**COMS E6111 Project 1**

1. Group name: Project 1 Group 23

Qi Wang (qw2197), Yongjia Huo ()

1. Submitted files:

main.py – the main program, get relevance feedback

parameters.py – parameter settings and parser

query\_processor.py – implement the Rocchio algorithm

stopwords.txt – list of stop words

transcript.txt

readme.pdf

1. Use the following command to run our program:

python main.py <account key> <target> <initial query>

1. Description of design:

Each query round we display the URL, title and description of top-10 results returned from the Bing Search API, and prompt user to mark each page as relevant or not relevant. We parse the text from URL, title and description into a list of words split by punctuations, then we use all terms not in the stop words list to construct the vector space. We use the Rocchio algorithm and tf-idf to calculate term weights, the two highest-weighted terms not presented in last query are added in the new query, terms in new query are ordered by their weights. Each round we calculate the precision@10, if the target value is achieved or the precision is zero, the program terminates.

1. Description of query-modification method:

For each query, we add results marked as relevant into the relevant set, and results marked as nor-relevant to non-relevant set. The URL, title and description are combined into a single string separated by white space. URL is pre-processed by the parseURL() function, we only keep the parts between the last two slashes, because the keywords usually appear here. We also remove “.html” and digits which are basically garbage information. Then we replace all punctuations with white spaces, convert the string into lower characters and split it into a bunch of words. Stop words are removed.

After parsing, we calculate the term frequency of relevant documents, term frequency of non-relevant documents and document frequency respectively. Then we construct the document vector using the Rocchio algorithm. The modified query is:



Where the term weight of document vector is calculated by tf-idf algorithm:



the vector is sorted decreasingly by term weights. Finally, we add weights to terms in old query, the two new terms of highest weights are considered good candidates for query expansion and appended to the old query.

1. Bing Search Account Key:

HIWkFhlcqfV0SsO9ac7smysylCtGDsuMVyqgSWPPDZI

1. References:

* Singhal, Amit. "Modern information retrieval: A brief overview." IEEE Data Eng. Bull. 24.4 (2001): 35-43.
* Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA.