AI-Based Intelligent Insight Extractor Using IBM Watson

An Industrial/Practical Training Report

Submitted to the Faculty of Engineering of

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA, KAKINADA

In partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

P. Chandrika (19481A05I6)
MD. Shaheed (19481A05F9)
M. Leela Rani (19481A05E4)

Under the Enviable and Esteemed Guidance of

Mrs. G. Bharathi, M.Tech, (Ph.D)

Sr. Gr. Assistant Professor, Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)

SESHADRI RAO KNOWLEDGE VILLAGE

GUDLAVALLERU – 521336
ANDHRA PRADESH

2021 - 2022

SMARTBRIDGE
Let's Bridge the Gap

INTERNSHIP REPORT APPROVAL FORM

With immense pleasure, this is to approved that the students of Seshadri Rao Gudlavalleru Engineering College i.e.

P. Chandrika (19481A05I6)
MD. Shaheed (19481A05F9)
M. Leela Rani (19481A05E4)

Successfully completed their Project and Project Report on "**AI-Based intelligent insight extractor using IBM Watson Studio**" under our guidance.

We are highly impressed with the work that they have done and commend them on their quick grasping skills. They have shown good intent to learn and have put the knowledge gained into application in the form of this project. We appreciate the hard work and commitment shown by them.

We, hereby approve that this document is completely checked and accepted by SmartBridge Technical Team. Its been an absolute pleasure to educate and mentor these students. We hope that this document will also serve as a Letter of Recommendation, to whomsoever applied.

We wish them success in all future endeavours and a great career ahead.

Jaya Prakash. Ch
Program Manager

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)

SESHADRI RAO KNOWLEDGE VILLAGE, GUDLAVALLERU

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled "AI-Based Intelligent Insight Extractor Using IBM Watson" is a bonafide record of work carried out by P. Chandrika(19481A05I6), MD. Shaheed(19481A05F9), M. Leela Rani(19481A05E4) under the guidance and supervision of *Mrs. G. Bharathi* in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of Jawaharlal Nehru Technological University, Kakinada during the academic year 2022-2023.

Project Guide
***(Mrs. G. Bharathi*)**

Head of the Department
(Dr. M. BABU RAO)

External Examiner

ACKNOWLEDGMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose constant guidance and encouragements crown all the efforts with success.

We would like to express our deep sense of gratitude and sincere thanks to ***Mrs. G. Bharathi*** Sr. Gr. Assistant Professor, Department of Computer Science and Engineering for her constant guidance, supervision and motivation in completing the project work.

We feel elated to express our floral gratitude and sincere thanks to **Dr. M. Babu Rao**, Head of the Department, Computer Science and Engineering for his encouragements all the way during analysis of the project. His annotations, insinuations and criticisms are the key behind the successful completion of the project work.

We would like to take this opportunity to thank our beloved principal **Dr. G. V. S. N. R. V. Prasad** for providing a great support for us in completing our project and giving us the opportunity for doing project.

Our Special thanks to the faculty of our department and programmers of our computer lab. Finally, we thank our family members, non-teaching staff, attendants and our friends, who had directly or indirectly helped and supported us in completing our project in time.

Team members

P. Chandrika (19481A05I6)
MD. Shaheed (18481A05F9)
M. Leela Rani (18481A05E4)

ABSTRACT

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 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 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 which includes 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 three paragraphs are summarized into short story less than a paragraph using NLP.

INDEX

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

Manually generating a summary can be time-consuming and tedious. Automatic text summarization promises to overcome such difficulties and allow you to generate the key ideas in a pieceof 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 with high frequencies and take the most important sentences to convey the message.

1.2 PRE REQUISITIES:

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

- Anaconda navigator and pycharm.

Anaconda Navigator is a free and open-source distribution of the Python and 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.

- Python packages**.**
    - Open anaconda prompt
    - Type **"pip install numpy"** and click enter.
    - Type **"pip install Flask"** and click enter**.**
    - Type "**pip install spacy**" and click enter.

1.3 OBJECTIVES OF THE PROJECT

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.

1.4 PROBLEM STATEMENT

Have you ever summarized a lengthy document into a short paragraph? How long did you take? Manually generating a summary can be time-consuming and tedious. Automatic text summarization promises to overcome such difficulties and allow you to generate the key ideas in a piece of writing easily.

CHAPTER 2
PROPOSED METHOD

2.1 PROJECT FLOW

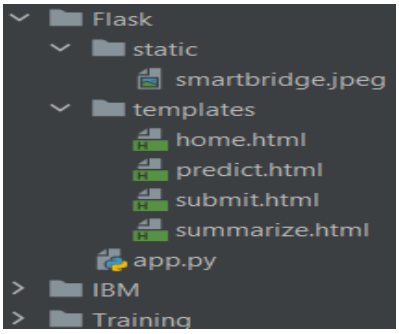
- The user interacts with the UI to enter the input.
- Entered input is analyzed by the model which is integrated.
- Once the model analyses the input the summary is showcased on the UI

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

- Import required libraries
    - Read dataset
- Text pre-processing
    - Stopwords
    - Load spacy language pipeline
    - Word tokenization
    - Word frequency
    - Normalization
    - Sentence tokenization
    - Sentence score
    - summary
- Application building
    - Create an html file
    - Build python code

2.2 PROJECT STRUCTURE:

Create the project folder which contains files as shown below.



- We are building a flask application that needs HTML pages stored in the templates folder and a python script app.py for scripting.
- in the training folder, the model training file (.ipynb) is available and in the IBM folder, the IBM model training file (.ipynb) is available.

2.3 IM**PLEMENTATION:**

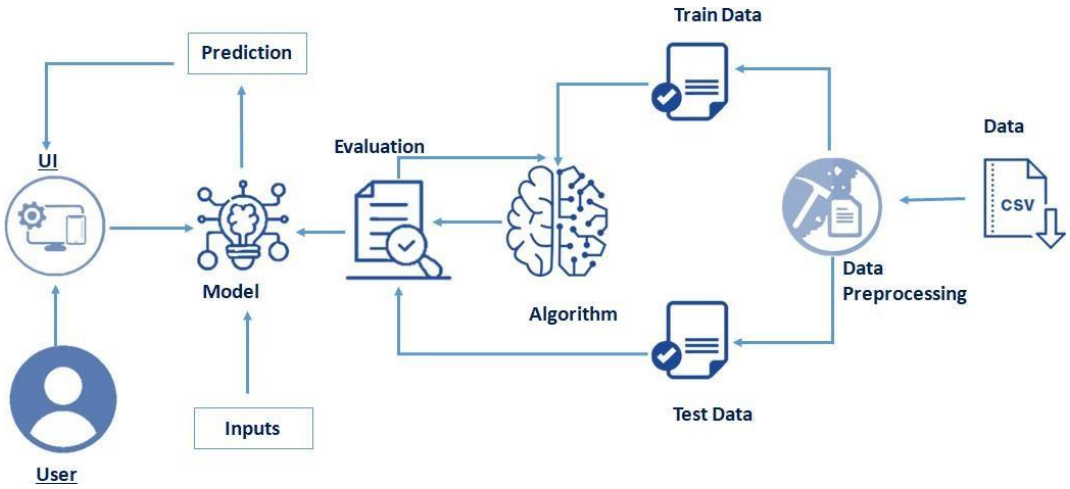


Fig: Demonstrating the implementation of the project through project flow diagram

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

1. **IMPORT REQUIRED LIBRARIES:**
    - Read Dataset
2. **TEXT PREPROCESSING:**
    - StopWords.
    - Load spacy language pipeline.
    - Word Tokenization.
    - Word Frequency.
    - Normalization.
    - Sentence Tokenization.
    - Sentence Score.
    - Summary.
3. **APPLICATION BUILDING:**
    - Create an HTML file
    - Build python code

2.3.1 IMPORT REQUIRED LIBRARIES:

Import the necessary libraries as shown in the figure.

