# Project Report
# On
# TEXT SUMMARIZATION

A Disseration submitted to the
RGUKT-AP in partial fulfillment of the
degree of
Bachelor of Technology
in
Computer Science

By

Chakka Meghana (R170035)
Purini Jnana Prasanna (R170103)

Under the guidance of :

**Ms.Shabana**
**Assistant professor**
Computer Science Department

Rajiv Gandhi University of Knowledge
Technologies AP-IIIT , Rk Valley , Idupulapaya ,
Kadapa – 516330 , Andhra Pradesh , India

# CERTIFICATE

This is to certify that the project work titled **"TEXT SUMMARIZATION"** submitted by Chakka Meghana and Purini Jnana Prasanna, bearing id R170035 and R170103, in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science is a bonafide work carried out by them under my supervision and guidance.

The disseration has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

**Mr.P.Harinadha,**
Head of the Department,
Computer Science Department,
RGUKT, RK VALLEY.

**Ms.Shaik Shabana,**
Project Internal Guide,
Computer Science Dept,
RGUKT, RK VALLEY.

# DECLARATION

We **Chakka Meghana** and **Purini Jnana Prasanna** hereby declare that this project work titled **“TEXT SUMMARIZATION”** submitted by us under the guidance and supervision of **Ms. Shabana** is a bonafide work. We also declare that it has not been submitted previously in part or in full to this University or other University or Institution for the award of any degree or diploma.

Date :

Chakka Meghana (R170035)
Purini Jnana Prasanna (R170103)

Place : RGUKT , RK VALLEY

# ACKNOWLEDGEMENT

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

We are extremely grateful to our respected Director, **Prof. K. SANDHYA RANI** for fostering an excellent academic climate in our institution.

We also express  my sincere gratitude to our respected Head of the Department **Mr.P.HARINADHA** for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

We would like to convey thanks to our guide at college **Ms. SHAIK SHABANA** for her guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

Our sincere thanks to all the members who helped us directly and indirectly in the completion of project work. We express our profound gratitude to all our friends and family members for their encouragement.

**With sincere regards,**
Chakka Meghana
Purini Jnana Prasanna

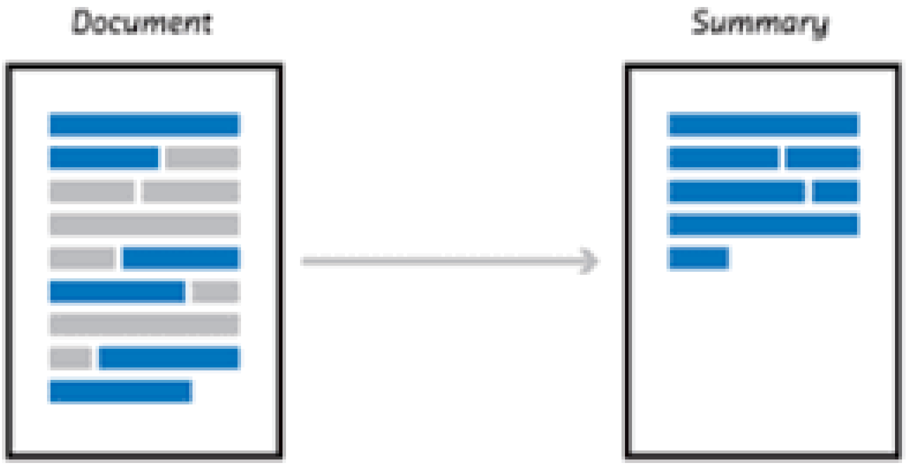
# INDEX

# ABSTRACT

In this new era, where tremendous information is available on the internet, it is most important to provide the improved mechanism to extract the information quickly and most efficiently. It is very difficult for human beings to manually extract the summary of a large documents of text. There are plenty of text material available on the Internet. So there is a problem of searching for relevant information from it. In order to solve the above two problems, the automatic text summarization is very much necessary. Automatic text summarization is basically summarizing of the given text using natural language processing and machine learning. There has been an explosion in the amount of text data from a variety of sources. This volume of text is an invaluable source of information and knowledge which needs to be effectively summarized to be useful. It is the process of identifying the most important and meaningful information in a document and compressing them into a shorter version preserving its overall meanings.

