Extract Sample document and apply Create representation of document by calculat Document Frequency following document preprocessing methods: Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization. Create representation of document by calculating Term Frequency and Inverse Document Frequency

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
text = '''It was a Thursday, but it felt like a Monday to John. And John lov
I should probably get another latte. I've just been sitting here with this e
John was always impatient on the weekends; he missed the formal structure of
Jesus, I've written another loser. '''
```

Tokenization of text

In [3]:
```
text_split = text.split()
```

In [4]:
```
text
```

Out[4]:
```
'It was a Thursday, but it felt like a Monday to John. And John loved Monda
ys. He\nI should probably get another latte. I've just been sitting here wi
th this empty cup. But\nJohn was always impatient on the weekends; he misse
d the formal structure of the business w\nJesus, I've written another lose
r. '
```

In [5]:
```
!pip install nltk
```

```
Defaulting to user installation because normal site-packages is not writeabl
e
Collecting nltk
  Downloading nltk-3.8.1-py3-none-any.whl (1.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.5/1.5 MB 7.8 MB/s eta 0:00:0
0[36m0:00:01[36m0:00:01:01
Requirement already satisfied: joblib in /home/mca01/.local/lib/python3.10/s
ite-packages (from nltk) (1.4.0)
Requirement already satisfied: click in /usr/lib/python3/dist-packages (from
nltk) (8.0.3)
Collecting regex>=2021.8.3
  Downloading regex-2023.12.25-cp310-cp310-manylinux_2_17_x86_64.manylinux20
14_x86_64.whl (773 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 774.0/774.0 KB 17.9 MB/s eta 0:0
0:0031m25.2 MB/s eta 0:00:01
Collecting tqdm
  Downloading tqdm-4.66.2-py3-none-any.whl (78 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 78.3/78.3 KB 15.6 MB/s eta 0:0
0:00
Installing collected packages: tqdm, regex, nltk
  WARNING: The script tqdm is installed in '/home/mca01/.local/bin' which is
not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
warning, use --no-warn-script-location.
  WARNING: The script nltk is installed in '/home/mca01/.local/bin' which is
not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
warning, use --no-warn-script-location.
Successfully installed nltk-3.8.1 regex-2023.12.25 tqdm-4.66.2
```

In [6]:
```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package stopwords to /home/mca01/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /home/mca01/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /home/mca01/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
```

Out[6]: True

In [7]:
```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
stop_words = stopwords.words('english')
```

In [8]:
```
tokenized = sent_tokenize(text)
for i in tokenized:
# Word tokenizers is used to find the words
# and punctuation in a string
    wordsList = nltk.word_tokenize(i)
# removing stop words from wordList
    wordsList = [w for w in wordsList if not w in stop_words]
```

```
# Using a Tagger. Which is part-of-speech
# tagger or POS-tagger.
    tagged = nltk.pos_tag(wordsList)
    print(tagged)
```

```
[('It', 'PRP'), ('Thursday', 'NNP'), (',', ','), ('felt', 'VBD'), ('like',
'IN'), ('Monday', 'NNP'), ('John', 'NNP'), ('.', '.')]
[('And', 'CC'), ('John', 'NNP'), ('loved', 'VBD'), ('Mondays', 'NNP'), ('.',
'.')]
[('He', 'PRP'), ('I', 'PRP'), ('probably', 'RB'), ('get', 'VB'), ('another',
'DT'), ('latte', 'NN'), ('.', '.')]
[('I', 'PRP'), ('’', 'VBP'), ('sitting', 'VBG'), ('empty', 'JJ'), ('cup', 'N
N'), ('.', '.')]
[('But', 'CC'), ('John', 'NNP'), ('always', 'RB'), ('impatient', 'JJ'), ('we
ekends', 'NNS'), (';', ':'), ('missed', 'VBN'), ('formal', 'JJ'), ('structur
e', 'NN'), ('business', 'NN'), ('w', 'NN'), ('Jesus', 'NNP'), (',', ','),
('I', 'PRP'), ('’', 'VBP'), ('written', 'VBN'), ('another', 'DT'), ('loser',
'NN'), ('.', '.')]
```

In [9]:
```
stopwords
```

Out[9]:
```
<WordListCorpusReader in '/home/mca01/nltk_data/corpora/stopwords'>
```

In [10]:
```
print(stopwords)
```

```
<WordListCorpusReader in '/home/mca01/nltk_data/corpora/stopwords'>
```

Stemming and Lemmatization

1. Stemming

In [11]:
```
from nltk.stem.porter import PorterStemmer
porter_stemmer = PorterStemmer()
nltk_token = nltk.word_tokenize(text)
```

In [14]:
```
for w in nltk_token:
    print("Actual : %s , Stem: %s" %(w, porter_stemmer.stem(w)))
```

