Data Mining Technology for Business and Society

Homework # 1

1536242 - Paolo Tamagnini

April 2016

1 Simple statistic on the used data-set

```
python data-set_stats.py

STATS ON CRAN DATA-SET
the number of documents is:
1400
the average number of words in a document is:\\
168.191428571
the number of words in the data-set is:
228502
but the number of different words in the data-set is:
12260
this means that out of 228502 the % of repeated words is:
94.6346202659 %
the total number of queries for Cran is:
225
the average number of word in a query is:
17.3244444444


STATS ON TIME DATA-SET
the number of documents is:
423
the average number of words in a document is:
629.520094563
the number of words in the data-set is:
264595
but the number of different words in the data-set is:
29842
this means that out of 264595 the % of repeated words is:
88.7216311722 %
the total number of queries for Time is:
83
the average number of word in a query is:
15.3012048193
```

2 List of used stemmers

- **Default stemmer**
  The index is built without using grammar details, but just doing basic operations like removing punctuation and symbols. The index will not be case sensitive.

- **English stemmer**
  The index is built using english grammar to stem word, merging words with the same root.

- **English stemmer with no stopwords**
  Same as before but this time the index won't contain any stopword (articles, preposition and similar grammar entities with no independent meaning).

3 List of used scorer functions

- **Count the occurencies**
  This scorer is really basic and it just counts how many times the query words are present in a suggested document.

- **TF/IDF**
  In this case we use the ratio:

  $$\frac{term frequency}{inverse\, document\, frequency}$$

  so that it is able to rank each document weighting more rare words than usual words.

- **BM25**
  This scorer is an improvement of the *TF/IDF*, taking into account lengths of documents to rank them.

4 The script to create the collection

We can use the java package MG4J to create a collection as follows.

```
find <DATASET PATH> -iname \*.html | java it.unimi.di.big.mg4j
    .document.FileSetDocumentCollection -f HtmlDocumentFactory
    -p encoding=UTF-8 <COLLECTION PATH>
```

5 The scripts to create the inverted indexes

We can use the java package MG4J to create an index as follows.

```
java it.unimi.di.big.mg4j.tool.IndexBuilder <STEMMER USED> -S
    <COLLECTION PATH> <INDEX-PATH>
```

6 The scripts to obtain the results from the search engine

We can use the java package MG4J to make our queries as follows.

```
java homework.RunAllQueries_HW "cranData/defStem/index/cranDef
    " <QUERY FILE FULL PATH> "<SCORER FUNCTION>" "1:2" <OUTPUT
    PATH>
```

7 Full script

This code is contained in MG4J-terminal-commands.sh

```
source set-my-classpath.sh

# Cran Dataset Default Stemmer

mkdir cranData/defStem

mkdir cranData/defStem/index

find cranData/Cranfield_DATASET -iname \*.html | java it.unimi
    .di.big.mg4j.document.FileSetDocumentCollection -f
    HtmlDocumentFactory -p encoding=UTF-8 cranData/defStem/
    index/cranDef.collection

java it.unimi.di.big.mg4j.tool.IndexBuilder --downcase -S
    cranData/defStem/index/cranDef.collection cranData/defStem
    /index/cranDef
```

```
mkdir cranData/defStem/results

java homework.RunAllQueries_HW "cranData/defStem/index/cranDef
    " /home/paolotamag/Desktop/HW1/cranData/Cranfield_DATASET/
    cran_all_queries.tsv "CountScorer" "1:2" cranData/defStem/
    results/cranDef_CountOcc.tsv

java homework.RunAllQueries_HW "cranData/defStem/index/cranDef
    " /home/paolotamag/Desktop/HW1/cranData/Cranfield_DATASET/
    cran_all_queries.tsv "TfIdfScorer" "1:2" cranData/defStem/
    results/cranDef_TfIdf.tsv

java homework.RunAllQueries_HW "cranData/defStem/index/cranDef
    " /home/paolotamag/Desktop/HW1/cranData/Cranfield_DATASET/
    cran_all_queries.tsv "BM25Scorer" "1:2" cranData/defStem/
    results/cranDef_BM25.tsv



# Cran Dataset English Stemmer

mkdir cranData/engStem

mkdir cranData/engStem/index

find cranData/Cranfield_DATASET -iname \*.html | java it.unimi
    .di.big.mg4j.document.FileSetDocumentCollection -f
    HtmlDocumentFactory -p encoding=UTF-8 cranData/engStem/
    index/cranEng.collection

java it.unimi.di.big.mg4j.tool.IndexBuilder -t it.unimi.di.big
    .mg4j.index.snowball.EnglishStemmer -S cranData/engStem/
    index/cranEng.collection cranData/engStem/index/cranEng

mkdir cranData/engStem/results

java homework.RunAllQueries_HW "cranData/engStem/index/cranEng
    " /home/paolotamag/Desktop/HW1/cranData/Cranfield_DATASET/
    cran_all_queries.tsv "CountScorer" "1:2" cranData/engStem/
    results/cranEng_CountOcc.tsv

java homework.RunAllQueries_HW "cranData/engStem/index/cranEng
    " /home/paolotamag/Desktop/HW1/cranData/Cranfield_DATASET/
    cran_all_queries.tsv "TfIdfScorer" "1:2" cranData/engStem/
    results/cranEng_TfIdf.tsv

java homework.RunAllQueries_HW "cranData/engStem/index/cranEng
    " /home/paolotamag/Desktop/HW1/cranData/Cranfield_DATASET/
    cran_all_queries.tsv "BM25Scorer" "1:2" cranData/engStem/
```

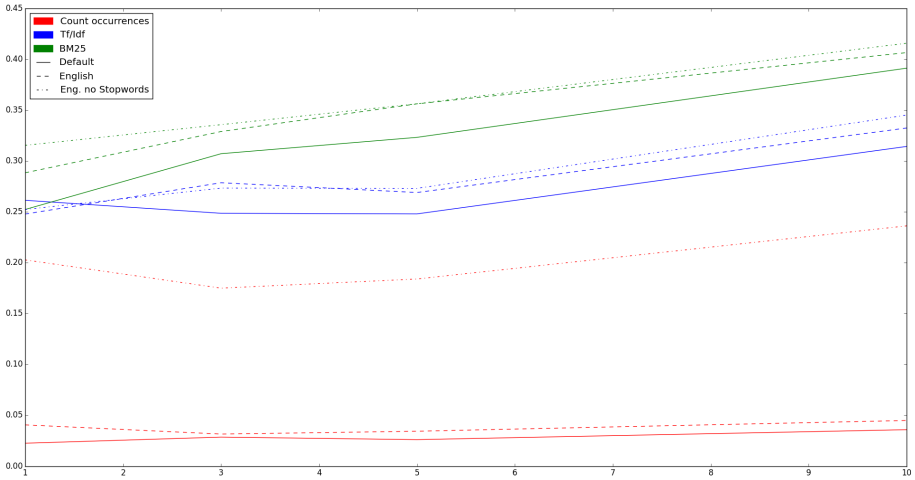
```
    results/cranEng_BM25.tsv



# Cran Dataset English Stemmer with no Stopwords

mkdir cranData/engNSWStem

mkdir cranData/engNSWStem/index

find cranData/Cranfield_DATASET -iname \*.html | java it.unimi
    .di.big.mg4j.document.FileSetDocumentCollection -f
    HtmlDocumentFactory -p encoding=UTF-8 cranData/engNSWStem/
    index/cranEngNSW.collection

java it.unimi.di.big.mg4j.tool.IndexBuilder -t homework.
    EnglishStemmerStopwords -S cranData/engNSWStem/index/
    cranEngNSW.collection cranData/engNSWStem/index/cranEngNSW

mkdir cranData/engNSWStem/results

java homework.RunAllQueries_HW "cranData/engNSWStem/index/
    cranEngNSW" /home/paolotamag/Desktop/HW1/cranData/
    Cranfield_DATASET/cran_all_queries.tsv "CountScorer" "1:2"
     cranData/engNSWStem/results/cranEngNSW_CountOcc.tsv

java homework.RunAllQueries_HW "cranData/engNSWStem/index/
    cranEngNSW" /home/paolotamag/Desktop/HW1/cranData/
    Cranfield_DATASET/cran_all_queries.tsv "TfIdfScorer" "1:2"
     cranData/engNSWStem/results/cranEngNSW_TfIdf.tsv

java homework.RunAllQueries_HW "cranData/engNSWStem/index/
    cranEngNSW" /home/paolotamag/Desktop/HW1/cranData/
    Cranfield_DATASET/cran_all_queries.tsv "BM25Scorer" "1:2"
    cranData/engNSWStem/results/cranEngNSW_BM25.tsv




# Time Dataset Default Stemmer

mkdir timeData/defStem

mkdir timeData/defStem/index

find timeData/Time_DATASET -iname \*.html | java it.unimi.di.
    big.mg4j.document.FileSetDocumentCollection -f
    HtmlDocumentFactory -p encoding=UTF-8 timeData/defStem/
```

```
    index/timeDef.collection

java it.unimi.di.big.mg4j.tool.IndexBuilder --downcase -S
    timeData/defStem/index/timeDef.collection timeData/defStem
    /index/timeDef

mkdir timeData/defStem/results

java homework.RunAllQueries_HW "timeData/defStem/index/timeDef
    " /home/paolotamag/Desktop/HW1/timeData/Time_DATASET/
    time_all_queries.tsv "CountScorer" "1:2" timeData/defStem/
    results/timeDef_CountOcc.tsv

java homework.RunAllQueries_HW "timeData/defStem/index/timeDef
    " /home/paolotamag/Desktop/HW1/timeData/Time_DATASET/
    time_all_queries.tsv "TfIdfScorer" "1:2" timeData/defStem/
    results/timeDef_TfIdf.tsv

java homework.RunAllQueries_HW "timeData/defStem/index/timeDef
    " /home/paolotamag/Desktop/HW1/timeData/Time_DATASET/
    time_all_queries.tsv "BM25Scorer" "1:2" timeData/defStem/
    results/timeDef_BM25.tsv



# Time Dataset English Stemmer

mkdir timeData/engStem

mkdir timeData/engStem/index

find timeData/Time_DATASET -iname \*.html | java it.unimi.di.
    big.mg4j.document.FileSetDocumentCollection -f
    HtmlDocumentFactory -p encoding=UTF-8 timeData/engStem/
    index/timeEng.collection

java it.unimi.di.big.mg4j.tool.IndexBuilder -t it.unimi.di.big
    .mg4j.index.snowball.EnglishStemmer -S timeData/engStem/
    index/timeEng.collection timeData/engStem/index/timeEng

mkdir timeData/engStem/results

java homework.RunAllQueries_HW "timeData/engStem/index/timeEng
    " /home/paolotamag/Desktop/HW1/timeData/Time_DATASET/
    time_all_queries.tsv "CountScorer" "1:2" timeData/engStem/
    results/timeEng_CountOcc.tsv

java homework.RunAllQueries_HW "timeData/engStem/index/timeEng
    " /home/paolotamag/Desktop/HW1/timeData/Time_DATASET/
```

