### [COMS E6111 Advanced Database Systems](http://www.cs.columbia.edu/~gravano/cs6111/index.html) Fall 2015

**Project 1**

1. **Name and UNI**

Bahul Jain – bkj2111

Parth Parekh – prp2121

1. **A list of all the files that you are submitting**

temp.py – Main Python Script which runs the project.

Account Key.txt – Contains the Bing Search Account Key and Customer ID.

Stopwords.txt – Contains a list of stop words and other symbols we want to eliminate.

Transcripts.txt – Log of the output on the basic 3 cases (gates, musk, taj mahal), with some additional test cases (apple, friends)

1. **A clear description of how to run your program** (note that your project must compile/run under Linux in your CS account)

To run the program, run the temp.py file using the command:

python temp.py <account key> <precision> <query>

For e.g. If we want to search for the animal ‘jaguar’ and we require a precision of 0.9, the command to run the project would be:

python temp.py 2dyKIv94jDETd7ClbVKoHvJSWFJ73ZvZRc7rjpBdkG8 0.9 jaguar

This should be run from the directory which consists the file temp.py

1. **A clear description of the internal design of your project**

The flow of the code is as follows:

1. Getting bing account key, precision and query as input from the user.
2. Storing the stop words in an array which are used to eliminate irrelevant highly frequent occurring words.
3. Connecting to Bing API and getting response of the query passed in the URL.
4. Storing the Titles, Descriptions and URL’s in their corresponding array.
5. Get relevance feedback for all the search results and calculate the new precision.
6. Check for the termination cases, if termination case not reached, continue.
7. Remove the punctuations and add the words from title and description of the result in the dictionary.
8. Design a Vector Space Model for the information retrieved
9. Reformulate the query (only query expansion by augmenting to the original query)
10. Repeat steps 3 through 10
11. **A detailed description of your query-modification method**

Once we have formed the vector space model of the information retrieved we have a vector corresponding to every relevant document, every non-relevant document, and the original query. Using these vectors we implement Rocchio’s algorithm. This algorithm computes a new query vector that is similar to the relevant documents vector and dissimilar from the non-relevant documents vector. The formula for Rocchio’s algorithm is as follows…

This algorithm increases the weight of the words that occur in the relevant documents and decreases the weight of the words that occur in the non-relevant documents in the revised query vector. Alpha, beta and gamma are constants that control the factor by which we are considering contributions from the original query, relevant documents and the non-relevant documents respectively. Since we are more concerned with minimizing the distance of the query vector and relevant documents than maximizing the distance of the query vector and the non-relevant documents we generally keep a high value of beta and a low value of beta. Hence we choose

1. **Bing Search Account Key** - 2dyKIv94jDETd7ClbVKoHvJSWFJ73ZvZRc7rjpBdkG8