Readme

[ME16B125] Durga Sandeep, [ME16B166] Sureddy Abhishek

February 2021

1 Instructions

To install all the requirements, run the below command in the python virtual environment.

```
pip install -r requirements.txt
```

NLP Team 6 Code.ipynb is the main file, where you just need to run all the cells in this notebook to generate the results mentioned in the report. And the other python files are used in this main file.

- **NLP Team 6 Code.ipynb** This notebook contains all the techniques we have tried in this project like Vector Space Model with two versions (VSM-1 and VSM-2), Latent Semantic Analysis (LSA), Query Expansion(QE) + LSA, Clustering techniques to improve retrieval time and hypothesis testings.
- **Query Completion.ipynb**This is the additional technique we tried to help the user in finishing the query.
- **sentenceSegmentation.py** This python file contains, different approaches to convert a document into list of sentences.
  - ‘naive‘: Here splitting the document based on ”pull stop”
  - ’punkt’: Here we used punkt sentence tokenizer.
- **stopwordRemoval.py** This file contains the process of removing stop-words like the, of, and,.. given a document. Here the list of stopwords are imported from nltk package.
- **tokenization.py**This python file contains, different approaches to convert a sentence into list of words.
  - ‘naive‘: Here splitting the sentence based on ”pull stop”, ”comma” and other common delimiters
  - ’pennTreeBank’: Here we used pennTreeBank tokenizer to tokenize the sentence.

- **evaluation.py** This file contains all the evaluation metric functions like queryPrecision, queryRecall, meanPrecision and so on.

- **evaluationAllMetrics.py** This python file will be used for plotting all the evaluation metrics for different k-values.

- **inflectionReduction.py** For stemming or lemmatization. (i.e. to reduce a word to it’s root form).

- **observations.py** This file contains

  - ’run’: plots the distribution of the recall, precision, ndcg, f-score.
  - ’run—comp’: to compare the distributions of recall, precision, ndcg, f-score of any two models.

- **tfidf.py** This file contains

  - ’TF_IDF’: gives tf_idf representation of the documents.

- **util.py** This file contains utility function.

  - ‘build_word_index‘: To build word to index mapping, used in tf-idf.

- **Folders**

  - cranfield: This folder contains the training dataset(cran_docs.json), queries to be evaluated(cran_queries.json) and true order of the documents (cran_qrels.json)

  - output: This folder contains the output files of the assignment 2 i.e segmented documents into sentences, then sentences to tokens, then lemmatization, then stopword removal.