```
Actual : It , Stem: it
Actual : was , Stem: wa
Actual : a , Stem: a
Actual : Thursday , Stem: thursday
Actual : , , Stem: ,
Actual : but , Stem: but
Actual : it , Stem: it
Actual : felt , Stem: felt
Actual : like , Stem: like
Actual : a , Stem: a
Actual : Monday , Stem: monday
Actual : to , Stem: to
Actual : John , Stem: john
Actual : . , Stem: .
Actual : And , Stem: and
Actual : John , Stem: john
Actual : loved , Stem: love
Actual : Mondays , Stem: monday
Actual : . , Stem: .
Actual : He , Stem: he
Actual : I , Stem: i
Actual : should , Stem: should
Actual : probably , Stem: probabl
Actual : get , Stem: get
Actual : another , Stem: anoth
Actual : latte , Stem: latt
Actual : . , Stem: .
Actual : I , Stem: i
Actual : ’ , Stem: ’
Actual : ve , Stem: ve
Actual : just , Stem: just
Actual : been , Stem: been
Actual : sitting , Stem: sit
Actual : here , Stem: here
Actual : with , Stem: with
Actual : this , Stem: thi
Actual : empty , Stem: empti
Actual : cup , Stem: cup
Actual : . , Stem: .
Actual : But , Stem: but
Actual : John , Stem: john
Actual : was , Stem: wa
Actual : always , Stem: alway
Actual : impatient , Stem: impati
Actual : on , Stem: on
Actual : the , Stem: the
Actual : weekends , Stem: weekend
Actual : ; , Stem: ;
Actual : he , Stem: he
Actual : missed , Stem: miss
Actual : the , Stem: the
Actual : formal , Stem: formal
Actual : structure , Stem: structur
Actual : of , Stem: of
Actual : the , Stem: the
Actual : business , Stem: busi
```

```
Actual : w , Stem: w
Actual : Jesus , Stem: jesu
Actual : , , Stem: ,
Actual : I , Stem: i
Actual : ’ , Stem: ’
Actual : ve , Stem: ve
Actual : written , Stem: written
Actual : another , Stem: anoth
Actual : loser , Stem: loser
Actual : . , Stem: .
```

2.Lemmatization

In [15]:
```
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
```

In [16]:
```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /home/mca01/nltk_data...
```

Out[16]:
```
True
```

In [17]:
```
for w in nltk_token:
    print("Actual : %s , Lemme: %s" %(w, wordnet_lemmatizer.lemmatize(w)))
```

```
Actual : It , Lemme: It
Actual : was , Lemme: wa
Actual : a , Lemme: a
Actual : Thursday , Lemme: Thursday
Actual : , , Lemme: ,
Actual : but , Lemme: but
Actual : it , Lemme: it
Actual : felt , Lemme: felt
Actual : like , Lemme: like
Actual : a , Lemme: a
Actual : Monday , Lemme: Monday
Actual : to , Lemme: to
Actual : John , Lemme: John
Actual : . , Lemme: .
Actual : And , Lemme: And
Actual : John , Lemme: John
Actual : loved , Lemme: loved
Actual : Mondays , Lemme: Mondays
Actual : . , Lemme: .
Actual : He , Lemme: He
Actual : I , Lemme: I
Actual : should , Lemme: should
Actual : probably , Lemme: probably
Actual : get , Lemme: get
Actual : another , Lemme: another
Actual : latte , Lemme: latte
Actual : . , Lemme: .
Actual : I , Lemme: I
Actual : ’ , Lemme: ’
Actual : ve , Lemme: ve
Actual : just , Lemme: just
Actual : been , Lemme: been
Actual : sitting , Lemme: sitting
Actual : here , Lemme: here
Actual : with , Lemme: with
Actual : this , Lemme: this
Actual : empty , Lemme: empty
Actual : cup , Lemme: cup
Actual : . , Lemme: .
Actual : But , Lemme: But
Actual : John , Lemme: John
Actual : was , Lemme: wa
Actual : always , Lemme: always
Actual : impatient , Lemme: impatient
Actual : on , Lemme: on
Actual : the , Lemme: the
Actual : weekends , Lemme: weekend
Actual : ; , Lemme: ;
Actual : he , Lemme: he
Actual : missed , Lemme: missed
Actual : the , Lemme: the
Actual : formal , Lemme: formal
Actual : structure , Lemme: structure
Actual : of , Lemme: of
Actual : the , Lemme: the
Actual : business , Lemme: business
```

```
Actual : w , Lemme: w
Actual : Jesus , Lemme: Jesus
Actual : , , Lemme: ,
Actual : I , Lemme: I
Actual : ’ , Lemme: ’
Actual : ve , Lemme: ve
Actual : written , Lemme: written
Actual : another , Lemme: another
Actual : loser , Lemme: loser
Actual : . , Lemme: .
```

In [ ]: