Information Retrieval System

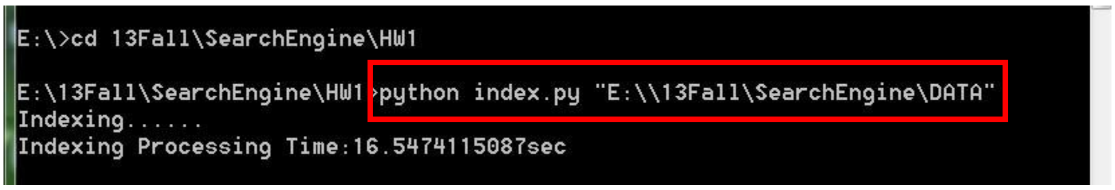
Kaili Zhang
kz2203
2013.10.10

1. Basic Information

1) **Language**: Python
2) **Toolkit**: NLTK
   !important: to use this system, you have to install NLTK firsly.
   This toolkit is only used for document preprocessing (tokenize, stem).
3) **Stop Word List**: Downloaded from Google.Code. https://code.google.com/p/stop-words/

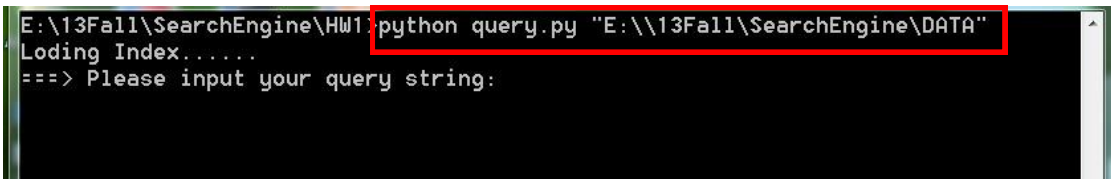
2. How To Use:

1) **Indexing:** python index.py folder_path



```
E:\>cd 13Fall\SearchEngine\HW1

E:\13Fall\SearchEngine\HW1>python index.py "E:\\13Fall\SearchEngine\DATA"
Indexing......
Indexing Processing Time:16.5474115087sec
```

2) **Query**: python query.py folder_path



```
E:\13Fall\SearchEngine\HW1>python query.py "E:\\13Fall\SearchEngine\DATA"
Loding Index......
===> Please input your query string:
```

3. Parameter:

1) **Time** needed to build the index: **16.547 Sec**
   Caused by large index amount.
2) **Indexing size**: **55.9 MB**
   It is big because firsly we build much more indexes, secondly, Dictionary Format costs more.
3) **Indexing File**: indexing_file.file sotring indexing Dictionary
4) **Query Time**: could be seen on top of each query result

4. New and Special!

1) Comprehensive Functions
   All functions required are implemented and query function is quite strong to avoid kinds of bugs.
2) **Indexing Structure:**
   I chose Dictionary as index data structure. (Dictionary is like Hashing Table). Its structure is:
   Index_dict{'Term' : {'docID': {'Part': frequency}} } Part is TITLE, AUTHOR, BIBLIO or TEXT
3) **More Gram Index**:

In order to solve the 'long phrase' problem, in index, except inserting single word as term into indexing, I also inserted phrases consisted by adjacent words as indexes. I call this kind of indexes 'Gram Index', a definition borrowed from NLP. In my indexing, Gram can up to 5. However, 'long phrase' problem is always a problem in realty search engine industry. Another way to solve it is to record the location of each term. But the data structure becomes much more complicated. Thus, I only applied Gram Index.

