## FIT5166 Information

## Retrieval Systems Practical Assignment - Semester 2 2014Information Retrieval System

### Manual

1. Index

**java popcorn index [file-collection-path] [index-to-directory-path] [stopwords.txt-path]**

e.g.

java popcorn index ../test/ ./indexdir/ ./stopwords.txt

With index command, system will go through file collection directory and index all terms from files and save the indexed terms in a file called index.txt under index-to-directory-path directory.

In index.txt file, terms will be listed line by line. Each line starts with the term, then all documents that contains the term and the *sublinear term frequency*\* ( ) which used to range the value of tf and the last value is the idf which is calculated from ( ). All units are separated by comma

**term,doc-1,1.2,doc-2,1.3,doc-3,1,0.871**

e.g.

imposit,chapter4-FinanceAndTrade.txt,3.485,chapter1-TheParliament.txt,1.693,2.398

1. Search

**java popcorn search [indexed-path] [number-of-top-K-related-document] <[keyword-list]>**

e.g.

java search ./indexdir/ 10 what do cats eat

Search function will read the indexed file and return top K documents with ranking score back to screen.

When retrieving, system will build a *candidate documents set\** to enhance the search performance with some heuristics like set a *document champion* *list\** for each term, *sort the index.txt file by idf descent order\** and only *considering a significant proportion of terms containing\*.*

**Note: Only after indexing, with an existing index.txt, the search function works.**

1. Set arguments

**java set [percentage of documents have the term] [proportion of queries the document has] [use implicit feedback or not] [use champion list or not]**

e.g.

java set 80 20 true false

This function is used to set the search parameters which will determine the system performance.

1st argument *Percentage of documents have the term (tp)* is an integer number which should be scaled ( 0 , 100 ]. It determines the minimum idf ( ). If the term has a lower idf value, it will not be retrieved.

2nd argument *proportion of queries the document has* (dp) is an integer number which should be scaled [ 0 , 100 ). It determines the minimum proportion the document contains the query terms. If a documents contains less then dp% of the terms, it will not be considered.

3rd argument *use implicit feedback or not* is a boolean value. Set true will calculate the first round top N ranked documents with Rocchio Algorithm to get an implicit feedback which will get a new list of ranked documents.

4th argument *use champion list or not* is a Boolean value. Set true will only considering a champion document for each term. The document which has never been a champion will not be retrieved.

Configuration file

Under the program project src folder, a config file maintains the parameters set.

### Features

1. sublinear term frequency (w)

to range the term frequency which could be effect the weight of the term if huge.

1. a candidate documents set
   1. document champion list

Indexed a term frequency champion documents list to produce the candidate ranking document.

* 1. index.txt file ordered by idf

if a term with a small idf value which means it is a common term for most of the documents, it will be ignored.

* 1. only considering documents which contains a higher proportion of the query

Assume the query have 5 keywords, for instance, the documents which only contains 1 or 0 keywords will not be ranked and retrieved.

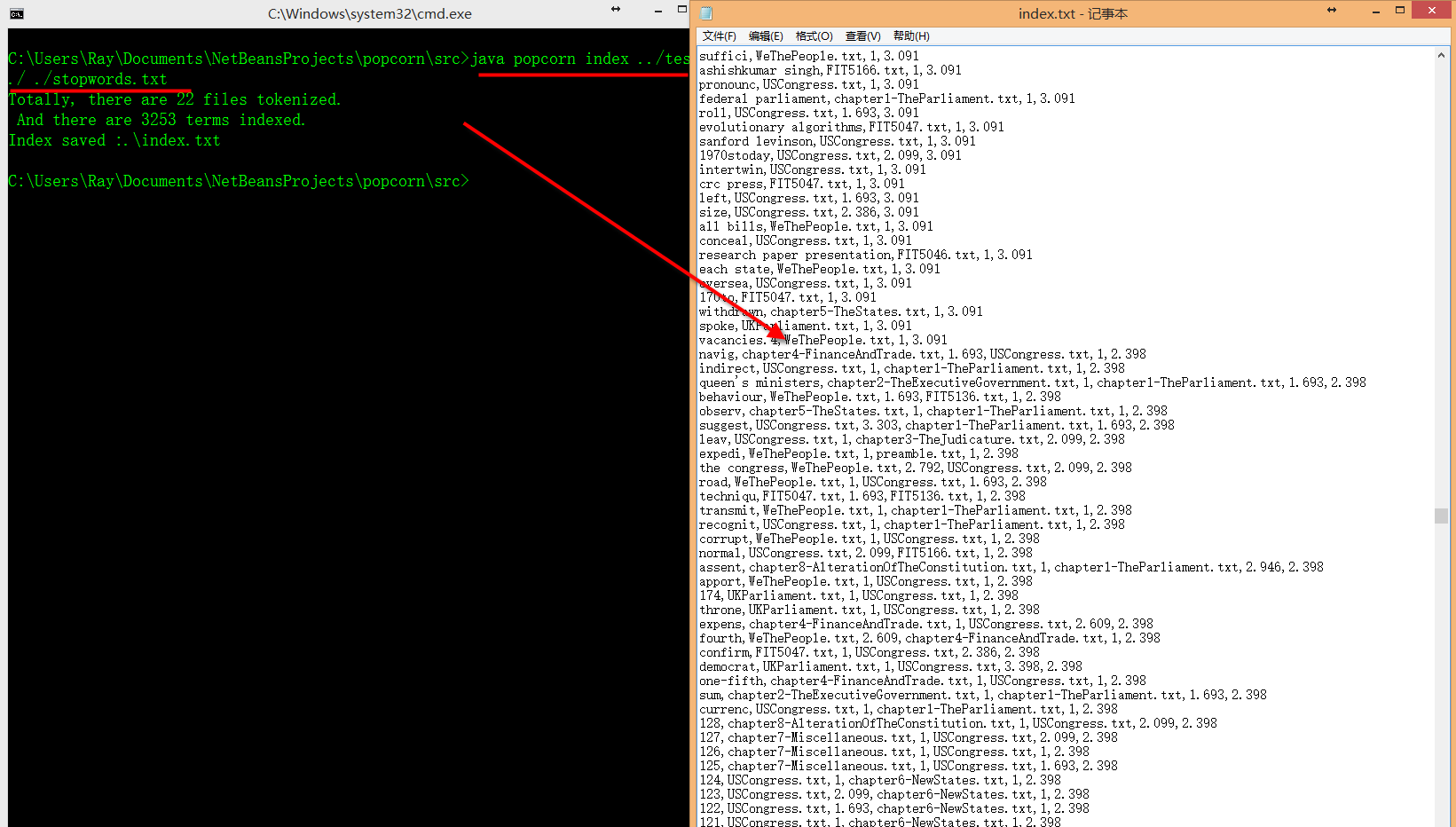
1. Implicit Feedback

Using Rocchio Algorithm and return top K relevant documents to re-calculate the query Q’

System will assume .

### Testing

1. Index



1. Search
   1. With / without implicit feedback

Test Collection: docs1

Steps:

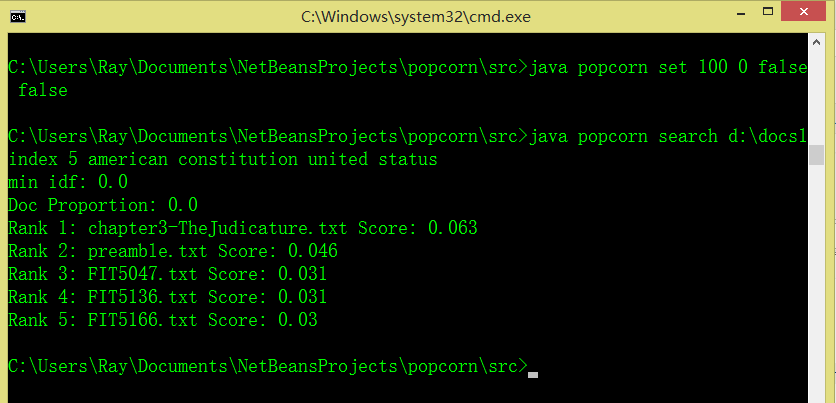
1. Set the implicit feedback switch to false with following command

java popcorn set 100 0 *false* false

1. Search with following statement

java popcorn search [the indexing direction ] 5 united status constitution American

1. Search result as below



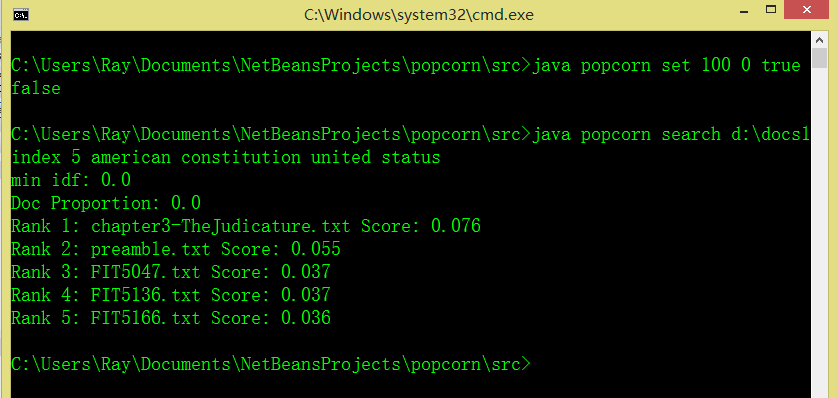
1. Then set the implicit feedback switch to true with following command

java popcorn set 100 0 *true* false

1. Searching with the same statement

java popcorn search [the indexing direction ] 5 united status constitution American

1. Search result show as below



Result: we can see the Score are different between the 2 search result before and after swith on the implicit feedback switch which will re-calculating the result with top K treated as ralevent documents using Rocchio Algorithm but not affect the ranking because in this system the tf is sublineared (limited range in a small scale) and only keep 3 digitals after the decimal point which means the re-calculated number will not have a big change which means the result is not obvious in some cases.

