DOCUMENTATION

1. *read\_all\_documents*( ) : This function takes no arguments and updates the *all\_document\_vectors* list with term-frequency vectors of each document in the dataset.

**Term Frequency (tft,d**) - of term t in document d is defined as the number of times that t occurs in d.

**Steps for calculating Term Frequency**

1. Open and read all the files

2 (a) Tokenization – Uses pre-defined word\_tokenize function of NLTK package module.

2 (b) Stemming – Uses pre-defined PorterStemmer function of NLTK package module.

**Tokenization –** Cuts character sequence into word tokens.

**Stemming** - We may wish different forms of a root to match

* + Reduce terms to their roots before indexing.
  + authorize, authorization

NLTK is a toolkit that provides various text processing libraries. It is used for tokenization and parse tree visualization.

1. *input\_vector*( query ) : This function takes the user query as input and returns a term-frequency dictionary.
2. *inv\_index\_all\_documents*( ) : This function takes no arguments and creates the inverted index for all documents in the *inv\_index* defaultdict.
3. *tf\_idf\_vectorize*( ) : This function takes no arguments and converts all the term-frequency vectors in *all\_document\_vectors* to TF-IDF vectors.
4. *calculate\_tf\_idf*( query\_vector ) : This function takes the term-frequency vector of the query as input and converts it to a TF-IDF vector.
5. *tf\_idf\_score*( word, frequency ) : This function takes a word and its frequency as inputs and returns its TF-IDF score.

***Score = (1 + log tf) x log (N/df)***

1. *dot\_product*( vector\_a, vector\_b ) : This function takes two unit normalized vectors as input and returns their dot product.
2. *stem\_and\_tokenize*( document\_text ) : This function takes a string of document text matter as input and returns a stemmed and tokenized list of words in the document.
3. *create\_vector*( token\_list ) : This function takes an input list and creates a document frequency dictionary in *document\_frequency* dict.
4. *document\_string*( document\_id ) : This function takes a document’s ID as input and reads it and returns its text content as a string.
5. *query\_result*( query\_vector ) : This function takes unit normalized TF-IDF query vector as input and returns a list of DocumentIDs and their weighted cosine scores in sorted order as output.