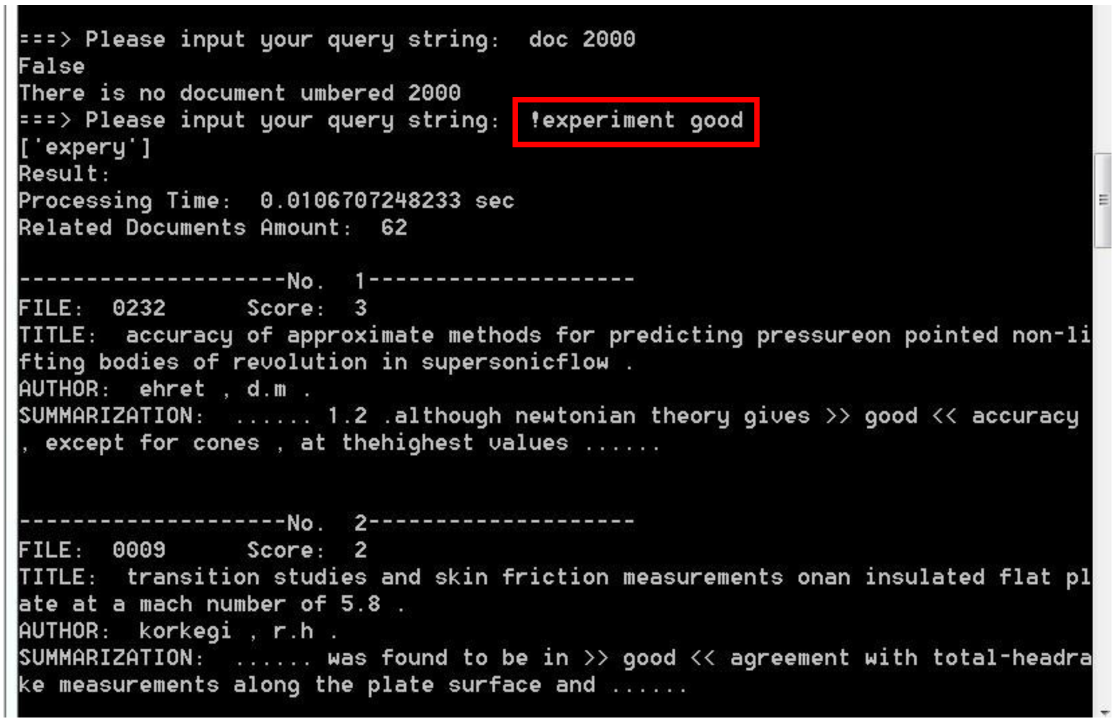
4) **More Boolean Query Acceptable**

This system could accept more complicated Boolean Queries.

For example: cat dog rat ...... up to infinite number of words

"" up to infinite number of phrases

!cat dog



```
===> Please input your query string:  doc 2000
False
There is no document umbered 2000
===> Please input your query string:  !experiment good
['expery']
Result:
Processing Time:  0.0106707248233 sec
Related Documents Amount:  62

--------------------No.  1--------------------
FILE:  0232      Score:  3
TITLE:  accuracy of approximate methods for predicting pressureon pointed non-li
fting bodies of revolution in supersonicflow .
AUTHOR:  ehret , d.m .
SUMMARIZATION:  ...... 1.2 .although newtonian theory gives >> good << accuracy
, except for cones , at thehighest values ......


--------------------No.  2--------------------
FILE:  0009      Score:  2
TITLE:  transition studies and skin friction measurements onan insulated flat pl
ate at a mach number of 5.8 .
AUTHOR:  korkegi , r.h .
SUMMARIZATION:  ...... was found to be in >> good << agreement with total-headra
ke measurements along the plate surface and ......
```

5) **Ranking Algorithm**

In traditional (the method described in hw1), we only count the frequency of words in query. When there is a phrase in query, score is added by size of phrase. However, I modified this algorithm to gain a better retrieval result.

For term appears in TITLE and AUTHOR, add double units point;

For term appears in BIBLIO and TEXT, add one unit point;

For term is phrase, unit point equals to size of phrase.

In this way, when a word A appears once in whole document of Doc1 and Doc2, if A appears in TITLE of Doc1 and TEXT of Doc2, Doc1 may gain a higher score than Doc2.

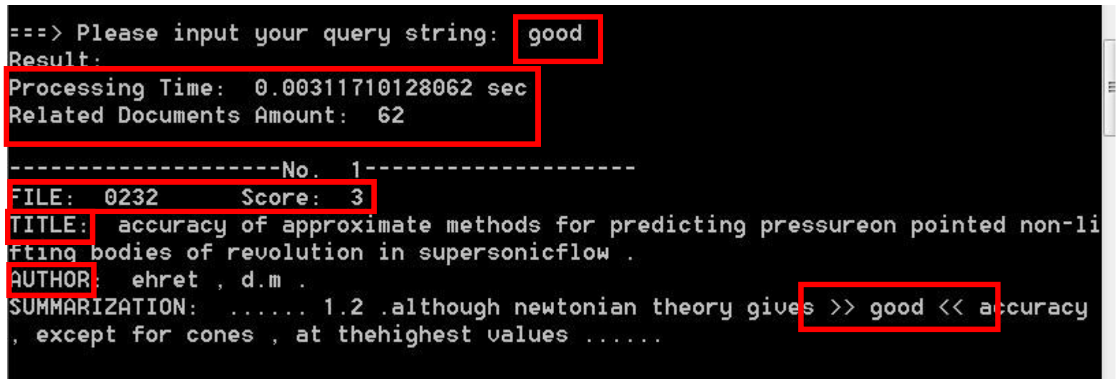
6) **More Statistical Data Provided**

You can find Query Time, Related Doc Amount for each query result and docID, score, TITLE, AUTHOR, SUMMARIZATION for each result entry.

