```
text = ""


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


stopwords = list(STOP_WORDS)


# clear the NLP Model by using (en_core_web_sm)
nlp = spacy.load('en_core_web_sm')


# inside this nlp pass the text
doc = nlp(text)


# build list of the token by calling [token = token.text for token in doc]

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


# remove the stop_words and punctuation (text-cleaning)
punctuation = punctuation + '\n'
punctuation


# build the word frequency
# we need to calculate each of these words how many times has occurred in this given text and then we based on those frequency
# we will indentify that the most important sentance.

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

# if any keys is being introduce the first time the word of that occurrence will be
# equal to 1 but after the first time it is being introduced for second and the third time
# it will just increment 1 in already present key in word_frequency.



print(word_frequencies)


max_frequency = max(word_frequencies.values())


max_frequency

     4


# now we are going divied each of this values by max_frequency so that a nomalized
# frequency can achieved

for word in word_frequencies.keys():
  word_frequencies[word] = word_frequencies[word] / max_frequency


print(word_frequencies)


# Sentence tokenization
sentence_tokens = [sent for sent in doc.sents]
print(sentence_tokens)


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

```
sentence_scores
```

```
# now get 30% of sentence with maximum score
from heapq import nlargest
```

```
select_length = int(len(sentence_tokens)*0.3)
select_length
```

```
# now we have to select (select_length) sentence which have maximum frequency count

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

```
summary
```

```
# After we got the most importent sentences we have to combined them
final_summary = [word.text for word in summary]
```

```
final_summary
```

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

```
print(text)
```

```
print(summary)
```

```
# compare original text and summary text
len(text)
```

```
593
```

```
len(summary)
```

```
0
```