Python for Computer Science and Data Science 2 (CSE 3652)

Minor Assignment - 3: Natural Language Processing

1. Define Natural Language Processing (NLP). Provide three real-world applications of NLP and explain how they impact society.

2. Explain the following terms and their significance in NLP:
   - Tokenization
   - Stemming
   - Lemmatization

3. What is Part-of-Speech (POS) tagging? Discuss its importance with an example.

4. Create a TextBlob named exercise_blob containing ”This is a TextBlob”

5. Write a Python script to perform the following tasks on the given text:
   - Tokenize the text into words and sentences.
   - Perform stemming and lemmatization using NLTK or SpaCy.
   - Remove stop words from the text.
   - Sample Text:
     ”Natural Language Processing enables machines to understand and process human languages. It is a fascinating field with numerous applications, such as chatbots and language translation.”

6. Web Scraping with the Requests and Beautiful Soup Libraries:
   - Use the **requests** library to download the *www.python.org* home page’s content.
   - Use the **Beautiful Soup library** to extract only the text from the page.
   - Eliminate the stop words in the resulting text, then use the **wordcloud** module to create a word cloud based on the text.

7. **(Tokenizing Text and Noun Phrases)** Using the text from above problem, create a TextBlob, then tokenize it into Sentences and Words, and extract its noun phrases.

8. **(Sentiment of a News Article)** Using the techniques in problem no. 5, download a web page for a current news article and create a TextBlob. Display the sentiment for the entire TextBlob and for each Sentence.

9. **(Sentiment of a News Article with the NaiveBayesAnalyzer)** Repeat the previous exercise but use the NaiveBayesAnalyzer for sentiment analysis.

10. **(Spell Check a Project Gutenberg Book)** Download a Project Gutenberg book and create a TextBlob. Tokenize the TextBlob into Words and determine whether any are misspelled. If so, display the possible corrections.

11.
    - Write a Python program that takes user input in English and translates it to French, Spanish, and German using TextBlob.
    - Create a program that takes multiple user-inputted sentences, analyzes polarity and subjectivity, and categorizes them as objective/subjective and positive/negative/neutral.

- Develop a function that takes a paragraph, splits it into sentences, and calculates the sentiment score for each sentence individually.
- Write a program that takes a sentence as input and prints each word along with its POS tag using TextBlob.
- Create a function that takes a user-inputted word, checks its spelling using TextBlob, and suggests top 3 closest words if a mistake is found.
- Build a Python script that extracts all adjectives from a given paragraph and prints them in order of occurrence.
- Write a program that takes a news article as input and extracts the top 5 most common noun phrases as keywords.
- Write a program that takes a news article as input and extracts the top 5 most common noun phrases as keywords.
- Write a program that summarizes a given paragraph by keeping only the most informative sentences, based on noun phrase frequency.

12. Write a Python program that takes a word as input and returns:
    - Its definition
    - Its synonyms
    - Its antonyms(if available)

13. - Write a Python program that reads a .txt file, processes the text, and generates a word cloud visualization.
    - Create a word cloud in the shape of an object (e.g., a heart, star) using WordCloud and a mask image.

14. **(Textatistic: Readability of News Articles)** Using the above techniques, download from several news sites current news articles on the same topic. Perform readability assessments on them to determine which sites are the most readable. For each article, calculate the average number of words per sentence, the average number of characters per word and the average number of syllables per word.

15. **(spaCy: Named Entity Recognition)** Using the above techniques, download a current news article, then use the spaCy library's named entity recognition capabilities to display the named entities (people, places, organizations, etc.) in the article.

16. **(spaCy: Shakespeare Similarity Detection)** Using the spaCy techniques, download a Shakespeare comedy from Project Gutenberg and compare it for similarity with Romeo and Juliet.

17. **(textblob.utils Utility Functions)** Use **strip_punc** and **lowerstrip** functions of TextBlob's **textblob.utils** module with **all=True** keyword argument to remove punctuation and to get a string in all lowercase letters with whitespace and punctuation removed. Experiment with each function on Romeo and Juliet.