LAB3 Report

Introduction:

This report details the findings and outcomes of the two exercises conducted within LAB3. Each exercise focuses on Natural Language Processing (NLP) techniques, specifically text processing and part-of-speech (POS) tagging.

Exercise 1: Text Processing

1. Data Acquisition and Preprocessing:

The webtext corpus was downloaded and accessed using the firefox.txt file.

2. Vocabulary Size:

The initial vocabulary size, representing the unique words in the corpus, was 9300.

3. Stemming:

Porter and Snowball stemmers were employed to reduce words to their root forms. Examples were provided to illustrate their behavior on different word types.

Example of Porter stemmer:

['cooki', 'manag', ':', '``', 'do', "n't", 'allow', 'site', 'that', 'set']

Example of Snowball stemmer:
['cooki', 'manag', ':', '``', 'do', "n't", 'allow', 'site', 'that', 'set']

4. Vocabulary size :

The vocabulary size after stemming with Porter stemmer was 5847 and with Snowball stemmer was 5702.

5. Stop-word list:

NLTK's stop-words list was imported and used to identify and remove common words like "the" and "a" that hold little semantic meaning.

Obtained list of stopwords:

```
['a', 'about', 'above', 'after', 'again', 'against', 'ain', 'all',
'am', 'an', 'and', 'any', 'are', 'aren', "aren't", 'as', 'at', 'be',
'because', 'been', 'before', 'being', 'below', 'between', 'both',
'but', 'by', 'can', 'couldn', "couldn't", 'd', 'did', 'didn',
"didn't", 'do', 'does', 'doesn', "doesn't", 'doing', 'don', "don't",
'down', 'during', 'each', 'few', 'for', 'from', 'further', 'had',
'hadn', "hadn't", 'has', 'hasn', "hasn't", 'have', 'haven', "haven't",
'having', 'he', 'her', 'here', 'hers', 'herself', 'him', 'himself',
'his', 'how', 'i', 'if', 'in', 'into', 'is', 'isn', "isn't", 'it',
"it's", 'its', 'itself', 'just', 'll', 'm', 'ma', 'me', 'mightn',
"mightn't", 'more', 'most', 'mustn', "mustn't", 'my', 'myself',
'needn', "needn't", 'no', 'nor', 'not', 'now', 'o', 'of', 'off', 'on',
'once', 'only', 'or', 'other', 'our', 'ours', 'ourselves', 'out',
'over', 'own', 're', 's', 'same', 'shan', "shan't", 'she', "she's",
'should', "should've", 'shouldn', "shouldn't", 'so', 'some', 'such',
't', 'than', 'that', "that'll", 'the', 'their', 'theirs', 'them',
'themselves', 'then', 'there', 'these', 'they', 'this', 'those',
'through', 'to', 'too', 'under', 'until', 'up', 've', 'very', 'was',
'wasn', "wasn't", 'we', 'were', 'weren', "weren't", 'what', 'when',
'where', 'which', 'while', 'who', 'whom', 'why', 'will', 'with',
'won', "won't", 'wouldn', "wouldn't", 'y', 'you', "you'd", "you'll",
"you're", "you've", 'your', 'yours', 'yourself', 'yourselves']
```

6. Stop-words removal:

In this section the stopwords listed in section 5 are removed from the text. Removing all those common words with little semantic meaning in the text

7. Vocabulary size display:

The vocabulary size after stop-word removal without stemming was: 9187.
The vocabulary size after stopwords with Snowball stemmer was: 5669.

8. Custom Stop-words:

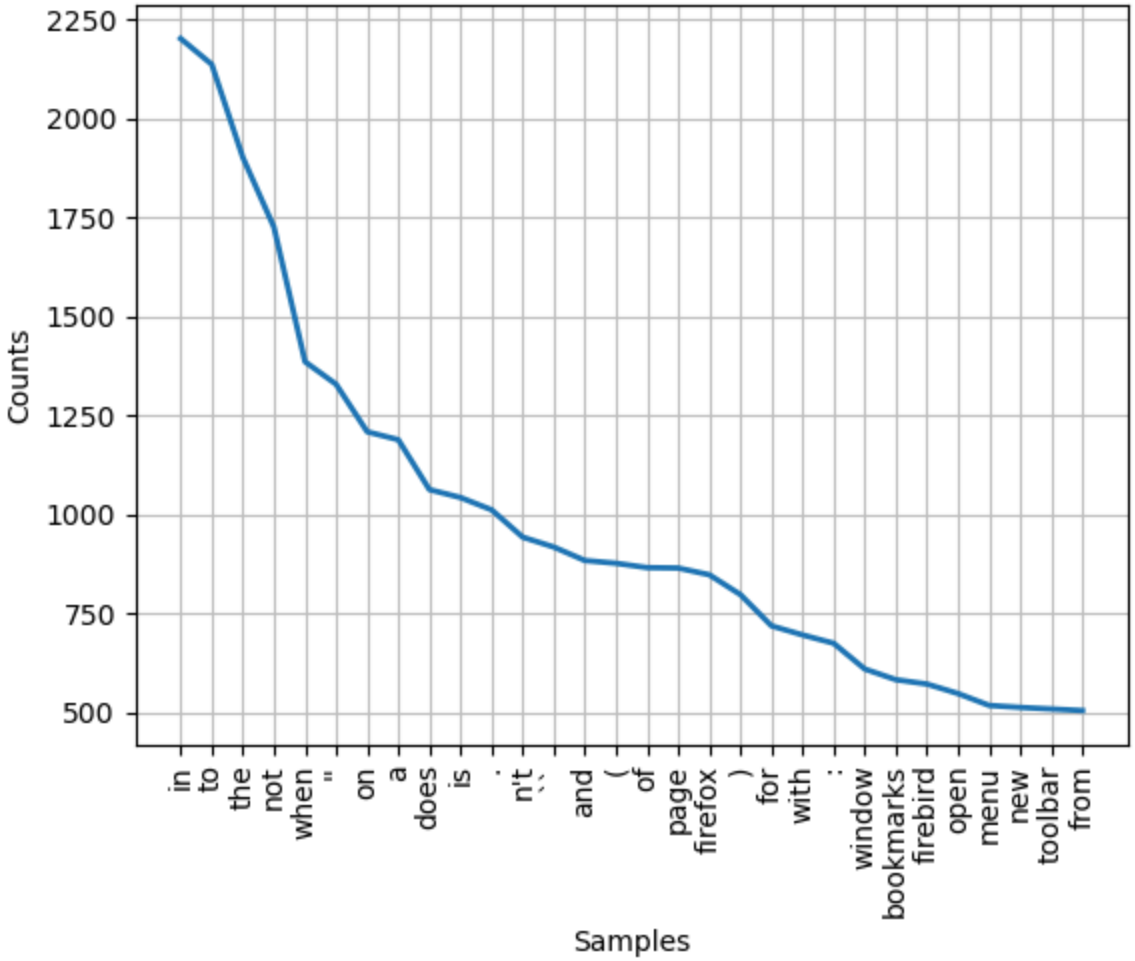
The concept of incorporating user-defined stop-words into the filtering process was introduced, along with an example demonstrating its application.
Added stopwords: 'example', 'with', 'is'
Example:
Original text: This is an example sentence with random words.
Filtered text: sentence random words .

