practical7

May 4, 2024

[2]:
```
import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Prathamesh\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Prathamesh\AppData\Roaming\nltk_data…
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Prathamesh\AppData\Roaming\nltk_data…
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Prathamesh\AppData\Roaming\nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

[2]:
```
True
```

[3]:
```
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.probability import FreqDist
from nltk.corpus import wordnet
import math

# Sample document
sample_document = "Text analytics is the process of analyzing unstructured text␣
 ↪data to extract \
               relevant information. It involves several preprocessing steps␣
 ↪such as tokenization, \
               POS tagging, stop words removal, stemming, and lemmatization."

# Tokenization
tokens = word_tokenize(sample_document)
```

```
# POS Tagging
pos_tags = nltk.pos_tag(tokens)

# Stopwords Removal
stop_words = set(stopwords.words('english'))
filtered_tokens = [word for word in tokens if word.lower() not in stop_words]

# Stemming
ps = PorterStemmer()
stemmed_tokens = [ps.stem(word) for word in filtered_tokens]

# Lemmatization
wnl = WordNetLemmatizer()
lemmatized_tokens = [wnl.lemmatize(word, pos='v') for word in filtered_tokens]

# Term Frequency Calculation
tf = FreqDist(lemmatized_tokens)

# Inverse Document Frequency Calculation
def idf(term, documents):
    doc_with_term = sum(1 for doc in documents if term in doc)
    if doc_with_term == 0:
        return 0
    else:
        return math.log(len(documents) / doc_with_term)

# Example collection of documents
documents = [
    "Text analytics is the process of analyzing unstructured text data to␣
 ↪extract relevant information.",
    "Text analytics involves several preprocessing steps such as tokenization,␣
 ↪POS tagging, stop words removal, stemming, and lemmatization.",
    "Text analytics helps in extracting insights from large volumes of text␣
 ↪data for various applications.",
]

# Calculate IDF for each term in the collection of documents
idf_scores = {}
for doc in documents:
    doc_tokens = word_tokenize(doc)
    doc_tokens = [wnl.lemmatize(word.lower(), pos='v') for word in doc_tokens␣
 ↪if word.lower() not in stop_words]
    for term in set(doc_tokens):
        idf_scores[term] = idf(term, documents)

# Print results
```

```
print("Tokenization:", tokens)
print("\nPOS Tagging:", pos_tags)
print("\nStopwords Removal:", filtered_tokens)
print("\nStemming:", stemmed_tokens)
print("\nLemmatization:", lemmatized_tokens)
print("\nTerm Frequency:", tf.most_common())
print("\nInverse Document Frequency:")
for term, score in idf_scores.items():
    print(term, ":", score)
```

```
Tokenization: ['Text', 'analytics', 'is', 'the', 'process', 'of', 'analyzing',
'unstructured', 'text', 'data', 'to', 'extract', 'relevant', 'information', '.',
'It', 'involves', 'several', 'preprocessing', 'steps', 'such', 'as',
'tokenization', ',', 'POS', 'tagging', ',', 'stop', 'words', 'removal', ',',
'stemming', ',', 'and', 'lemmatization', '.']

POS Tagging: [('Text', 'NN'), ('analytics', 'NNS'), ('is', 'VBZ'), ('the',
'DT'), ('process', 'NN'), ('of', 'IN'), ('analyzing', 'VBG'), ('unstructured',
'JJ'), ('text', 'NN'), ('data', 'NNS'), ('to', 'TO'), ('extract', 'VB'),
('relevant', 'JJ'), ('information', 'NN'), ('.', '.'), ('It', 'PRP'),
('involves', 'VBZ'), ('several', 'JJ'), ('preprocessing', 'VBG'), ('steps',
'NNS'), ('such', 'JJ'), ('as', 'IN'), ('tokenization', 'NN'), (',', ','),
('POS', 'NNP'), ('tagging', 'NN'), (',', ','), ('stop', 'VB'), ('words', 'NNS'),
('removal', 'JJ'), (',', ','), ('stemming', 'VBG'), (',', ','), ('and', 'CC'),
('lemmatization', 'NN'), ('.', '.')]

Stopwords Removal: ['Text', 'analytics', 'process', 'analyzing', 'unstructured',
'text', 'data', 'extract', 'relevant', 'information', '.', 'involves',
'several', 'preprocessing', 'steps', 'tokenization', ',', 'POS', 'tagging', ',',
'stop', 'words', 'removal', ',', 'stemming', ',', 'lemmatization', '.']

Stemming: ['text', 'analyt', 'process', 'analyz', 'unstructur', 'text', 'data',
'extract', 'relev', 'inform', '.', 'involv', 'sever', 'preprocess', 'step',
'token', ',', 'po', 'tag', ',', 'stop', 'word', 'remov', ',', 'stem', ',',
'lemmat', '.']

Lemmatization: ['Text', 'analytics', 'process', 'analyze', 'unstructured',
'text', 'data', 'extract', 'relevant', 'information', '.', 'involve', 'several',
'preprocessing', 'step', 'tokenization', ',', 'POS', 'tag', ',', 'stop', 'word',
'removal', ',', 'stem', ',', 'lemmatization', '.']

Term Frequency: [(',', 4), ('.', 2), ('Text', 1), ('analytics', 1), ('process',
1), ('analyze', 1), ('unstructured', 1), ('text', 1), ('data', 1), ('extract',
1), ('relevant', 1), ('information', 1), ('involve', 1), ('several', 1),
('preprocessing', 1), ('step', 1), ('tokenization', 1), ('POS', 1), ('tag', 1),
('stop', 1), ('word', 1), ('removal', 1), ('stem', 1), ('lemmatization', 1)]
```

```
Inverse Document Frequency:
unstructured : 1.0986122886681098
text : 0.4054651081081644
information : 1.0986122886681098
. : 0.0
process : 0.4054651081081644
analyze : 0
analytics : 0.0
data : 0.4054651081081644
relevant : 1.0986122886681098
extract : 0.4054651081081644
step : 1.0986122886681098
, : 1.0986122886681098
several : 1.0986122886681098
stop : 1.0986122886681098
removal : 1.0986122886681098
tag : 1.0986122886681098
lemmatization : 1.0986122886681098
preprocessing : 1.0986122886681098
word : 1.0986122886681098
pos : 0
involve : 1.0986122886681098
tokenization : 1.0986122886681098
stem : 1.0986122886681098
applications : 1.0986122886681098
insights : 1.0986122886681098
volumes : 1.0986122886681098
help : 1.0986122886681098
large : 1.0986122886681098
various : 1.0986122886681098
```