Report  
  
1. Imports:

* The required imports are the first thing the code does.
* nltk, pandas, os, torch, re, numpy, tensorflow, transformers To build the GUI, imports modules from tkinter.

2. Data Collection:

* read\_data function, which reads text data from files in archive\_2 folder, by which each folder inside is a category.

# 3. Text Preprocessing:

* preprocess\_text and clean\_text are defined for text preprocessing.
* preprocess\_text function:
  + Eliminates stopwords, The clean\_text function eliminates white spaces, URLs, and punctuation from the text.

# 4. Indexing:

* The index\_documents function produces a DataFrame containing the processed text and document numbers after preprocessing the text data and generating document numbers.
* An inverted index for the documents is created by the build\_inverted\_index function.

5. Retrieval of Information:

* The search\_index function uses the inverted index to look for documents that contain the query phrase.
* The rank\_tfidf function uses the query's TF-IDF score to rank the documents.

6. BERT Model and Tokenizer:

* The transformers library's AutoModel and AutoTokenizer are used to load a BERT model and tokenizer.

7. User Interface (GUI):

* Tkinter is used to create a basic GUI.
* A text field displays the results of queries entered by users.
* Upon clicking the search button, the search function is invoked, retrieving documents and displaying them.

# 8. Functionality:

* Based on a query entered by the user, the system looks for pertinent documents.
* The ranked documents and their ratings are shown by the system.
* Additionally, it expands the query using RM3, displaying the results of the extended query side by side with the original query.
* Furthermore, it uses a pre-trained BERT model to display the embeddings for every term in the query.