* 1. Compare the rank list with / without champion list

Test Collection: docs1

Steps:

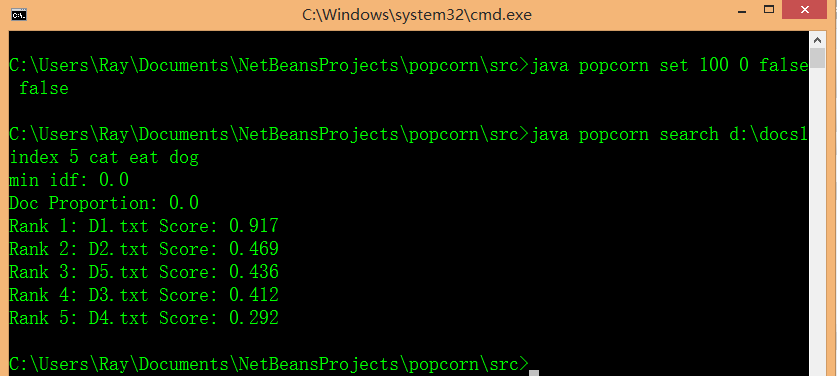
1. Set the Champion List switch off with following command

java popcorn set 100 0 false *false*

1. Search with following statement

java popcorn search [the indexing direction ] 5 cat eat dog

1. Check the result we can see 5 documents retrieved which contains the key words cat, eat and dog



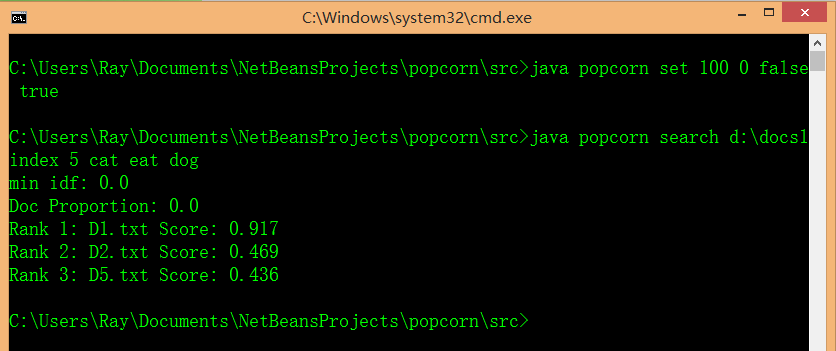
1. Then we turn the Champion List switch on

java popcorn set 100 0 false *true*

1. Search the same key words

java popcorn search [the indexing direction ] 5 cat eat dog

1. Observe the result



Result: We can see D3.txt and D4.txt are not ranked after we turning the Champion List switch On. Which means D3.txt and D4.txt are not the champions for any term so it will not be added to the candidate set.

* 1. check the setting of minimum idf feature

Test Collection: docs3

Steps:

1. Set the first argument after set to 100 (%)

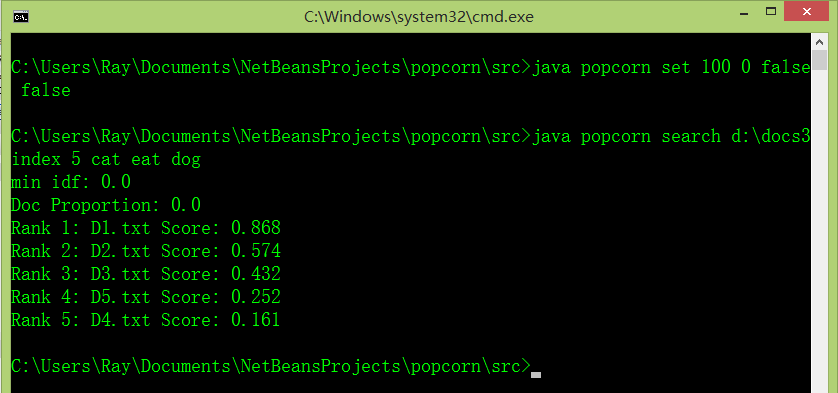
java popcorn set 100 0 false false

which means even a terms appears in all documents, the dfmax = N and idfmin = 0, it will be considered.