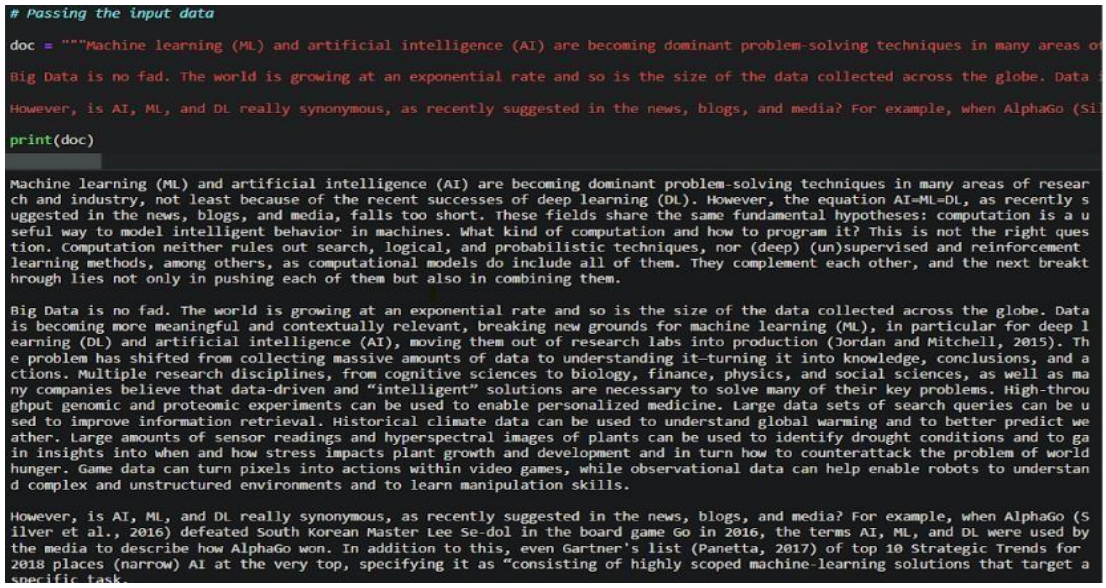
- Spacy is an open-source software library for advanced natural language processing.
- Package string implements simple functions to manipulate UTF-8 encoded strings.

```
# Importing the required packages

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

2.3.1.1 READ DATASET:

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.



```
# 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.
```

2.3.2 TEXT PREPROCESSING:

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

- StopWords.
- Load spacy language pipeline.
- Word Tokenization.
- Word Frequency.

- Normalization.
- Sentence Tokenization.
- Sentence Score.
- Summary

In natural language processing, text pre-processing is the practice of cleaning and preparing the data

2.3.2.1 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 only on important text data.

The list of stopwords in the English language is shown below.

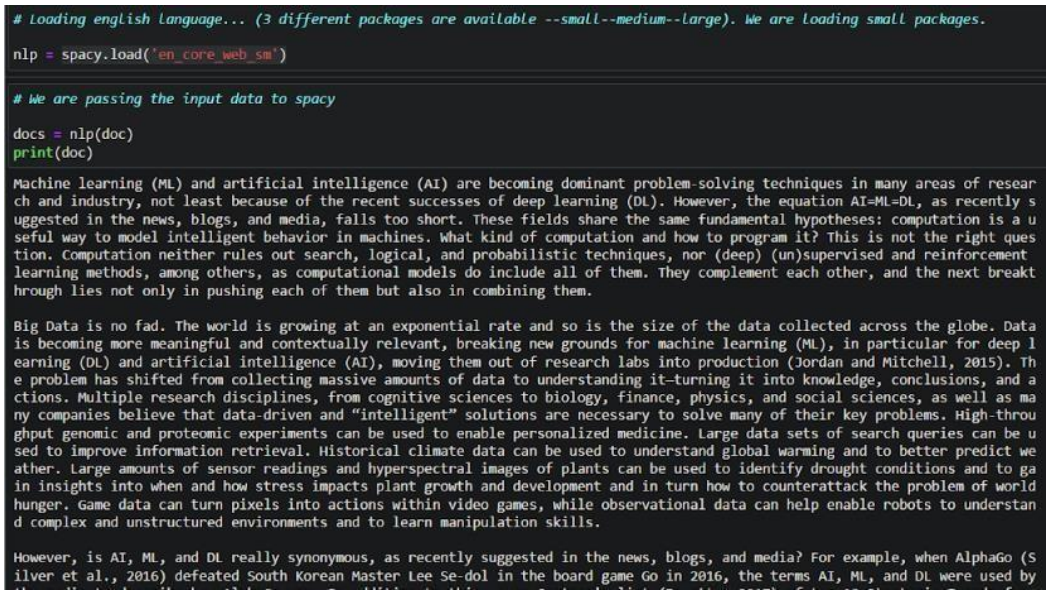
```
# To visualize the stop words

