': 1, 'rules': 1, 'search': 2, 'logical': 1, 'probabilistic': 1, 'u
n)supervised': 1, 'reinforcement': 1, 'methods': 1, 'computational': 1, 'models': 1, 'include': 1, 'complement': 1, 'breakthrou
gh': 1, 'lies': 1, 'pushing': 1, 'combining': 1, '\n\n': 2, 'Big': 1, 'Data': 2, 'fad': 1, 'world': 2, 'growing': 1, 'exponenti
al': 1, 'rate': 1, 'size': 1, 'data': 7, 'collected': 1, 'globe': 1, 'meaningful': 1, 'contextually': 1, 'relevant': 1, 'breaki
ng': 1, 'new': 1, 'grounds': 1, 'machine': 2, 'particular': 1, 'moving': 1, 'labs': 1, 'production': 1, 'Jordan': 1, 'Mitchel
l': 1, '2015': 1, 'shifted': 1, 'collecting': 1, 'massive': 1, 'amounts': 2, 'understanding': 1, '—': 1, 'turning': 1, 'knowled
ge': 1, 'conclusions': 1, 'actions': 2, 'Multiple': 1, 'disciplines': 1, 'cognitive': 1, 'sciences': 2, 'biology': 1, 'financ
e': 1, 'physics': 1, 'social': 1, 'companies': 1, 'believe': 1, 'driven': 1, '“': 2, '”': 1, 'solutions': 2, 'necessary': 1, 's
olve': 1, 'key': 1, 'problems': 1, 'High': 1, 'throughput': 1, 'genomic': 1, 'proteomic': 1, 'experiments': 1, 'enable': 2, 'pe
rsonalized': 1, 'medicine': 1, 'Large': 2, 'sets': 1, 'queries': 1, 'improve': 1, 'information': 1, 'retrieval': 1, 'Historica
l': 1, 'climate': 1, 'understand': 2, 'global': 1, 'warming': 1, 'better': 1, 'predict': 1, 'weather': 1, 'sensor': 1, 'reading
s': 1, 'hyperspectral': 1, 'images': 1, 'plants': 1, 'identify': 1, 'drought': 1, 'conditions': 1, 'gain': 1, 'insights': 1, 's
tress': 1, 'impacts': 1, 'plant': 1, 'growth': 1, 'development': 1, 'turn': 2, 'counterattack': 1, 'hunger': 1, 'Game': 1, 'pix
els': 1, 'video': 1, 'games': 1, 'observational': 1, 'help': 1, 'robots': 1, 'complex': 1, 'unstructured': 1, 'environments':
1, 'learn': 1, 'manipulation': 1, 'skills': 1, 'synonymous': 1, 'example': 1, 'AlphaGo': 2, 'Silver': 1, 'et': 1, 'al': 1, '201
6': 2, 'defeated': 1, 'South': 1, 'Korean': 1, 'Master': 1, 'Lee': 1, 'Se': 1, 'dol': 1, 'board': 1, 'game': 1, 'terms': 1, 'de
```

```
# Taking max frequency for normalization

maxFrequency = max(word_frequencies.values())
maxFrequency

7

# Normalizing the data

for i in word_frequencies.keys():
    word_frequencies[i] = word_frequencies[i]/maxFrequency

print(word_frequencies)

