Advance Database Systems
Project 1

Ajay Siva Santosh Reddy Challa (ac3647)
Arun Swaminathan (as4522)

List of files:-
Assignment.py

How to run:-
Install the NLTK and the xml.dom modules, if not already present.
Run the following command:
Python Assignment.py <Bing account key> <precision> <query>

Internal Design:-
The program consists of the following modules:-
query_ask(query) : This module takes the query as an input, runs it through the Bing API and returns the results which are written to a xml file for future parsing.

parse(): This module, reads the xml file to which the query results were stored and parses it to extract the 'title' , 'url' and 'description' of each of the results separately and stores them as a list.

index(): This module builds the index using the parsed data and tf-idf values of each of the words in each of the results is calculated and stored.

rocchio(): This module computes the terms to be added to the query using the rocchio query expansion algorithm, adds them and proceeds to the next iteration.

feedback(): This module collects the relevance feedback from the user.

When the user runs the program, the initial query is passed to the query_ask() method which generates the results and calls the parse() module to parse them. The parse module then separates the results and in turn separated the title, url and description of each of the results. Which are then send to the feedback() method to collect the relevance feedback from the user. Once the user enters the feedback, the precision is checked with the target precision and if it turns out to be less, new words to be added to the query are selected, for which the index() module builds an index of the terms from all the results. This index module generates the tf-idf values of each of the terms in each of the results. These tf-idf values are the passed on to the Rocchio() method which calculates the weights of each of the terms using the Rocchio algorithm for query expansion and then the words with the highest weight are added to the query and a next iteration is called by calling the query_ask() method again.

Query Modification Method:-

This program uses the Rocchio algorithm for query expansion. To compute the terms for the formula to applied, the following steps are carried out.

a) Data Pre-processing

Analyzing data that has not been carefully screened for such problems can produce misleading results, hence for this reason we preprocess our data. This involves performing certain operations on data in order to get it ready for training the classifier. Our Preprocessing involved the following steps:-

1) Reading the training data - The data was provided in the form of a.xml file. It had to be read first in order to perform the rest of the preprocessing steps.

2) Extracting relevant text from each of the file - Relevant content from the .xml file had to be extracted as separate entities, so that they can be labeled as such. In this case the title, url and description of each result are extracted.

3) Removal of Punctuations – Punctuations which may act as a hindrance to the accurate processing of text are removed.

4) Removal of stop words - Stop words are those trivial words which do not contribute much to the purpose of training, hence they are removed from the list of tokens. The stop-words to be removed were taken from the nltk corpus.

4) Segregating results into relevant and non-relevant based on the user feedback – The results are labeled and stored in in the list format as relevant and not-relevant.

b) Building an Index

To proceed with the rest of the task, an index needed to be built comprising of the words from the results. For this process, we perform the following steps.

1) Tokenization – The results were split into tokens

2) Indexing – Each of the tokens are stored as a dictionary key with the value being a list of 10 values which indicate the frequency of the token in of the 10 results

c) Computing the TF-IDF values

This step involves using the indexed data to calculate the TF-IDF values of each term for each result.

TF is the frequency of a term t in a document divided by the total number of terms in the document.

IDF is the log(Total no. of documents / No. of documents with term t in it)

IF-IDF is TF*IDF

The TF-IDF values of each term per each result are stored.

d) Implementing Rocchio Algorithm

Rocchio algorithm is used to compute the terms that need to added to the query to get a better precision. It is computed using the formula (expressed in simple terms)

Weight of a term =

$$\frac{b * \sum \text{TF-IDF values of term t in relevant documents}}{Total\ number\ of\ relevant\ documents} - \frac{c * \sum TF-IDF\ Values\ of\ term\ t\ in\ non-relevant\ documents}{Total\ number\ of\ non-relevant\ documents}$$

Where b and c are constants.

Once the weight is computed, the terms with are sorted according to the weights and the top two terms are selected and added to the query. Then the process is repeated again till required precision is achieved.

Bing Account Key:-
pZyOYhNovB61fTsh3yf6QW6sVCbldiH2QLctVAk2zyw