# INTRODUCTION

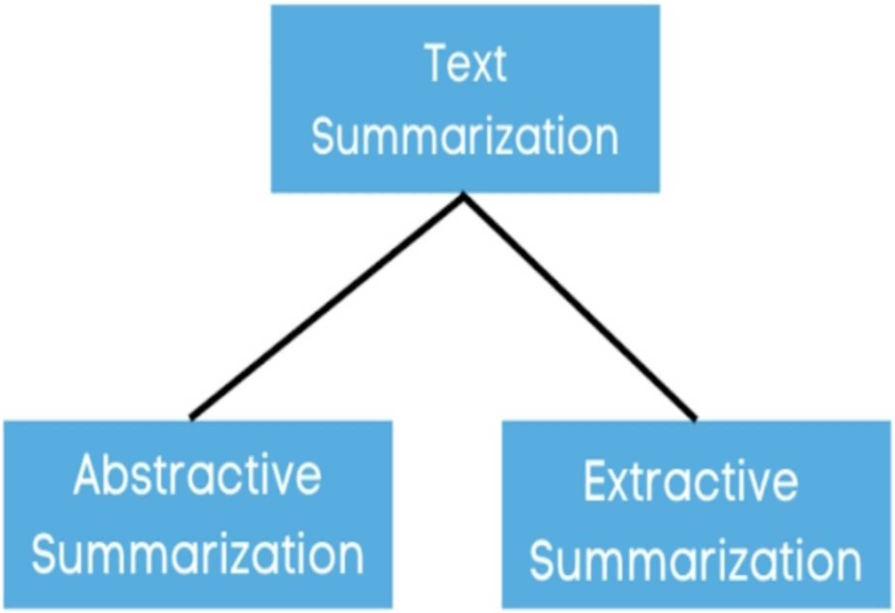
In the modern Internet age, textual data is ever increasing. Need some way to condense this data while preserving the information and meaning. We need to summarize textual data for that. Text summarization is the process of automatically generating natural language summaries from an input document while retaining the important points. It would help in easy and fast retrieval of information.



Summarization is a process of automatically condensing and rewriting a large chunk of text to create a small, crisp summary. A summarization system should give the reader most of the information present in the original document while also ensuring that no information has been lost during condensation.

Text summarization is commonly used by several websites and applications to create news feed and article summaries. It has become very essential for us due to our busy schedules. We prefer short summaries with all the important points over reading a whole report and summarizing it ourselves. So we are trying to automate the summarizing process.

There are two types of summarization algorithms .



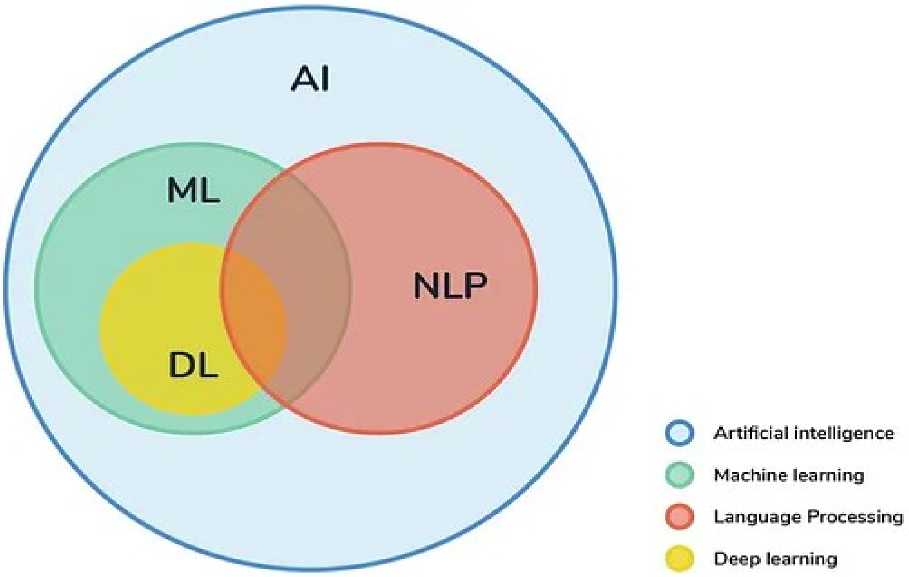
- Extractive summarization system forms summaries by copying parts of the source text through some measure of importance and then combine those parts together to render a summary.

- Abstractive summarization systems generate new phrases, possibly rephrasing or using words that were not in original text. Naturally abstractive approaches are harder when compared to extractive approach.

# MACHINE LEARNING AND NLP

Machine Learning : It is an application of AI that provides system the ability to automatically learn and improve from experience without being explicitly programmed. It can be used to help solve AI problems and to improve NLP by automating processors and delivering accurate responses.

NLP : It is a form of AI that gives machines the ability to not just read, but to understand and interpret human language. With NLP machines can make sense of written or spoken text and perform tasks including speech recognition, sentiment analysis, and automatic text summarization.



Machine Learning is a necessity for natural language processing . We use mathematics to improve to represent problems in physics as equations and use mathematical techniques like calculus to solve them. We formulate NLP problems using machine learning techniques. Machine Learning is considered a pre requisite for NLP as we used techniques like POS tagging, Bag of words, Word to vector for structuring text data.

Applying Machine Learning techniques to NLP problems would require converting unstructured text data into structured data. Machine Learning for NLP involves using statistical methods for identifying parts of speech, sentiments, entities, etc . . . These techniques are formulated as a model and then applied to other text data sets. This is called Supervised Learning. We can also use set of algorithms in large datasets to extract patterns and for decision making. This is known as unsupervised learning.

# OBJECTIVE

The objective of the project is to create a tool for text summarization. The concern in the automatic summarization is increasing broadly so the manual work is removed. The project concentrates on creating a tool which automatically summarizes the document.

# SCOPE

The project is wide in scope, all of the limitations stated below may seem to contradict that, but they are only restrictions applied.

- This project looks at single document summarization – the area of multi document summarization is not covered .
- The summaries produced are largely extracts of the document being summarized, rather than newely generated abstracts.
- The parameters used are optimal for news articles, although that can be changed easily.

# ADVANTAGES

- ➢ Summaries reduce reading time.
- ➢ When researching documents, summaries make the selection process easier.
- ➢ Automatic summarization algorithms are less biased than human summarizers.
- ➢ Computers are noticeably faster than humans and are capable of generating summaries faster.
- ➢ Makes information easier to find.

# DISADVANTAGES

- ➢ Possibility of missing relevant information.
- ➢ Can’t always tell what is Important.

# APPLICATIONS

◆ **Media monitoring :** Automatic summarization presents an opportunity to condense the continuous torrent of information into smaller pieces of information.

◆ **Financial research :** When you are a financial analyst looking at market reports and news every day. It is difficult to read every thing. Summarization helps in quick analysis of financial documents.

◆ **Medical cases :** Summarization can be a crucial component in the tele-health supply chain when it comes to analysing medical cases and routing these to be appropriate health professional.

◆ **Books and literature :** Summarization can help consumers quickly understand what a book is about as part of a buying process.

◆ **E-learning and class assignments :** Many teachers utilise case studies and news to frame thier lectures. Summarization can help teachers more quickly update their content by producing summarized reports on their subject of interest.

◆ **Patent research :** Researching patents can be a tedious process. Whether you are doing market intelligence research or looking to file a new patent, a summarizer to extract the most salient claims across patents could be a time saver.

◆ **Helping disabled people :** People with hearing disabilities could benefit from summarization to keep up with content in a more efficient way.

◆ **Science and R&D :** Systems that can group papers and further compress abstracts can become useful for this task.

◆ **Legal contract analysis :** Summarizer might add value by condensing a contract to the riskier clauses, or help you compare agreements.

