Assignment 1

Problem Statement:

Write a program for pre-processing of a text document such as stop word removal, stemming.

Objective:

To understand the concepts of information retrieval and web mining

Theory:

Text data derived from natural language is unstructured and noisy. Text preprocessing involves transforming text into a clean and consistent format that can then be fed into a model for further analysis and learning.

Text preprocessing techniques may be general so that they are applicable to many types of applications, or they can be specialized for a specific task. For example, the methods for processing scientific documents with equations and other mathematical symbols can be quite different from those for dealing with user comments on social media.

However, some steps, such as sentence segmentation, tokenization, spelling corrections, and stemming, are common to both.

Here's what you need to know about text preprocessing to improve your natural language processing (NLP).

The NLP Preprocessing Pipeline

A natural language processing system for textual data reads, processes, analyzes, and interprets text. As a first step, the system preprocesses the text into a more structured format using several different stages. The output from one stage becomes an input for the next—hence the name "preprocessing pipeline."

An NLP pipeline for document classification might include steps such as sentence segmentation, word tokenization, lowercasing, stemming or lemmatization, stop word removal, and spelling correction. Some or all of these commonly used text preprocessing stages are used in typical NLP systems, although the order can vary depending on the application.

Segmentation

Segmentation involves breaking up text into corresponding sentences. While this may seem like a trivial task, it has a few challenges. For example, in the English language, a period normally indicates the end of a sentence, but many abbreviations, including "Inc.," "Calif.," "Mr.," and "Ms.," and all fractional numbers contain periods and introduce uncertainty unless the end-of-sentence rules accommodate those exceptions.

Tokenization

The tokenization stage involves converting a sentence into a stream of words, also called “tokens.” Tokens are the basic building blocks upon which analysis and other methods are built.

Many NLP toolkits allow users to input multiple criteria based on which word boundaries are determined. For example, you can use a whitespace or punctuation to determine if one word has ended and the next one has started. Again, in some instances, these rules might fail. For example, *don’t*, *it’s*, etc. are words themselves that contain punctuation marks and have to be dealt with separately.

Change Case

Changing the case involves converting all text to lowercase or uppercase so that all word strings follow a consistent format. Lowercasing is the more frequent choice in NLP software.

Spell Correction

Many NLP applications include a step to correct the spelling of all words in the text.

Stop-Words Removal

"Stop words" are frequently occurring words used to construct sentences. In the English language, stop words include *is*, *the*, *are*, *of*, *in,* and *and*. For some NLP applications, such as document categorization, sentiment analysis, and spam filtering, these words are redundant, and so are removed at the preprocessing stage.

Stemming

The term *word stem* is borrowed from linguistics and used to refer to the base or root form of a word. For example, *learn* is a base word for its variants such as *learn, learns, learning,* and *learned.*

Stemming is the process of converting all words to their base form, or stem. Normally, a lookup table is used to find the word and its corresponding stem. Many search engines apply stemming for retrieving documents that match user queries. Stemming is also used at the preprocessing stage for applications such as emotion identification and text classification.

Lemmatization

Lemmatization is a more advanced form of stemming and involves converting all words to their corresponding root form, called “lemma.” While stemming reduces all words to their stem via a lookup table, it does not employ any knowledge of the parts of speech or the context of the word. This means stemming can’t distinguish which meaning of the word *right* is intended in the sentences “Please turn right at the next light” and “She is always right.”

The stemmer would stem *right* to *right* in both sentences; the lemmatizer would treat *right* differently based upon its usage in the two phrases.

A lemmatizer also converts different word forms or inflections to a standard form. For example, it would convert *less* to *little*, *wrote* to *write*, *slept* to *sleep*, etc.

A lemmatizer works with more rules of the language and contextual information than does a stemmer. It also relies on a dictionary to look up matching words. Because of that, it requires more processing power and time than a stemmer to generate output. For these reasons, some NLP applications only use a stemmer and not a lemmatizer.

Text Normalization

Text normalization is the preprocessing stage that converts text to a canonical representation. A common application is the processing of social media posts, where input text is shortened or words are spelled in different ways. For example, *hello* might be written as *hellooo* or *something* might appear as *smth*, and different people might choose to write *real time, real-time,* or *realtime.* Text normalization cleans the text and ideally replaces all words with their corresponding canonical representation. In the last example, all three forms would be converted to *realtime*. Many text normalization stages also replace emojis in text with a corresponding word. For example, *:-)* is replaced by *happy face*.

Parts of Speech Tagging

One of the more advanced text preprocessing techniques is parts of speech (POS) tagging. This step augments the input text with additional information about the sentence's grammatical structure. Each word is, therefore, inserted into one of the predefined categories such as a noun, verb, adjective, etc. This step is also sometimes referred to as grammatical tagging.

Conclusion:

By using above steps, we have performed pre-processing of a text document such as stop word removal, stemming successfully.

Oral Questions

1. What are the different NLTK libraries?
2. How to remove stop words from the file?
3. What is mean by stemming?
4. What is mean by Lemmatization?