{'Machine': 0.14285714285714285, 'learning': 0.8571428571428571, 'ML': 0.7142857142857143, 'artificial': 0.2857142857142857, 'i
ntelligence': 0.2857142857142857, 'AI': 0.8571428571428571, 'dominant': 0.14285714285714285, 'problem': 0.42857142857142855, 's
olving': 0.14285714285714285, 'techniques': 0.2857142857142857, 'areas': 0.14285714285714285, 'research': 0.42857142857142855,
'industry': 0.14285714285714285, 'recent': 0.14285714285714285, 'successes': 0.14285714285714285, 'deep': 0.42857142857142855,
'DL': 0.7142857142857143, 'equation': 0.14285714285714285, 'recently': 0.2857142857142857, 'suggested': 0.2857142857142857, 'ne
ws': 0.2857142857142857, 'blogs': 0.2857142857142857, 'media': 0.42857142857142855, 'falls': 0.14285714285714285, 'short': 0.14
285714285714285, 'fields': 0.14285714285714285, 'share': 0.14285714285714285, 'fundamental': 0.14285714285714285, 'hypotheses':
0.14285714285714285, 'computation': 0.2857142857142857, 'useful': 0.14285714285714285, 'way': 0.14285714285714285, 'model': 0.1
4285714285714285, 'intelligent': 0.2857142857142857, 'behavior': 0.14285714285714285, 'machines': 0.14285714285714285, 'kind':
0.14285714285714285, 'program': 0.14285714285714285, 'right': 0.14285714285714285, 'question': 0.14285714285714285, 'Computatio
n': 0.14285714285714285, 'rules': 0.14285714285714285, 'search': 0.2857142857142857, 'logical': 0.14285714285714285, 'probabili
stic': 0.14285714285714285, 'un)supervised': 0.14285714285714285, 'reinforcement': 0.14285714285714285, 'methods': 0.1428571428
5714285, 'computational': 0.14285714285714285, 'models': 0.14285714285714285, 'include': 0.14285714285714285, 'complement': 0.1
4285714285714285, 'breakthrough': 0.14285714285714285, 'lies': 0.14285714285714285, 'pushing': 0.14285714285714285, 'combinin
g': 0.14285714285714285, '\n\n': 0.2857142857142857, 'Big': 0.14285714285714285, 'Data': 0.2857142857142857, 'fad': 0.142857142
85714285, 'world': 0.2857142857142857, 'growing': 0.14285714285714285, 'exponential': 0.14285714285714285, 'rate': 0.1428571428
5714285, 'size': 0.14285714285714285, 'data': 1.0, 'collected': 0.14285714285714285, 'globe': 0.14285714285714285, 'meaningfu
```

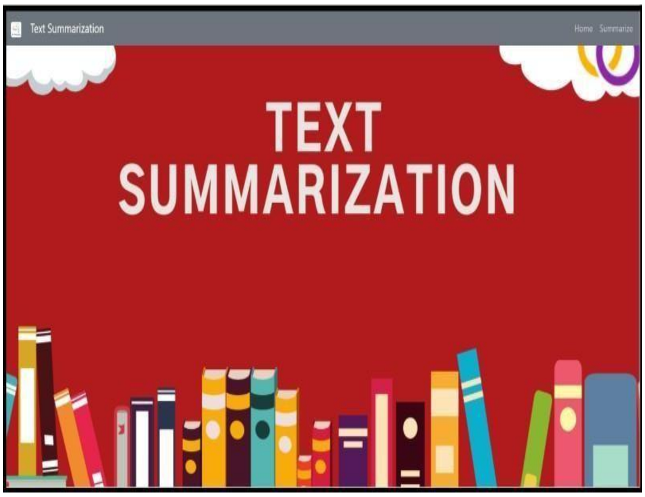
```
sent_tokenz = [sent for sent in docs.sents]
print(sent_tokenz)

[Machine learning (ML) and artificial intelligence (AI) are becoming dominant problem-solving techniques in many areas of resea
rch and industry, not least because of the recent successes of deep learning (DL)., However, the equation AI=ML=DL, as recently
suggested in the news, blogs, and media, falls too short., These fields share the same fundamental hypotheses: computation is a
useful way to model intelligent behavior in machines., What kind of computation and how to program it?, This is not the right q
uestion., Computation neither rules out search, logical, and probabilistic techniques, nor (deep) (un)supervised and reinforcem
ent learning methods, among others, as computational models do include all of them., They complement each other, and the next b
reakthrough lies not only in pushing each of them but also in combining them.

, Big Data is no fad., The world is growing at an exponential rate and so is the size of the data collected across the globe.,
Data is becoming more meaningful and contextually relevant, breaking new grounds for machine learning (ML), in particular for d
eep learning (DL) and artificial intelligence (AI), moving them out of research labs into production (Jordan and Mitchell, 201
5)., The problem has shifted from collecting massive amounts of data to understanding it—turning it into knowledge, conclusion
s, and actions., Multiple research disciplines, from cognitive sciences to biology, finance, physics, and social sciences, as w
ell as many companies believe that data-driven and “intelligent” solutions are necessary to solve many of their key problems.,
High-throughput genomic and proteomic experiments can be used to enable personalized medicine., Large data sets of search queri
es can be used to improve information retrieval., Historical climate data can be used to understand global warming and to bette
r predict weather., Large amounts of sensor readings and hyperspectral images of plants can be used to identify drought conditi
ons and to gain insights into when and how stress impacts plant growth and development and in turn how to counterattack the pro
blem of world hunger., Game data can turn pixels into actions within video games, while observational data can help enable robo
ts to understand complex and unstructured environments and to learn manipulation skills.