# STEPS FOR TEXT SUMMARIZATION

## Step 1 : Getting our document / text

```
text = """There was a villager. He was illiterate. He did not know how to read and write. He
 often saw people wearing spectacles for reading books or papers. He thought, “If I have
spectacles, I can also read like these people. I must go to town and buy a pair of spectacles
for   myself.”
So one day he went to a town. He entered a spectacles shop He asked the shopkeeper for a
pair of spectacles for reading. The shopkeeper gave him various pairs of spectacles and a
book. The villager tried all the spectacles one by one. But he could not read anything. He
told the shopkeeper that all those spectacles were useless for him. The shopkeeper gave
him a doubtful look. Then he looked at the book. It was upside down! The shopkeeper said,
“Perhaps you don’t know how to read.”
The villager said, “No, I don’t. I want to buy spectacles so that I can read like others. But I
can’t read with any of these spectacles.” The shopkeeper controlled his laughter with great
difficulty when he learnt the real problem of his illiterate customer.
He explained to the villager, “My dear friend, you are very ignorant. Spectacles don’t help to
read or write. They only help you to see better. First of all you must learn to read and
write.”
Moral: Ignorance is blindness. """
```

Checking length of document or text

```
len(text)
       1246
```

## Step 2 : Importing the libraries

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

## Step 3 : Loading english tokenizer , tagger , parser and NER

➔ Tokenization : The process of segmenting a document / paragraph / text into words, sentences, punctuation marks , etc ... is called tokenization.

➔ Tagger : Assigning word types to tokens, like verb or noun.

➔ Parser : Assigning syntactic dependency lables, describing the relations between individual token, like subject or object.

➔ NER(Named Entity Recognition) : Labeling named “Real-World” objects, like persons, companies or locations.

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

**Step 4 : Calling the ‘nlp’ object on a string of text which will return a processed document**

```
doc=nlp(text)
```

performing word tokenization to check the tokens

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

Output :

```
['There', 'was', 'a', 'villager', '.', 'He', 'was', 'illiterate', '.', 'He', 'did', 'not', 'know', 'how', 'to', 'read',
'and', 'write', '.', 'He', 'often', 'saw', '\n', 'people', 'wearing', 'spectacles', 'for', 'reading', 'books', 'or',
'papers', '.', 'He', 'thought', ',', '“', 'If', 'I', 'have', 'spectacles', ',', 'I', '\n', 'can', 'also', 'read', 'like',
'these', 'people', '.', 'I', 'must', 'go', 'to', 'town', 'and', 'buy', 'a', 'pair', 'of', 'spectacles', 'for', 'myself',
     '.', '”', '\n', 'So', 'one', 'day', 'he', 'went', 'to', 'a', 'town', '.', 'He', 'entered', 'a', 'spectacles',
'shop', 'He', 'asked', 'the', 'shopkeeper', 'for', 'a', '\n', 'pair', 'of', 'spectacles', 'for', 'reading', '.',
'The','shopkeeper', 'gave', 'him',   'various', 'pairs', 'of', 'spectacles', 'and', 'a', '\n', 'book', '.', 'The',
'villager', 'tried', 'all', 'the','spectacles', 'one', 'by', 'one', '.', 'But', 'he', 'could', 'not',
'read','anything', '.', 'He', '\n', 'told', 'the',  'shopkeeper',   'that', 'all', 'those', 'spectacles', 'were',
'useless', 'for', 'him', '.', 'The', 'shopkeeper', 'gave', '\n', 'him', 'a', 'doubtful', 'look', '.', 'Then', 'he',
'looked', 'at', 'the', 'book', '.', 'It', 'was', 'upside', 'down', '!', 'The', 'shopkeeper', 'said', ',', '\n', '“',
'Perhaps', 'you', 'do', 'n’t', 'know', 'how', 'to', 'read', '.', '”', '\n', 'The',     'villager', 'said', ',', '“',
'No', ',', 'I', 'do', 'n’t', '.', 'I', 'want', 'to', 'buy', 'spectacles', 'so', 'that', 'I', 'can' 'read', 'like', 'others',
'.', 'But', 'I', '\n', 'ca', 'n’t', 'read', 'with', 'any', 'of', 'these', 'spectacles', '.', '”', 'The', 'shopkeeper',
'controlled', 'his', 'laughter', 'with', 'great', '\n', 'difficulty', 'when', 'he', 'learnt', 'the',     'real',
'problem', 'of', 'his', 'illiterate', 'customer', '.', '\n', 'He', 'explained', 'to', 'the', 'villager', ',', '“',
'My', 'dear', 'friend', ',', 'you', 'are', 'very', 'ignorant', '.', 'Spectacles', 'do', 'n’t', 'help', 'to', '\n',
'read',   'or', 'write', '.', 'They', 'only', 'help', 'you', 'to', 'see', 'better', '.', 'First', 'of', 'all', 'you',
'must', 'learn', 'to', 'read', 'and', '\n', 'write', '.', '”', '\n', 'Moral', ':', 'Ignorance', 'is', 'blindness', '.']
```