9. Frequency Distribution and Wordcloud:

The frequency distribution of words in the corpus was visualized using a line chart highlighting the most frequently occurring words:
Top most used words with stop words and the number of times appearing :
'in': 2203, 'to': 2138, 'the': 1903, 'not': 1727, 'when': 1387, """: 1330, 'on': 1210, 'a': 1190, 'does': 1064, 'is': 1044,

Top most used words without stop words and the number of times appearing :
"""": 1330, '.': 1013, "n't": 944, '``': 919, '(': 878, 'page': 866, 'firefox': 849, ')': 799, ':': 676, 'window': 611,

Frequency distribution plot with stopwords


2250
2000
1750
1500
1250
1000
750
500
Counts
in
to
the
not
when
"
on
a
does
is
.
n't
`
and
(
of
page
firefox
)
for
with
:
window
bookmarks
firebird
open
menu
new
toolbar
from
Samples

A word cloud was generated to depict the word frequencies visually, with larger words representing higher frequencies.

Exercise 2: Part-of-Speech Tagging

1. Data Preparation:

The Treebank Corpus was downloaded and accessed, providing sentences with their corresponding part-of-speech tags.

2. Feature Extraction:

A function named ufeatures was defined to extract various features from each word in a sentence, including its position, capitalization, prefixes, suffixes, and neighboring words.
These features were used to create a comprehensive representation of each word's context.

3. Ground-Truth Labels:

The part-of-speech tags associated with each word in the corpus served as the ground truth labels for the classification task.
Example:
The first 10 words are:
['Pierre', 'Vinken', ',', '61', 'years', 'old', ',', 'will', 'join', 'the']
The first 10 part-of-speech tags are:
['NNP', 'NNP', ',', 'CD', 'NNS', 'JJ', ',', 'MD', 'VB', 'DT']

4. Data Splitting:

The extracted features and labels were split into training and testing sets using train_test_split from scikit-learn to train and evaluate a machine learning model.

5. Model Training:

A decision tree classifier was chosen and trained on the training data, learning to predict the part-of-speech tag of a word based on its extracted features.

6. Prediction of tags for the test subset:

The trained model was used to predict the part-of-speech tags for the words in the testing set.

```
word: ‘Fred’, Actual POS: 'NNP', Predicted POS: 'NNP'
word: ‘Minera’, Actual POS: 'NNP', Predicted POS: 'NNP'
word: ‘$’, Actual POS: '$', Predicted POS: '$'
word: ‘is’, Actual POS: 'VBZ', Predicted POS: 'VBZ'
word: ‘Walt’, Actual POS: 'NNP', Predicted POS: 'NNP'
word: ‘out’, Actual POS: 'IN', Predicted POS: 'IN'
word: ‘.’, Actual POS: '.', Predicted POS: '.'
word: ‘0’, Actual POS: '-NONE-', Predicted POS: '-NONE-'
word: ‘It’, Actual POS: 'PRP', Predicted POS: 'PRP'
word: ‘-1’, Actual POS: '-NONE-', Predicted POS: '-NONE-'
```

7. Calculate the performance of the POS tagger:

The accuracy of the predictions on the testing set was 94%.
The F1 score of the predictions of the testing ser was 93%.

8. Top Feature of the model:

The top 10 most important features are: ['prefix1=*', 'prefix3=,', 'capitalized', 'suffix1=s', 'suffix2=.', 'word=the', 'word=to', 'suffix3=a', 'word=of', 'word=and']

Conclusions:

This report successfully documented the exploration of text processing and part-of-speech tagging techniques in LAB3. The exercises provided valuable hands-on experience with NLTK libraries and machine learning algorithms, demonstrating their capabilities in manipulating and analyzing textual data.