1. Our Name and UNI
   Andrew Shu ans2120@columbia.edu
   Ran Bi rb2651@columbia.edu
2. Files that we submit
   a. Jave code files
   b. Makefile
   c. Readme
   d. transcript_tiger.txt, transcript_trainer.txt, transcript_rock.txt
3. How to run our program
   a. Unpack the .tar.gz archive.
   b. Build the program using 'make'
   c. Run the program using 'make exec'
      i. The program will launch the GUI version by default, which requires a graphical windowing environment. If running on CLIC machines remotely, it will be necessary to use a VNC session. When starting the vncserver, use command '`vncserver -geometry 1024x768`'. From the xterm in this VNC session, run 'make exec' to launch the GUI.
4. Internal design of our project
   a. Set up a way to talk to Yahoo! Search BOSS API through Java.
   b. After the user input the query, there will be a Resultset returned from Yahoo! BOSS. If the specific value of precision is reached, we will stop. Otherwise, we do the query expansion.
   c. Return the new Resultset to the user.

5. Query-Modification Method
   a. According to the feedback of the user, we can divide the original Resultset into two parts, relevant and non-relevant Resultsets.
   b. Through Apache Lucene, we can find all the terms which appear in the Resultset. Then for a given term, we compute its frequency, which is the number of documents in which this term appears. We do this computing for both relevant and non-relevant Resultsets.
   c. Set the frequencies of terms in non-relevant Resultset to be negative. Then we combine the terms in two Resultsets. If the terms are the same in two Resultsets, we add their frequency (i.e., subtract non-relevant frequency) and return only one term vector, which consists of term name and new frequency.
   d. We sort all the terms by their frequencies and pick up top two terms.
   e. Add these two terms into the old query. To put them in the best order, we compute the positions for all terms present in the new query. Then we select the best document which contains the most of these terms.
   f. We put the terms in the same order as in that best document. Any leftover terms we append to the end. Finally we use the new query with Yahoo! BOSS.

6. Yahoo! BOSS Application ID
   8ZdPHgrV34GqaKRLc2FEMULwfYT9_rn1xE0swnm6JcfB_IflTBEdzba7HAuPJcDwGA--