Experiment 1: Preprocessing of Text (Tokenization, Filtration, Script validation, Stop Word Removal, Stemming)

Theory:

1. Tokenization:

Tokenization is the process of splitting text into individual words or tokens. This is the first step in many NLP tasks. Tokens are the basic units for analysis and are typically separated by spaces. For example, the sentence "Natural language processing is fascinating!" can be tokenized into: ["Natural", "language", "processing", "is", "fascinating!"].

2. Filtration:

Filtration involves removing special characters, punctuation, and other unwanted symbols from text data. This helps clean the text and makes it more suitable for analysis. For instance, the sentence "Hello, world!" after filtration becomes "Hello world".

3. Script Validation:

Script validation is important when working with multilingual text data. It ensures that the text is in the expected script or language. For example, it can detect and handle text in languages like English, Chinese, or Arabic correctly.

4. Stop Word Removal:

Stop words are common words such as "the," "and," "is," and "in" that are often removed from text data during preprocessing. These words don't provide much information for NLP tasks and can introduce noise into the analysis.

5. Stemming:

Stemming is the process of reducing words to their root form. For example, the word "running" would be stemmed to "run." This helps in text normalization and reduces the dimensionality of the data.

Example:

1. Tokenization:

Tokenizing the sentence yields: ["The", "challenges", "of", "natural", "language", "processing", "are", "vast", "and", "fascinating!"]

2. Filtration:

After filtration, we remove the exclamation mark: ["The", "challenges", "of", "natural", "language", "processing", "are", "vast", "and", "fascinating"]

3. Script Validation:

In this example, we don't have to perform script validation as the text is in English.

4. Stop Word Removal:

Removing common stop words, we get: ["challenges", "natural", "language", "processing", "vast", "fascinating"]

5. Stemming:

Stemming the words, we obtain: ["challeng", "natur", "languag", "process", "vast", "fascin"]

Algorithm:

1. Import Libraries:

- Import the necessary Python libraries and modules for text preprocessing, including NLTK.

2. Tokenization:

- Tokenize the input text into words and sentences using NLTK's word_tokenize and sent_tokenize functions.
- Print the word and sentence tokens.

3. Stop Word Removal:

- Download the list of English stop words using nltk.download("stopwords").
- Remove stop words from the word tokens using NLTK's stop words list.
- Print the filtered word tokens.

4. Stemming:

- Use the Porter Stemmer to perform stemming on a list of example words.
- Print the original words and their stems.

5. Tokenization, Stop Word Removal, and Stemming on Movie Reviews:

- Download the movie reviews dataset using nltk.download("movie_reviews").
- Get positive and negative movie reviews from the dataset.
- Combine and shuffle the reviews.
- Tokenize, remove stop words, and perform stemming on a sample of reviews.
- Print the results for each review.

Conclusions:

- Text preprocessing is a vital step in NLP tasks.
- It aids in cleaning and normalizing text data for analysis.
- Techniques like tokenization, filtration, script validation, stop word removal, and stemming are crucial in NLP.
- Mastery of these techniques enhances one's ability to handle and analyze text data in NLP applications.