AIWR

Assignment - 1

Implement a search engine by choosing a relevant corpus.

ANALYSIS REPORT

Team Members:

Name	SRN
Sanmat Sanjayakumar Payagoudar	PES1UG20CS385
Shamith V S	PES1UG20CS390
Monisha N	PES1UG21CS820
Sumukha S S	PES1UG21CS833

- Justify pre-processing techniques used

1. **Case folding** involves transforming all of the text's characters to lowercase or uppercase. Thishelps to shorten the vocabulary, minimize using terms more than once, and make word comparisons easier.

2. **Tokenization** of sentences is the process of dividing a text into separate sentences. Without sentence tokenization, the algorithm may treat the entire text as a single unit and miss importantinformation that is contained within individual sentences.

3. **Word tokenization**: This process includes separating the text into tokens, or individual words. This allows the algorithm to handle each word independently.

4. **Stop word removal**: Stop words such as "the," "and," "a," "an," and "in" should be removedfrom sentences. To lessen the dimensionality of the data, these words are frequently deleted because they don't add much sense to a statement.

5. **Stemming**: This involves breaking down of words to their most basic or root form. It involves removing suffixes or prefixes from words. The goal of stemming is to reduce the number of wordforms that must be processed.

6. **Lemmatization:** This method is similar to stemming, but it reduces words to their dictionary form rather than their root form. This is significant because it decreases the volume of word formsthat NLP algorithms must analyse and assures that the words produced are relevant and appropriate for usage in the language.

7. **TF-IDF vectorization**: Used in reranking
It converts the text into a numerical matrix representation, which reflects the importance of eachword in a document and across all documents in the corpus.

These pre-processing methods are used to enhance the text data's quality and get it readyfor more analysis.

- Justification for choosing an appropriate data structure

1. **Dictionary:** Used dictionary for generating inverted index, positional index, Finding out query intention.

Inverted index **-** Each term in the index is a key in the dictionary, and its value is a list of documentIDs where the term occurs.

Positional index - Keys are the terms in the documents, and the values are a list containing the term frequency, and a dictionary where the keys are the document ids and the values are a list ofpositions where the term occurs in that document.

Finding out query intention: efficient access to the counts for each tag, using the tag as the key.

This data structure is chosen because it allows for efficient access to the documentscontaining a particular term, as well as the positions where the term occurs in those documents.

2. **Positional index:** Used in Phrase Queries, Retrieving relevant text.

Along with the document IDs where a term appears, it also keeps track of where in each of those documents the term is located. As a result, search results can be scored and ranked more precisely according to how closely the recovered documents match the search criteria.

3. **List:** Used in Retrieving relevant text using likelihood language model

we can quickly search through all of the n-grams in the text to discover matches for the query.

4. Sparse matrix: Used in relevance feedback and reranking to store the TF-IDF vectors**.**

The input data is a collection of text documents, which typically results in a large number of features or words, resulting in a very high-dimensional feature space. But Sparse Matrix only stores the non-zero elements and their indices, resulting in significant savings in memory and computational resources. Also, it allows efficient storage and computation of the similarity scores.

- Justification for choosing a typical similarity scoring scheme

1. **Cosine similarity**: Used in Retrieving relevant text based on a query.
Even when the documents are of different lengths, it is effective in determining the degree ofsimilarity between two texts or between a document and a query.

2. **likelihood Scoring:** Used to Retrieve relevant text using likelihood language model likelihood

scoring takes into account the probability of the query terms occurring in proximity to each other in the text, which is a relevant factor for information retrieval. Also, it takes into accountthe probability of the query terms occurring in proximity to each other in the text, which is a relevant factor for information retrieval.