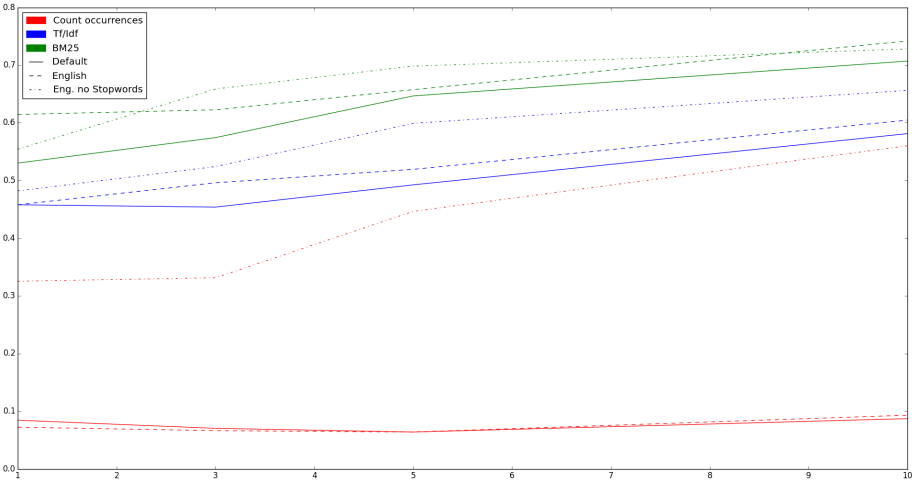
```
    time_all_queries.tsv "TfIdfScorer" "1:2" timeData/engStem/
    results/timeEng_TfIdf.tsv

java homework.RunAllQueries_HW "timeData/engStem/index/timeEng
    " /home/paolotamag/Desktop/HW1/timeData/Time_DATASET/
    time_all_queries.tsv "BM25Scorer" "1:2" timeData/engStem/
    results/timeEng_BM25.tsv



# Time Dataset English Stemmer with no Stopwords

mkdir timeData/engNSWStem

mkdir timeData/engNSWStem/index

find timeData/Time_DATASET -iname \*.html | java it.unimi.di.
    big.mg4j.document.FileSetDocumentCollection -f
    HtmlDocumentFactory -p encoding=UTF-8 timeData/engNSWStem/
    index/timeEngNSW.collection

java it.unimi.di.big.mg4j.tool.IndexBuilder -t homework.
    EnglishStemmerStopwords -S timeData/engNSWStem/index/
    timeEngNSW.collection timeData/engNSWStem/index/timeEngNSW

mkdir timeData/engNSWStem/results

java homework.RunAllQueries_HW "timeData/engNSWStem/index/
    timeEngNSW" /home/paolotamag/Desktop/HW1/timeData/
    Time_DATASET/time_all_queries.tsv "CountScorer" "1:2"
    timeData/engNSWStem/results/timeEngNSW_CountOcc.tsv

java homework.RunAllQueries_HW "timeData/engNSWStem/index/
    timeEngNSW" /home/paolotamag/Desktop/HW1/timeData/
    Time_DATASET/time_all_queries.tsv "TfIdfScorer" "1:2"
    timeData/engNSWStem/results/timeEngNSW_TfIdf.tsv

java homework.RunAllQueries_HW "timeData/engNSWStem/index/
    timeEngNSW" /home/paolotamag/Desktop/HW1/timeData/
    Time_DATASET/time_all_queries.tsv "BM25Scorer" "1:2"
    timeData/engNSWStem/results/timeEngNSW_BM25.tsv
```

8 The Plots of the average P@k

Plot for the Cranfield data-set:



Plot for the Time data-set:



9 Conclusion

From what we can see from the graph the best performance stemmer is English Stemmer with no Stopwords, while the best scorer function is BM25.
Even then it is not better by far from the combination:

$$< English\,Stemmer - BM25 >$$

the combination:

$$< English\,Stemmer\,with\,no\,Stopwords - BM25 >$$

is giving us the highest P@k values, meaning it is the best combination in both data-set.