, However, is AI, ML, and DL really synonymous, as recently suggested in the news, blogs, and media?, For example, when AlphaGo
(Silver et al., 2016) defeated South Korean Master Lee Se-dol in the board game Go in 2016, the terms AI, ML, and DL were used
by the media to describe how AlphaGo won., In addition to this, even Gartner's list (Panetta, 2017) of top 10 Strategic Trends
for 2018 places (, narrow) AI at the very top, specifying it as “consisting of highly scoped machine-learning solutions that ta
rget a specific task.]
```

```
sentence_score = {}

for sent in sent_tokenz:
    for word in sent:
        if word.text.lower() in word_frequencies.keys():
            if sent not in sentence_score.keys():
                sentence_score[sent] = word_frequencies[word.text.lower()]
            else:
                sentence_score[sent] += word_frequencies[word.text.lower()]


sentence_score

{Machine learning (ML) and artificial intelligence (AI) are becoming dominant problem-solving techniques in many areas of resea
rch and industry, not least because of the recent successes of deep learning (DL).: 4.999999999999998,
 However, the equation AI=ML=DL, as recently suggested in the news, blogs, and media, falls too short.: 1.9999999999999998,
 These fields share the same fundamental hypotheses: computation is a useful way to model intelligent behavior in machines.: 1.
8571428571428568,
 What kind of computation and how to program it?: 0.5714285714285714,
 This is not the right question.: 0.2857142857142857,
 Computation neither rules out search, logical, and probabilistic techniques, nor (deep) (un)supervised and reinforcement learn
ing methods, among others, as computational models do include all of them.: 3.428571428571428,
 They complement each other, and the next breakthrough lies not only in pushing each of them but also in combining them.
 : 0.9999999999999999,
 Big Data is no fad.: 1.1428571428571428,
 The world is growing at an exponential rate and so is the size of the data collected across the globe.: 2.142857142857143,
 Data is becoming more meaningful and contextually relevant, breaking new grounds for machine learning (ML), in particular for
deep learning (DL) and artificial intelligence (AI), moving them out of research labs into production (Jordan and Mitchell, 201
5).: 6.000000000000001,
 The problem has shifted from collecting massive amounts of data to understanding it—turning it into knowledge, conclusions, an
d actions.: 3.1428571428571423,
```

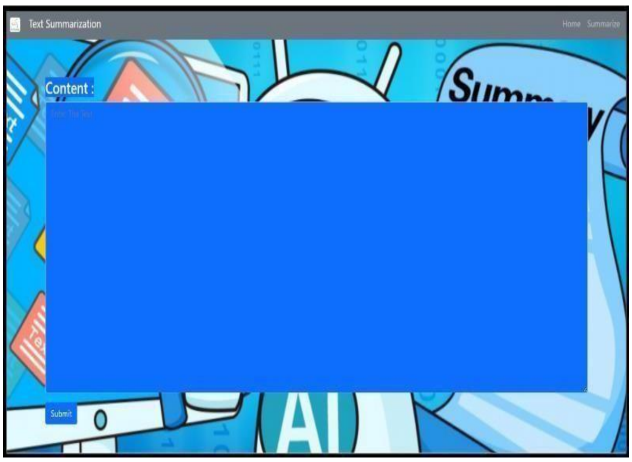
```
from heapq import nlargest

select_len = int(len(sent_tokenz)*0.3)
select_len

6

summary = nlargest(select_len,sentence_score,sentence_score.get)
summary

[Data is becoming more meaningful and contextually relevant, breaking new grounds for machine learning (ML), in particular for
deep learning (DL) and artificial intelligence (AI), moving them out of research labs into production (Jordan and Mitchell, 201
5).,
 Game data can turn pixels into actions within video games, while observational data can help enable robots to understand compl
ex and unstructured environments and to learn manipulation skills.
 ,
 Machine learning (ML) and artificial intelligence (AI) are becoming dominant problem-solving techniques in many areas of resea
rch and industry, not least because of the recent successes of deep learning (DL).,
 Multiple research disciplines, from cognitive sciences to biology, finance, physics, and social sciences, as well as many comp
anies believe that data-driven and "intelligent" solutions are necessary to solve many of their key problems.,
 Large amounts of sensor readings and hyperspectral images of plants can be used to identify drought conditions and to gain ins
ights into when and how stress impacts plant growth and development and in turn how to counterattack the problem of world hunge
r.,
 Computation neither rules out search, logical, and probabilistic techniques, nor (deep) (un)supervised and reinforcement learn
ing methods, among others, as computational models do include all of them.]
```