### Step 5 : Adding extra punctuations

```
punctuation=punctuation+'\n'
```

### Step 6 : Text pre-processing and cleaning

```
word_freq={}
stop_words=list(STOP_WORDS)
for word in doc:
  if word.text.lower() not in stop_words:
    if word.text.lower() not in punctuation:
      print(word)

word_freq={}
stop_words=list(STOP_WORDS)
for word in doc:
  if word.text.lower() not in stop_words:
    if word.text.lower() not in punctuation:
      if word.text not in word_freq.keys():
        word_freq[word.text]=1
      else:
          word_freq[word.text] +=1

print(word_freq)
```

Output :

```
{'villager': 4, 'illiterate': 2, 'know': 2, 'read': 8, 'write': 3, 'saw': 1, 'people': 2, 'wearing':
1,'spectacles': 10, 'reading': 2, 'books': 1, 'papers': 1, 'thought': 1, '“': 4, 'like': 2, 'town': 2, 'buy':
2,       'pair': 2, '”': 4, 'day': 1, 'went': 1, 'entered': 1, 'shop': 1, 'asked': 1, 'shopkeeper': 6, 'gave': 2,
'pairs': 1, 'book': 2, 'tried': 1, 'told': 1, 'useless': 1, 'doubtful': 1, 'look': 1, 'looked': 1, 'upside': 1,
'said': 2, 'want': 1, 'controlled': 1, 'laughter': 1, 'great': 1, 'difficulty': 1, 'learnt': 1, 'real': 1,
'problem': 1, 'customer': 1,          'explained': 1, 'dear': 1, 'friend': 1, 'ignorant': 1, 'Spectacles': 1,
'help': 2, 'better': 1, 'learn': 1, 'Moral': 1, 'Ignorance': 1, 'blindness': 1}
```

Checking the max count of words

```
max_freq=max(word_freq.values())
```

### Step 7 : Normalizing frequency counts

```
for word in word_freq.keys():
  word_freq[word]=word_freq[word]/max_freq

print(word_freq)
```

Output :

```
{'villager':  0.4,  'illiterate':  0.2,  'know':  0.2,  'read':  0.8,  'write':  0.3,  'saw':  0.1,  'people':  0.2,
'wearing':    0.1, 'spectacles': 1.0, 'reading': 0.2, 'books': 0.1, 'papers': 0.1, 'thought': 0.1, '“': 0.4,
'like': 0.2, 'town' : 0.2, 'buy': 0.2, 'pair': 0.2, '”': 0.4, 'day': 0.1, 'went': 0.1, 'entered': 0.1, 'shop':
0.1, 'asked': 0.1        'shopkeeper': 0.6, 'gave': 0.2, 'pairs': 0.1, 'book': 0.2, 'tried': 0.1, 'told': 0.1,
'useless':  0.1,  'doubtful':       0.1,  'look':  0.1,  'looked':  0.1,  'upside':  0.1,  'said':  0.2,  'want':  0.1,
'controlled': 0.1, 'laughter': 0.1, 'great': 0.1, 'difficulty': 0.1, 'learnt': 0.1, 'real': 0.1, 'problem': 0.1,
'customer':  0.1,  'explained':  0.1,           'dear':  0.1,  'friend':  0.1,  'ignorant':  0.1,  'Spectacles':  0.1,
'help': 0.2, 'better': 0.1, 'learn': 0.1, 'Moral':          0.1, 'Ignorance': 0.1, 'blindness': 0.1}
```

