Write a program for pre-processing of a text document such as stop word removal, stemming.

Import Necessary Libraries

```
import string
```

```
import nltk
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
True
```

```
nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

```
from nltk.corpus import stopwords
```

```
from nltk.tokenize import word_tokenize
```

```
from nltk.stem import PorterStemmer
ps = PorterStemmer()
```

Read text from file

```
text = ""
with open("Information_Retrieval.txt") as file:
  for line in file:
    text += line
```

```
text
```

```
'Information retrieval (IR) in computing and information science is the task of identifying and retrieving information system resources that are relevant to an information need. The information need can be specified in the form of a search query. In the case of document retrieval, queries can be based on full-text or other content-based indexing. Information retrieval is the science of searching for inf ormation in a document, searching for documents themselves, and also searching for the metadata that describes data, and for databases of texts, images or sounds.\n\nAutomated information retrieval systems are used to reduce what has been called information overload. An IR system is a software system that provides access to books, journals and other documents; it also stores and manages those documents. Web search engines are the most visible IR applications.\n\nOverview\nAn information retrieval process begins when a user enters a quer y into the system. Queries are formal statements of i. '
```

Word Tokenization

```
word_token = word_tokenize(text)
```

```
word_token
```

```
['Information',
 'retrieval',
 '(',
 'IR',
 ')',
 'in',
 'computing',
 'and',
 'information',
```

```
      'science',
      'is',
      'the',
      'task',
      'of',
      'identifying',
      'and',
      'retrieving',
      'information',
      'system',
      'resources',
      'that',
      'are',
      'relevant',
      'to',
      'an',
      'information',
      'need',
      '.',
      'The',
      'information',
      'need',
      'can',
      'be',
      'specified',
      'in',
      'the',
      'form',
      'of',
      'a',
      'search',
      'query',
      '.',
      'In',
      'the',
      'case',
      'of',
      'document',
      'retrieval',
      ',',
      'queries',
      'can',
      'be',
      'based',
      'on',
      'full-text',
      'or',
      'other',
      'content-based',
```

Removing Punctuations

```
def remove_punctuations(words):
  return [word for word in words if word not in string.punctuation]


clean = remove_punctuations(word_token)


clean
```

Show hidden output

Stopwords Removal

```
swords = stopwords.words("english")


def remove_stopwords(clean_words):
  return [word for word in clean_words if word.lower() not in swords]


removed = remove_stopwords(clean)


removed
```

Show hidden output

Stemming using PorterStemmer

```
def stemming(cleaned):
  return [ps.stem(stem) for stem in cleaned]


stemmed = stemming(removed)


stemmed
```

Show hidden output