Part B
RELEVANCE FEEDBACK

Group No- 14

Group Members	**Roll No.**
Ajay Kumar Meena	18CS30006
Kushal Natani	18CS30025
Yash Fulzele	18CS10058
Mudit Gupta	18CY20019

Task B1 (Relevance Feedback):

In this part, we loaded the Inverted Index built-in Part A-Task 1, and the ranked results for "ltc.lnc" scheme of Part A- Task 2 along with the gold standard file.
Now, preprocess the data and parse the preprocessed queries using the query_14.txt file.
Then precompute the normalization coefficients for all docs.

The following steps were followed to find a modified query vector for each query:

- Consider the top 20 documents in the retrieved ranked list.
  - For RF, consider docs having relevance score of 2 as relevant while others as non-relevant.
  - For PsRF, consider the top 10 docs as relevant while the set of non-relevant docs is empty.
- Compute the TFIDF value for the query terms only and store them in a queryVector dictionary
- Iterate over the relevant and non-relevant docs and find TFIDF value for each term in the doc and add the term in the queryvector dictionary if it is not present already, otherwise modify the value in queryVector.
- Now, iterate over all the docs and find score(cosine similarity) for (doc, query) pair and store (score, docno) in a list.
- Sort the doc score and return the top20 docs for each query.
- Evaluate the NDCG and AP score for each query(using the evaluator in Part A-Task 2) and return the mean values.

Task B2 (Identifying words from pseudo relevant documents closer to query):

- Consider the top 10 documents of the retrieved ranked list as relevant for each query.
- Compute the normalized TF-IDF vectors for each of these documents.
- For each query, obtain a vector of vocabulary size by averaging the TF-IDF vectors of the considered docs
- The value at each location of the obtained vector represents the average TF-IDF score of a specific word in the vocabulary.
- Sort the obtained vector values and print the top 5 terms having the highest score.

Note: For tasks B1, B2 we also need the path of pre-processed query file <path_to_queries_14.txt>as 2nd argument. For more details check running steps.

Python version: Python 3.8.10
Libraries Required:
os, sys, pickle, math, csv, collections

Running Steps:

```
$ python PB_14_rocchio.py <path to the en_BDNews24 folder> <path_to_queries_14.txt>
<path_to_model_queries_14.pth> <path_to_gold_standard_ranked_list.csv>
<path_to_PAT2_14_ranked_list_A.csv>
```

```
$ python PB_14_important_words.py <path to the en_BDNews24 folder>
<path_to_queries_14.txt> <path_to_model_queries_14.pth>
<path_to_PAT2_14_ranked_list_A.csv>
```