### Step 8 : Sentence tokenization

```
sent_tokens=[sent for sent in doc.sents]

print(sent_tokens)
```

Output :

```
[There was a villager., He was illiterate., He did not know how to read and write
people wearing spectacles for reading books or papers., He thought, “If I have spectacles
can also read like these people., I must go to town and buy a pair of spectacles for myself
, So one day he went to a town., He entered a spectacles shop He asked the shopkeeper
pair of spectacles for reading., The shopkeeper gave him various pairs of spectacles
book., The villager tried all the spectacles one by one., But he could not read any thing
told the shopkeeper that all those spectacles were useless for him., The shopkeeper
him a doubtful look., Then he looked at the book., It was upside down!, The shopkeeper said,
“Perhaps you don’t know how to read.”
, The villager said, “No, I don’t., I want to buy spectacles so that I can read like other. But I
can’t read with any of these spectacles.”, The shopkeeper controlled his laughter with great
difficulty when he learnt the real problem of his illiterate customer.
, He explained to the villager, “My dear friend, you are very ignorant., Spectacles don’t help
read or write., They only help you to see better., First of all you must learn to read and
write.”
, Moral: Ignorance is blindness.]
```

**Step 9 : Finding sentence scores**

```
sent_score={}

for sent in sent_tokens:
  for word in sent:
    print(word)

for sent in sent_tokens:
  for word in sent:
    if word.text.lower() in word_freq.keys():
      if sent not in sent_score.keys():
        sent_score[sent]=word_freq[word.text.lower()]
      else:
        sent_score[sent] +=word_freq[word.text.lower()]

print(sent_score)
```

Output :

```
{There was a villager.: 0.4, He was illiterate.: 0.2, He did not know how to read
people wearing spectacles for reading books or papers.: 1.8, He thought, “If I hav
can also read like these people.: 2.7, I must go to town and buy a pair of spectac
: 2.0, So one day he went to a town.: 0.4, He entered a spectacles shop He asked t
pair of spectacles for reading.: 3.3000000000000007, The shopkeeper gave him vario
book.: 2.1, The villager tried all the spectacles one by one.: 1.5, But he could n
told the shopkeeper that all those spectacles were useless for him.: 1.8, The shop
him a doubtful look.: 1.0, Then he looked at the book.: 0.30000000000000004, It wa
“Perhaps you don’t know how to read.”
: 2.6, The villager said, “No, I don’t.: 1.0, I want to buy spectacles so that I c
can’t read with any of these spectacles.”: 2.2, The shopkeeper controlled his laug
difficulty when he learnt the real problem of his illiterate customer.
: 1.6, He explained to the villager, “My dear friend, you are very ignorant.: 1.20
read or write.: 2.3, They only help you to see better.: 0.30000000000000004, First
write.”
: 1.6, Moral: Ignorance is blindness.: 0.1}
```

**Step 10 : Selecting 30% sentences with maximum score**

```
from heapq import nlargest

len(sent_score)*0.3

summary=nlargest(n=6,iterable=sent_score,key=sent_score.get)

print(summary)
```

Output :

```
[He entered a spectacles shop He asked the shopkeeper for a
pair of spectacles for reading., He thought, “If I have spectacles, I
can also read like these people., The shopkeeper said,
“Perhaps you don’t know how to read.”
, I want to buy spectacles so that I can read like others., Spectacles don’t help
read or write., But I
can’t read with any of these spectacles.”]
```

**Step 11 : Creating the Final Summary**

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

print(final_summary)
```

Output :

