Information Retrieval 2 Report

Experiment 1: Indexing the datasets

a. Implemantion Overview

The Python script communicates with the Apache Solr server through the `pysolr` package. It uses the `pysolr` library's `add` method to send pages to the Solr server for indexing.

We first defined the solr server config to connect to the Apache Solr Admin page. Then we defined a Function `index_documents` to Index the Documents. This function iterates over all the files in the given directory and then indexes the documnets using solr server which uses lucene internally.

Experiment 2: Vector-based Models

a. Implemantion Overview

1. Preprocessing: We first preprocessed the given documents by tokenizing them, converting them to lowercase, removing punctuation, removing stopwords, and stemming the tokens.
2. Calculate the term frequencies: We then used the processed documents and calculated the term frequencies in a collection of processed documents.
3. Calculate the document frequencies: Now after calculating the term frequencies we used the processed documents and calculated the document frequencies in a collection of processed documents.

Example:

```
processed_docs = [

["apple", "banana", "orange"],

["banana", "kiwi", "mango"],

["apple", "kiwi", "orange"]

]

term_frequencies = calculate_term_frequencies(processed_docs)

print(term_frequencies)

# defaultdict(<function__main__.lambda at 0x7ff911f20960>, {

#     'apple': defaultdict(<class 'int'>, {0: 1, 2: 1}),

#     'banana': defaultdict(<class 'int'>, {0: 1, 1: 1}),

#     'orange': defaultdict(<class 'int'>, {0: 1, 2: 1}),

#     'kiwi': defaultdict(<class 'int'>, {1: 1, 2: 1}),

#     'mango': defaultdict(<class 'int'>, {1: 1,})

# })
```