Assignment 2

(i) Describe the steps you have performed for data preprocessing.

- 'parse_reuters_first', 'parse_reuters_last', 'parse_article_id','parse_title_first', 'parse_title_last', 'parse_body_first' and 'parse_body_last' variables provide the values necessary to perform parse operations.
- The 'find_str' function returns the starting index or the last index with the control parameter, which is the boolean expression, whether there is a string sent as 'char' in the 's' parameter and the its index.
- The 'get_reuters_index' function finds the start and last index of the 'REUTERS' tag.
- The 'get_reuters_index' function finds the start and last index of the 'NEWID' tag and also finds 'article_id'.
- The 'get_title' function finds the start and last index of the 'TITLE' tag and also finds 'title'.
- The 'get_body' function finds the start and last index of the 'BODY' tag and also finds 'body'.
- The 'removal_utilities' function removes punctuation marks, lowers all characters, and also removes all stopwords from the given parameter.
- Putting all the words in 'stopwords.txt' into a set.
- In order to parse all the articles in the 'reuters21578' folder, the necessary variable is defined.
- It pulls all the files in the 'reuters21578' folder one by one and sends them to the auxiliary functions described above.

(ii) Describe the data structures that you used for representing the inverted index.

- As stated in the project description, I create 'inverted_index.json' and fill them with the 'dict_reuters' data structure.
- This data structure contains ‘string’ as key and list of article_id as value.

(iii) Describe the trie data structure that you used in your code and provide your well-commented trie code in the report.

- 'Node' class contains its children and whether it is the last word, and it does this with two fields, 'children', 'is_end_of_word'

- The 'Trie' class uses the 'Node' class created above as the field and holds the 'search', 'insert' functions and the 'traverse_node' helper function that allows us to traverse all children of a node.
- The 'insert' function allows to insert strings into the 'Trie' class.
- The 'search' function allows to serach strings in the 'Trie' class.
- The 'traverse_node' helper function that allows us to traverse all children of a node.
- As stated in the project description, I create trie.pickle and fill them with the 'trie' class.