```
['He entered a spectacles shop He asked the shopkeeper for a\npair of spectacles for reading.',
'He    thought, “If I have spectacles, I\ncan also read like these people.', 'The shopkeeper said,\
n“Perhaps  you don’t know how to read.”\n', 'I want to buy spectacles so that I can read like
others.',       'Spectacles don’t help to\nread or write.', 'But I\ncan’t read with any of these
spectacles.”'
```

**Step 12 : Finding length of summary**

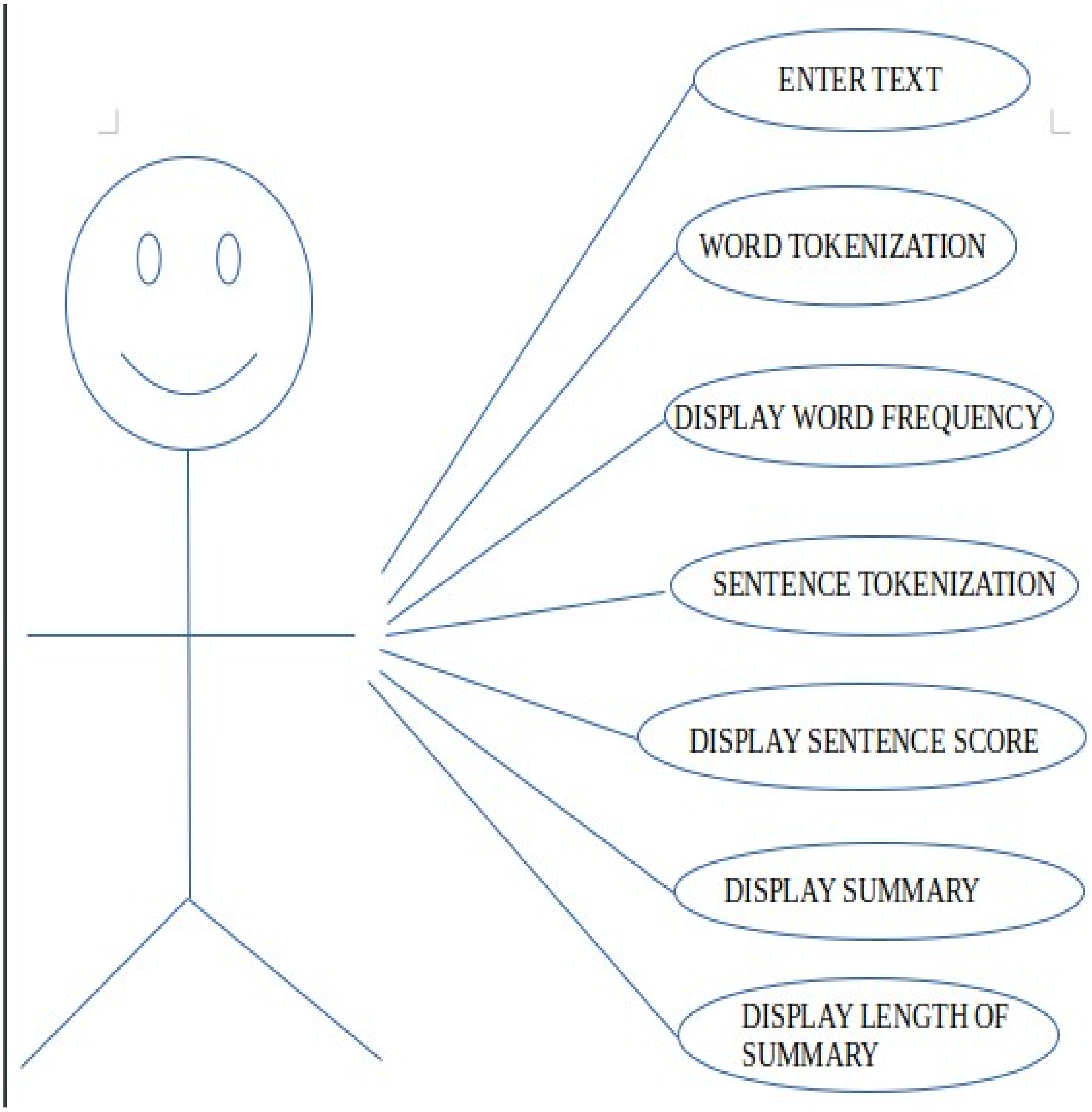
```
summary=" ".join(final_summary)

len(summary)
```

Output :