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']
```

2.3.2.2 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 are four 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.
- Now the input is passed to the spacy pipelines. (refer below image)



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

nlp = spacy.load('en_core_web_sm')

# 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
```

2.3.2.3 WORD TOKENIZATION:

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

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

```
# 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
```

2.3.2.4 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 the dictionary and the counts of the words are passed as values with the help of for loop and conditions.

```
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
```

2.3.2.5 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.
- For the output refer to the below image.

```
# 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': 0.4285714285714855, 's
olving': 0.14285714285714285, 'techniques': 0.2857142857142857, 'areas': 0.14285714285714285, 'research': 0.4285714285714855,
'industry': 0.14285714285714285, 'recent': 0.14285714285714285, 'successes': 0.14285714285714285, 'deep': 0.4285714285714855,
'DL': 0.7142857142857143, 'equation': 0.14285714285714285, 'recently': 0.2857142857142857, 'suggested': 0.2857142857142857, 'ne
ws': 0.2857142857142857, 'blogs': 0.2857142857142857, 'media': 0.4285714285714855, '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
```

2.3.2.6 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 done by list comprehension

```
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.]
```

2.3.2.7 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

```
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,
```

2.3.2.8 SUMMARY:

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

```
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.]
```

- To join the sentence list comprehension and join() method is used.

```
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.'
```

2.3.3 APPLICATION BUILDING:

In this section, we will be building a web application that is integrated into the model we built. 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

2.3.3.1 BUILDING HTML PAGES:

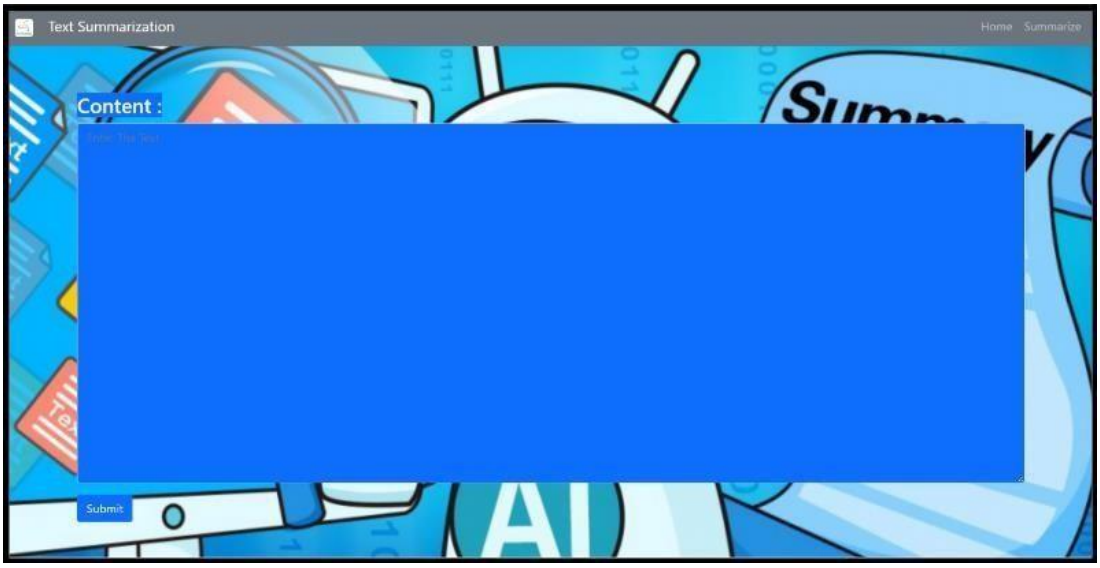
For this project create 3 HTML files namely

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

And save them in template folder.