7) High Light Function

Since settings of string's color and background are different between different OS, I only used ">> Term <<" as Hight Light Mark. But, my high light algorithm could high light all the queried words including title, author and summarization.

```
'\033[92m '+token_text[pos]+' \033[0m'
```



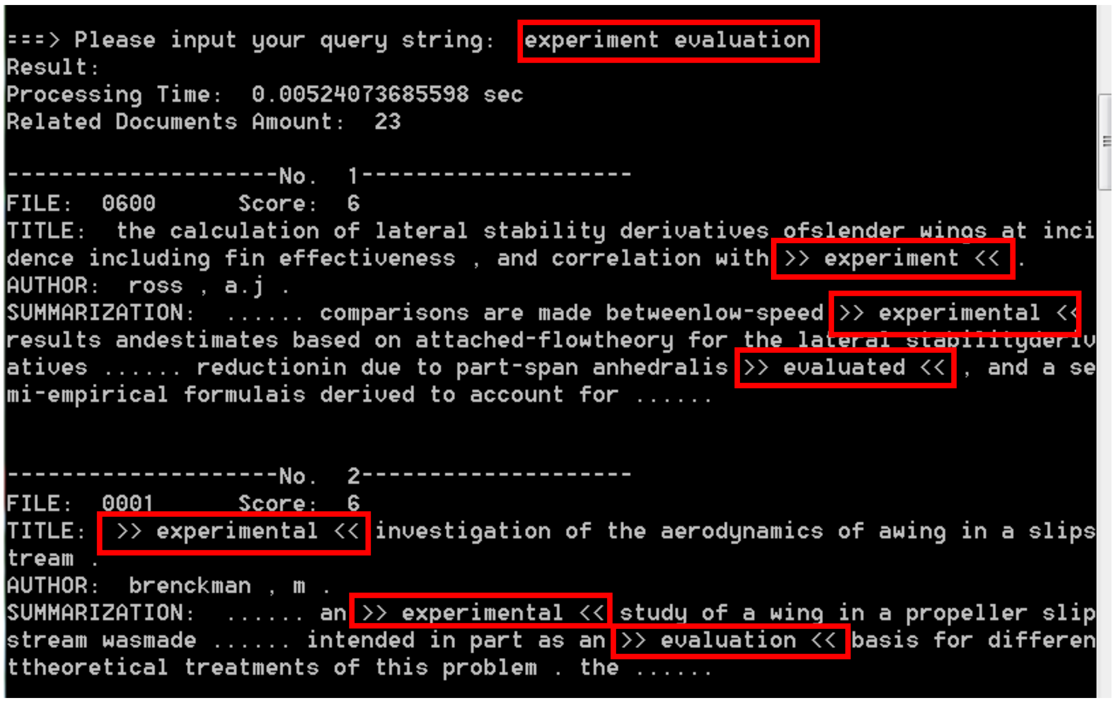
```
===> Please input your query string:  good
Result:
Processing Time:  0.00311710128062 sec
Related Documents Amount:  62

--------------------No. 1--------------------
FILE:  0232       Score:  3
TITLE:  accuracy of approximate methods for predicting pressureon pointed non-li
fting bodies of revolution in supersonicflow .
AUTHOR:  ehret , d.m .
SUMMARIZATION:  ...... 1.2 .although newtonian theory gives >> good << accuracy
, except for cones , at thehighest values ......
```

8) Great Summarization

If queried word appears in text or biblio, then summarization is generated from text, then biblio. If queried word only appears in title or author, then summarization is consisted by title and author.

If no queried word appears in doc, (eg, query = !cat), summarization is generated from text randomly.

Further, if there are two queried words, summarization contains both parts of these two words.

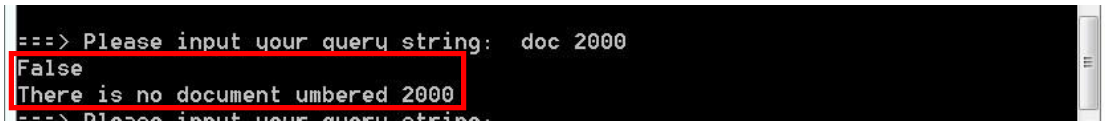


```
===> Please input your query string:  experiment evaluation
Result:
Processing Time:  0.00524073685598 sec
Related Documents Amount:  23

--------------------No. 1--------------------
FILE:  0600       Score:  6
TITLE:  the calculation of lateral stability derivatives ofslender wings at inci
dence including fin effectiveness , and correlation with >> experiment << .
AUTHOR:  ross , a.j .
SUMMARIZATION:  ...... comparisons are made betweenlow-speed >> experimental <<
results andestimates based on attached-flowtheory for the lateral stabilityderiv
atives ...... reductionin due to part-span anhedralis >> evaluated << , and a se
mi-empirical formulais derived to account for ......


--------------------No. 2--------------------
FILE:  0001       Score:  6
TITLE:  >> experimental << investigation of the aerodynamics of awing in a slips
tream .
AUTHOR:  brenckman , m .
SUMMARIZATION:  ...... an >> experimental << study of a wing in a propeller slip
stream wasmade ...... intended in part as an >> evaluation << basis for differen
ttheoretical treatments of this problem . the ......
```

9) **Great Query Interface**

As you've seen, the interface looks neat and elegant.

10) **Careful Error Remind**



```
===> Please input your query string:  doc 2000
False
There is no document umbered 2000
===> Please input your query string:
```

11) **Convinient API**

All the functions could be directly and easily used.