```
365
```

# USE CASE DIAGRAM



# MODULES AND LIBRARIES USED

## Spacy :

It is an open source software library for advanced natural language processing written in the programming languages python and cython. The library is published under the MIT(Massachusetts Institute of Technology) license and it’s main developers are Mathew Honnibal and Ines Monty, the founders of the software company explosion. Unlike NLTK, which is widely used for teaching and research, spacy focuses on providing software for production usage.

## String :

It is a built-in module that contains some constants, utility functions and classes for sting manipulation. Some of them are :

ascii_letters
ascii_uppercase
punctuation
whitespace

## Heapq :

Heap datastructure is mainly used to represent a priority queue. In python, it is available using the “heapq” module. The property of this data structure in python is that each time the smallest heap element is popped. Whenever elements are pushed or popped, heap structure is maintained. The heap[0] element also returns the smallest element each time.

## Spacy.lang.en.stop_words :

It contains the english stopwords. Similarly, we can access stopwords for any language using it’s extension such as ‘en’ for english language.

### stop_words :

Stop words are set of commonly used words in a language. Examples of stop words in english are “a”, “the”, “is”, “are” and etc ... Stop words are commonly used in text mining and natural language processing to eliminate words that are commonly used that they carry very little useful information.

### Punctuation :

In python string.punctuation is a pre initialised string used as string constant. In python string.punctuation will give the all set of punctuations. It doesn’t take any parameter since it’s not a function.

### Load :

spacy.load() is a convenience wrapper that reads the pipeline’s config.cfg, uses the language and pipeline information to construct a language object, loads in the model data and weights, and returns it.

### en_core_web_sm :

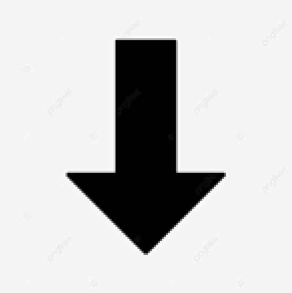
It is a smaller english pipeline trained on written web text, that includes vocabulary, syntax and entities.

### nlargest(k, iterable, key = fun) :

This function is used to return the k largest elements from the iterable specified like a list, tuple and others. The function nlargest() can also be passed a key function that returns a comparison key to be used in the sorting.

# FINAL OUTPUT

```
"""There was a villager.He was illiterate. He did not know how to read and write.He often
saw people wearing spectacles for reading books or papers.He thought,“If I have spectacles, I
can also read like these people. I must go to town and buy a pair of spectacles for myself.”
So one day he went to a town. He entered a spectacles shop He asked the shopkeeper for a
pair of spectacles for reading. The shopkeeper gave him various pairs of spectacles and a
book. The villager tried all the spectacles one by one. But he could not read anything. He
told the shopkeeper that all those spectacles were useless for him. The shopkeeper gave
him a doubtful look. Then he looked at the book. It was upside down! The shopkeeper said,
“Perhaps you don’t know how to read.”
The villager said, “No, I don’t. I want to buy spectacles so that I can read like others. But I
can’t read with any of these spectacles.” The shopkeeper controlled his laughter with great
difficulty when he learnt the real problem of his illiterate customer.
He explained to the villager, “My dear friend, you are very ignorant. Spectacles don’t help to
read or write. They only help you to see better. First of all you must learn to read and
write.”
Moral: Ignorance is blindness. """
```



```
['He entered a spectacles shop He asked the shopkeeper for a\npair of spectacles for reading.',
'He thought, “If I have spectacles, I\ncan also read like these people.', 'The shopkeeper said,\
n“Perhaps  you don’t know how to read.”\n', 'I want to buy spectacles so that I can read like
others.','Spectacles don’t help to\nread or write.', 'But I\ncan’t read with any of these
spectacles.”']
```

# CONCLUSION

Text summarization is one of the major problems in the field of natural language processing. As with time internet is growing at a very fast rate and with it data and information is also increasing, it will going to be difficult for human to summarize large amount of data. Thus there is a need of automatic text summarization because of this huge amount of data. We have learned all the basics of Extractive and Abstractive Method of automatic text summarization and tried to implement extractive one. We have made a basic automatic text summarizer using spacy library using python and it is working on small documents. We have used Extractive approach to do text summarization.

# REFERENCES

1. https://machinelearningmastery.com

2. https://www.google.com

3. https://www.encora.com

4. https://www.projectpro.io

5. https://aparnamishra144.medium.com

6. https://data-flair.training

7. https://www.jespublication.com

8. https://thecleverprogrammer.com