**Project 1 – Group 27**

Parth Mehta (pm2877) Dheeraj Kalmekolan (drk2143)

**List of files submitted**:

bing.py

read\_result.py

stop.txt

transcript.txt

**How to run the code**:

Put the files bing.py, read\_result.py and stop.txt in the same directory and execute the following line:

* python read\_result.py <key> <precision-required> <query>

**Note**: For a query with more than 2 space-separated words, enclose the query in single quotes. For example, ‘milky way’.

**Internal design of the project**:

**Function in file: read\_result.py**

word\_count() – takes HTML source text and returns an Ordered Dictionary of words, i.e, the term frequency for each page.

get\_from\_docs() – It calls word\_count() for each relevant document and returns a combined ordered dictionary of words across all relevant documents.

**Functions in file: bing.py**

execQuery() –Takes query and key parameters and fires a query to Bing using the parameters provided.

**Flow of the program**: The main function of read\_result.py first calls execQuery() from bing.py, which returns the results of the query. Then we call the get\_from\_docs() function, by passing it a list of relevant documents. This function in turn calls word\_count(), by passing it entire string of one HTML page.

**Query modification method**:

In each iteration, we add exactly 2 words: The first word is the word that has the highest frequency in the headings, paragraphs, and titles of the HTML pages that are marked relevant during the relevance feedback in the previous iteration. To find this word, we first count the total occurrences of each word that occurs across the relevant documents. We then remove punctuations, Unicode characters and stop-words from this word list. Any word present in the former query is also deleted. The word that now has the highest frequency across the relevant documents is chosen as the first word to be appended to the original query.

The second word is chosen from the Titles and Descriptions of the results returned by Bing. For achieving this, we look for all words that have their first letter capitalized we did this because intuitively user searches for the nouns and in general nouns are capitalized. We remove stop-words, punctuations and words from the older query. If in case the first and the second words turn out to be the same, the first word is changed before augmenting the query, to avoid duplicate words in the query.

We also consider a case when the content of the relevant documents is not large enough to find a word that is frequent (eq. When you search python for snakes). If the number of words found from the HTML pages is less than 10, then we take words only from the Titles and Descriptions of the results returned by Bing. In this case, we calculate the frequency of each word across all titles and descriptions, and return the words with the highest frequencies after removing stop-words, punctuations and duplicate words.

**Bing search account key**:

yaTtNJf/mq/VOXKXCliOjmUaeoL4hiK4akoPVAjvsdk