Let's see how our home.html page looks like:

Let's see how our summarize.html page looks like:



2.3.3.2 BUILD PYTHON CODE:

Import the libraries

```
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
```

Importing the flask module into the project is mandatory. An object of the Flask class is our WSGI application.

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

Flask constructor takes the name of the current module(--name--) as an argument.

Render HTML page:

Here we will be using the declared constructor to route to the HTML page that we have created earlier.

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.

```
@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')
```

Retrieves the summary from UI:

Copy and paste the code of the model building.

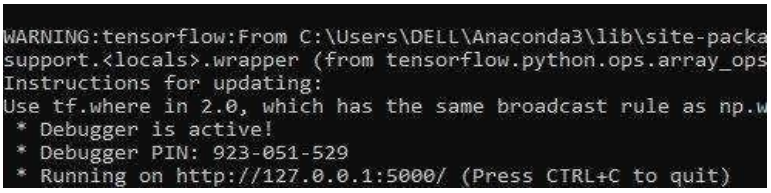
```python
@app.route("/summary", methods=['POST'])
def summary():
    stopWords = list(STOP_WORDS)
    nlp = spacy.load('en_core_web_sm')
    doc = request.form['text']
    print(doc)
    docs = nlp(doc)
    tokens = [i.text for i in docs]
    punctuation = '!"#$%&\()*+,-./:;<=>?@[\\]^_`{|}~\n'
    print(punctuation)
    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)
```

Main function.

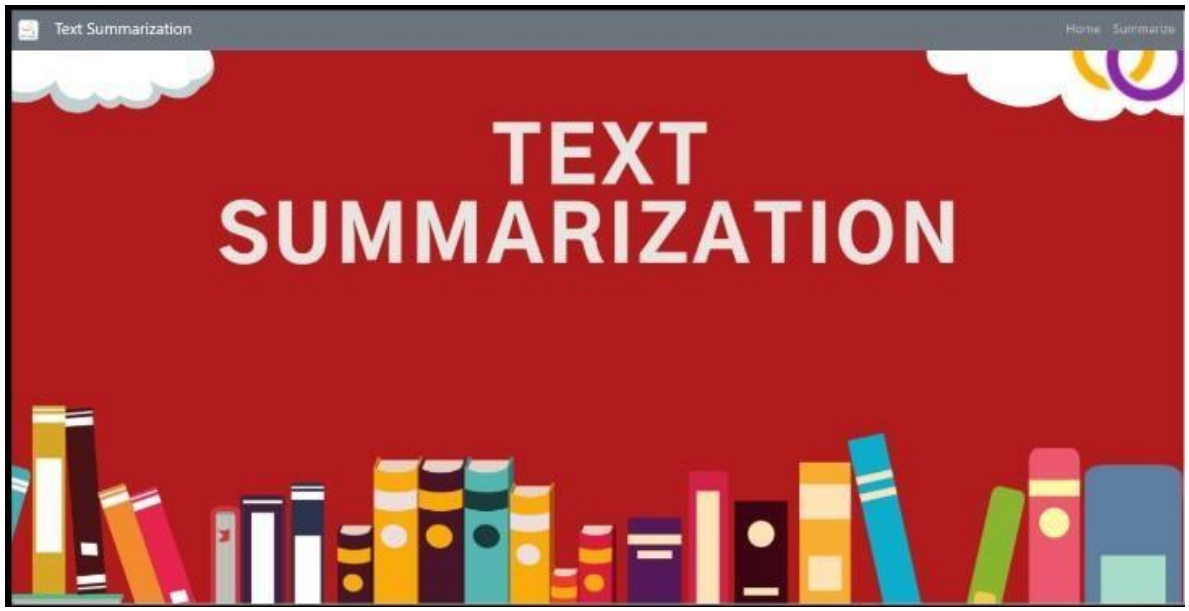
```python
if __name__ == "__main__":
    app.run(debug=True)
```