1. Search with following statement

java popcorn search [the indexing direction ] 5 cat eat dog

1. Check the search result



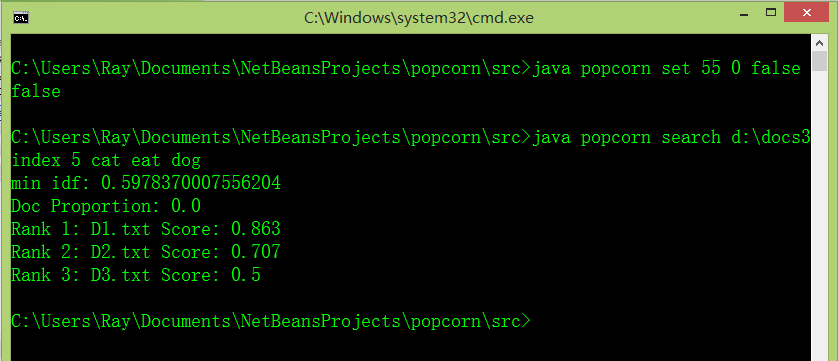
1. Then we set the percentage to 55% idfmin = 0.598

java popcorn set 55 0 false false

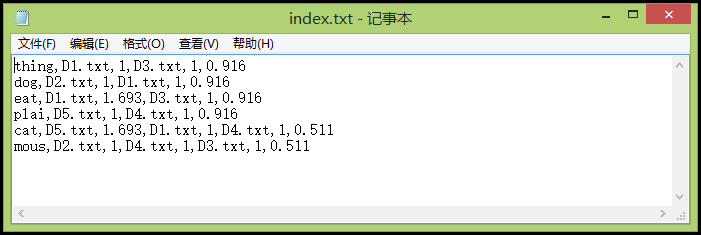
1. Search again

java popcorn search [the indexing direction ] 5 cat eat dog

1. The search ranking result



Result: Checking the index.txt file



We can see the key word cat has an idf 0.511 which is smaller than the idfmin. The D4.txt and D5.txt which are not have term dog and eat are not retrieved as expecting.

* 1. Demonstrate the percentage of keywords containing heuristic

Test Collect: dosc3

Steps

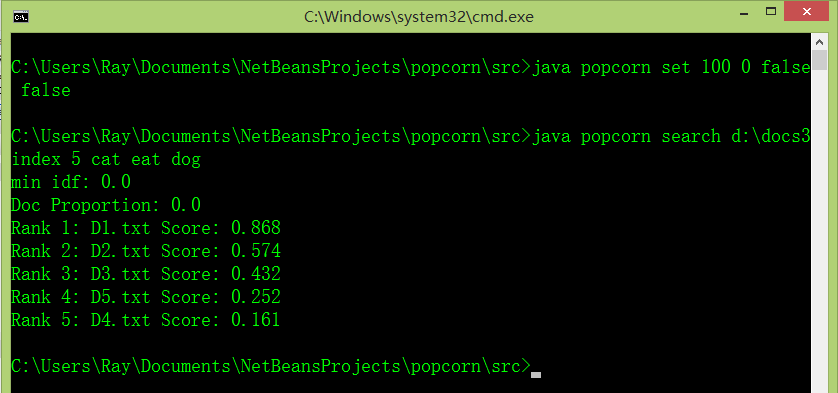
1. Set the percentage to 0 which means if the documents contains 0 terms, it will also be treated into candidate set (But obviously, it will not be retrieved)

java set 100 0 false false

1. Search with key words dog eat cat

java popcorn search [the indexing direction ] 5 cat eat dog

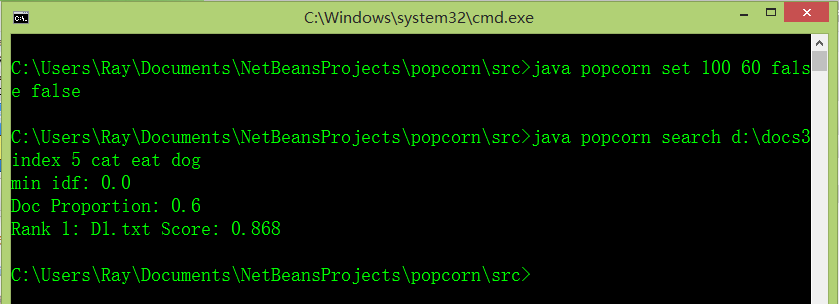
1. Check the result



1. Then set the percentage to 60 which means which documents contains 2/3 terms will be retrieved.

java set 100 60 false false

1. Check the result



Result: From the index.txt screenshots (see Figure 2.3.result), we can see that only d1 contains all 3 of the keywords so the result is as expected.