```
summary = [word.text for word in summary]
summary

['Data is becoming more meaningful and contextually relevant, breaking new grounds for machine learning (ML), in particular for
deep learning (DL) and artificial intelligence (AI), moving them out of research labs into production (Jordan and Mitchell, 201
5).',
 'Game data can turn pixels into actions within video games, while observational data can help enable robots to understand comp
lex and unstructured environments and to learn manipulation skills.\n\n',
 'Machine learning (ML) and artificial intelligence (AI) are becoming dominant problem-solving techniques in many areas of rese
arch and industry, not least because of the recent successes of deep learning (DL).',
 'Multiple research disciplines, from cognitive sciences to biology, finance, physics, and social sciences, as well as many com
panies believe that data-driven and "intelligent" solutions are necessary to solve many of their key problems.',
 'Large amounts of sensor readings and hyperspectral images of plants can be used to identify drought conditions and to gain in
sights into when and how stress impacts plant growth and development and in turn how to counterattack the problem of world hung
er.',
 'Computation neither rules out search, logical, and probabilistic techniques, nor (deep) (un)supervised and reinforcement lear
ning methods, among others, as computational models do include all of them.']

summary = " ".join(summary)
summary

'Data is becoming more meaningful and contextually relevant, breaking new grounds for machine learning (ML), in particular for
deep learning (DL) and artificial intelligence (AI), moving them out of research labs into production (Jordan and Mitchell, 201
5). Game data can turn pixels into actions within video games, while observational data can help enable robots to understand co
mplex and unstructured environments and to learn manipulation skills.\n\n Machine learning (ML) and artificial intelligence (A
I) are becoming dominant problem-solving techniques in many areas of research and industry, not least because of the recent suc
cesses of deep learning (DL). Multiple research disciplines, from cognitive sciences to biology, finance, physics, and social s
ciences, as well as many companies believe that data-driven and "intelligent" solutions are necessary to solve many of their ke
y problems. Large amounts of sensor readings and hyperspectral images of plants can be used to identify drought conditions and
to gain insights into when and how stress impacts plant growth and development and in turn how to counterattack the problem of
world hunger. Computation neither rules out search, logical, and probabilistic techniques, nor (deep) (un)supervised and reinfo
rcement learning methods, among others, as computational models do include all of them.'
```

Building html pages



Home.html



Summarize.html

Build python code

```python
from flask import Flask, render_template, request
import spacy
from spacy.lang.en.stop_words import STOP_WORDS
from heapq import nlargest
# from nltk. import punctuation
```

```python
app = Flask(__name__)
```

```python
@app.route("/")
def about():
    return render_template('home.html')

@app.route("/home")
def home():
    return render_template('home.html')

@app.route("/summarize")
def home1():
    return render_template('summarize.html')

@app.route("/submit")
def home2():
    return render_template('submit.html')
```

```python
@app.route("/summary", methods=['POST'])
def summary():
    stopWords = list(STOP_WORDS)
    nlp = spacy.load('en_core_web_sm')
    doc = request.form['text']
    print(doc)
    docs = nlp(doc)
    tokens = [i.text for i in docs]
    punctuation = '!"#$%&\()*+,-./:;<=>?@[\\]^_`{|}~\n'
    print(punctuation)
    word_frequencies = {}
    for word in docs:
        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
    maxFrequency = max(word_frequencies.values())
    for i in word_frequencies.keys():
        word_frequencies[i] = word_frequencies[i] / maxFrequency
    sent_tokenz = [sent for sent in docs.sents]
    sentence_score = {}
    for sent in sent_tokenz:
        for word in sent:
            if word.text.lower() in word_frequencies.keys():
                if sent not in sentence_score.keys():
                    sentence_score[sent] = word_frequencies[word.text.lower()]
                else:
                    sentence_score[sent] += word_frequencies[word.text.lower()]
    select_len = int(len(sent_tokenz) * 0.3)
    summary = nlargest(select_len, sentence_score, sentence_score.get)
    summary = [word.text for word in summary]
    summary = " ".join(summary)
    return render_template('submit.html', predictionText=summary)
```

```python
if __name__ == "__main__":
    app.run(debug=True)
```

```
WARNING:tensorflow:From C:\Users\DELL\Anaconda3\lib\site-packa
support.<locals>.wrapper (from tensorflow.python.ops.array_ops
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.w
 * Debugger is active!
 * Debugger PIN: 923-051-529
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```