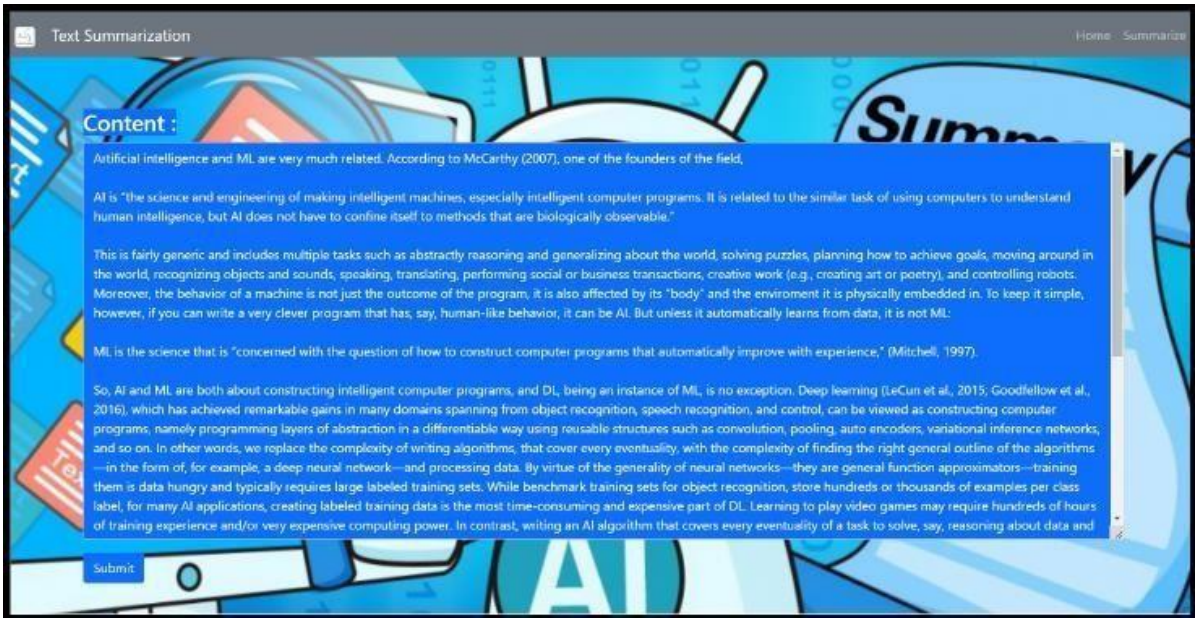
2.3.3.3 RUN THE APPLICATION:

- 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 the submit button, and see the result/prediction on the web.


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)

CHAPTER 3
RESULTS AND DISCUSSION






Text Summarization
Home
Summarize
Summary : In contrast, writing an AI algorithm that covers every eventuality of a task to solve, say, reasoning about data and knowledge to label data automatically (Ratner et al., 2016; Roth, 2017) and, in turn, make, for example, DL less data-hungry–is a lot of manual work, but we know what the algorithm does by design and that it can study and that it can more easily understand the complexity of the problem it solves. Deep learning (LeCun et al., 2015; Goodfellow et al., 2016), which has achieved remarkable gains in many domains spanning from object recognition, speech recognition, and control, can be viewed as constructing computer programs, namely programming layers of abstraction in a differentiable way using reusable structures such as convolution, pooling, auto encoders, variational inference networks, and so on. In other words, we replace the complexity of writing algorithms, that cover every eventuality, with the complexity of finding the right general outline of the algorithms—in the form of, for example, a deep neural network—and processing data. By virtue of the generality of neural networks—they are general function approximators—training them is data hungry and typically requires large labeled training sets. Machine learning is specifically about learning—learning from examples, from definitions, from being told, and from behavior. While benchmark training sets for object recognition, store hundreds or thousands of examples per class label, for many AI applications, creating labeled training data is the most time-consuming and expensive part of DL. This is fairly generic and includes multiple tasks such as abstractly reasoning and generalizing about the world, solving puzzles, planning how to achieve goals, moving around in the world, recognizing objects and sounds, speaking, translating, performing social or business transactions, creative work (e.g., creating art or poetry), and controlling robots. In other words, AI has to cover knowledge, reasoning, and learning, using programmed and learning-based programmed models in a combined fashion. ML is the science that is "concerned with the question of how to construct computer programs that automatically improve with experience." (Mitchell, 1997).
Text
AI

