

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

text = """summarization is a process within NLP that involves condensing a
longer piece of text, such as an article, blog post, or document, into a
shorter version while retaining its most important information and main
ideas.

There are two main types of summarization:
Extractive Summarization: In this approach, the summary is generated by
selecting and extracting sentences directly from the original text. These
selected sentences are usually the ones that contain the most important
information or are representative of the main ideas. Extractive
summarization methods often involve ranking sentences based on their
relevance and then choosing the top-ranked sentences for the summary.
Abstractive Summarization: Abstractive summarization involves generating a
summary that might include new sentences that are not present in the
original text. This approach goes beyond simply extracting sentences and
aims to generate coherent and concise sentences that capture the essence
of the original text. Abstractive summarization requires a more
sophisticated understanding of language and context, often involving
techniques like natural language generation and language modeling.

Summarization in NLP is a challenging task as it requires the model to
understand the context, identify important information, and present it in
a coherent and concise manner. Researchers have developed various
algorithms and techniques to improve the quality of summaries generated by
NLP models, and these models have found applications in content
summarization, news aggregation, document summarization, and more."""

# def summarizer(rawdocs):
stopwords = list(STOP_WORDS)
# print(stopwords)
nlp = spacy.load('en_core_web_sm')
doc = nlp(text)
# print(doc)
tokens = [token.text for token in doc]
# print(tokens)
```

```

word_freq = {}
for word in doc:
    if word.text.lower() not in stopwords and 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)

max_freq = max(word_freq.values())
# print(max_freq)

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

sent_tokens = [sent for sent in doc.sents]
# print(sent_tokens)

sent_scores = {}
for sent in sent_tokens:
    for word in sent:
        if word.text in word_freq.keys():
            if sent not in sent_scores.keys():
                sent_scores[sent]= word_freq[word.text]
            else:
                sent_scores[sent] += word_freq[word.text]
# print(sent_scores)

select_len = int(len(sent_tokens) * 0.3)
# print(select_len)

summary = nlargest(select_len, sent_scores, key = sent_scores.get)
# print(summary)

final_summary = [word.text for word in summary]
summary = ' '.join(final_summary)
print(text)
print(summary)

print("length of original text ",len(text.split(' ')))
print("length of summary text ",len(summary.split(' ')))

```

Input Text:

```
[Running] python -u "c:\Users\saiye\Downloads\text_summary\text_summary.py"
summarization is a process within NLP that involves condensing a longer piece of text, such as an article, blog post, or document, into a shorter version while retaining its most important information and main ideas.
There are two main types of summarization:
Extractive Summarization: In this approach, the summary is generated by selecting and extracting sentences directly from the original text. These selected sentences are usually the ones that contain the most important information or are representative of the main ideas. Extractive summarization methods often involve ranking sentences based on their relevance and then choosing the top-ranked sentences for the summary.
Abstractive Summarization: Abstractive summarization involves generating a summary that might include new sentences that are not present in the original text. This approach goes beyond simply extracting sentences and aims to generate coherent and concise sentences that capture the essence of the original text. Abstractive summarization requires a more sophisticated understanding of language and context, often involving techniques like natural language generation and language modeling.
```

Summarize text:

```
There are two main types of summarization:
Extractive Summarization: In this approach, the summary is generated by selecting and extracting sentences directly from the original text. summarization is a process within NLP that involves condensing a longer piece of text, such as an article, blog post, or document, into a shorter version while retaining its most important information and main ideas.
Researchers have developed various algorithms and techniques to improve the quality of summaries generated by NLP models, and these models have found applications in content summarization, news aggregation, document summarization, and more.
length of original text 222
length of summary text 92
```

Full & short Length:

```
[Running] python -u "c:\Users\saiye\Downloads\text_summary\text_summary.py"
summarization is a process within NLP that involves condensing a longer piece of text, such as an article, blog post, or document, into a shorter version while retaining its most important information and main ideas.
There are two main types of summarization:
Extractive Summarization: In this approach, the summary is generated by selecting and extracting sentences directly from the original text. These selected sentences are usually the ones that contain the most important information or are representative of the main ideas. Extractive summarization methods often involve ranking sentences based on their relevance and then choosing the top-ranked sentences for the summary.
Abstractive Summarization: Abstractive summarization involves generating a summary that might include new sentences that are not present in the original text. This approach goes beyond simply extracting sentences and aims to generate coherent and concise sentences that capture the essence of the original text. Abstractive summarization requires a more sophisticated understanding of language and context, often involving techniques like natural language generation and language modeling.

Summarization in NLP is a challenging task as it requires the model to understand the context, identify important information, and present it in a coherent and concise manner. Researchers have developed various algorithms and techniques to improve the quality of summaries generated by NLP models, and these models have found applications in content summarization, news aggregation, document summarization, and more.
```