AIWR - Assignment 1

Team Members - 099_222_717_729
Ashrita B Kumar - PES1UG20CS099
Kshitij Saha - PES1UG20CS222
Shal Ritvik Sinha - PES1UG20CS717
Gagan - PES1UG20CS729

Demonstration and Analysis

- Justification for preprocessing techniques used Tokenisation: Tokenization is the process of breaking up a text into smaller units, called tokens, which are typically words or phrases.
- It is used to prepare text for analysis, such as in natural language processing, machine learning, and data mining.
- Tokenization helps to standardize and organize text data, making it easier to work with and extract meaningful information.
- Performing tokenization is important for creating a search engine because it helps to break up the text into individual units, such as words or phrases, which can be indexed and searched more efficiently.
- By tokenizing the text, the search engine can quickly identify and match relevant keywords or phrases from the search query to the corresponding tokens in the indexed text, resulting in more accurate and relevant search results.
- Stopword Removal: Stopword removal is important for creating a search engine because it eliminates common, non-informative words like "and" and "the" from the text, allowing the search engine to focus on more meaningful and relevant terms.
- This can improve the accuracy and efficiency of the search engine, as it reduces the number of irrelevant results returned and helps to prioritize the most important words in the query.
- Casefolding: Casefolding is essential for creating a search engine because it normalizes text by converting all letters to lowercase or uppercase.
- This reduces the complexity of the search process and ensures that searches for the same word in different cases return the same results.
- This can improve the accuracy and consistency of the search engine, making it more user-friendly and easier to use.
- Lemmatization: Lemmatization is important for creating a search engine because it reduces words to their base or dictionary form, known as a lemma.
- This helps to group together different forms of the same word, such as "run" and "running", improving the accuracy and relevance of the search results.
- It also reduces the dimensionality of the search space, making the search process faster and more efficient.
- Stemming: Stemming is important for creating a search engine because it reduces words to their root or stem form, which helps to group together different variations of the same word.

- This can improve the recall of the search engine by ensuring that all variations of a word are included in the search results.
- It also reduces the complexity of the search space, making the search process faster and more efficient.
- Justification for choosing an appropriate data structure There are several data structures that can be used when creating a search engine, the one we are using here is the Inverted index: This is the most common data structure used in search engines.
- It is a mapping of terms to the documents that contain them.
- Each term in the index is associated with a list of documents in which it appears.
  How does it work?
- The code you provided is a Python function that takes a list of tokenized tweets as input and returns an inverted index as output.
- The function works by iterating over each document (tweet) in the corpus and then iterating over each word in the document.
- For each word, the function checks whether it already exists in the inverted index dictionary.
- If the word is not yet in the dictionary, it adds an empty list for the word.
- Then, the function appends the document index (tweet index) to the list associated with the word in the inverted index dictionary.
- Once all the tweets have been processed, the function goes through the inverted index dictionary and removes any duplicate document indexes that may have been added during the pre-processing stage.