CHAPTER 4
CONCLUSION

In this project, Text summarization is defined as the process of refining the most useful information from the source document to provide an abridged version for the specific task. In this paper, an automatic text summarization approach is proposed which uses NLP with python concept. This summarizer works on a variety of long stories with more than three paragraphs that are very long to read and time consuming. The proposed method results in better precision and saves time than the existing system Various NLP based algorithms such as TF-IDF can be used to achieve this objective. Thus we would first produce a long story as a input text provided and then extractive methods are employed to shorten those sentences produced. This will ensure that the summary produced is to the highest condensed form which can be made in the mobile industry.

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.

REFERENCES

[01] **Wenbin Wei** and **Mark Hansen** used a logarithmic function for estimating the demand 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

[02] 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

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

[04] The study by **Ya-Ling Huang** and **Chin - Tsai Lin** who used The **Gray Envelope Prediction Model (GEPM**) for monthly, seasonal and annual passenger traffic forecasts (Ya-Ling Huang , Chin - Tsai Lin, 2011).
https://www.researchgate.net/publication/335877614_Forecasting_Passenger_Traffic_for_a_Regional_Airport

[05] A **GreyMarkov** model was applied by **Zhang Wei, Zhu Jinfu** in 2009 in order to estimate passenger traffic by integrating the Grey model.
https://www.researchgate.net/publication/335877614_Forecasting_Passenger_Traffic_for_a_Regional_Airport

[06] Another study developed by **Wai Hong Kan Tsui** et al for predicting passenger traffic at Hong Kong airport using **SARIMA** and **ARIMAX** models has resulted in high accuracy, with very small prediction errors .
https://www.researchgate.net/publication/259121979_Forecasting_of_Hong_Kong_airport's_passenger_throughput

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
Seshadri Rao Knowledge Village, Gudlavalleru

Department of Computer Science and Engineering

Program Outcomes (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions., component, or software to meet the desired needs.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1 : Design, develop, test and maintain reliable software systems and intelligent systems.

PSO2 : Design and develop web sites, web apps and mobile apps.

PROJECT PROFORMA

Classification of Project	Application	Product	Research	Review
	√			

Note: Tick Appropriate category

Project Outcomes	
Course Outcome (CO1)	Identify and analyze the problem statement using prior technical knowledge in the domain of interest.
Course Outcome (CO2)	Design and develop engineering solutions to complex problems by employing systematic approach.
Course Outcome (CO3)	Examine ethical, environmental, legal and security issues during project implementation.
Course Outcome (CO4)	Prepare and present technical reports by utilizing different visualization tools and evaluation metrics.

Mapping Table

CS1536 : INDUSTRIAL/PRACTICAL TRAINING															
Course outcomes	**Program Outcomes and Program Specific Outcome**														
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**		**PSO1**	**PSO2**
CO1: Acquire technical competence in the specific domain during the training.	3	2	2	2	2			2	2	2	1	2		2	
CO2: Identify the problem statement based on the requirements of the industry	3	3	2	2	1			2	2	2	1	2		2	2
CO3: Adapt project management skills on par with industrial standards.	3	2	2	1	3	1	1	2	2	2	3	2		2	
CO4: Develop a system model to obtain a solution and generate a report.	3	2	3	3	3	2	1	2	2	2	3	2		2	2
INDUSTRIAL/PRACTICAL TRAINING	3	3	3	2	3	1	1	2	2	2	2	2		2	1

Note: Map each project outcomes with POs and PSOs with either 1 or 2 or 3 based on level of mapping as follows:

1-Slightly (Low) mapped 2-Moderately (Medium) mapped 3-Substantially (High) mapped