Sapienza
Università di Roma

Data Mining Technology for Business and Society

Homework 1

Author:
Francesco Romeo
(Student ID: 1618216)

Supervisors:
Stefano Leonardi
Adriano Fazzone

April 2020

1 Search Engine Evaluation (Part 1)

1.1 Software Information

SW_1: it imports the .html files and stores them into a .csv file, creates the inverted index for the all Search Engines configurations and performs the query search.
SW_2: it performs 4 Search Engine evaluation metrics: MRR, R-Precision, P@k and nDCG.

1.2 Indexed Documents, Queries Numbers and Ground Truth Information

For Cranfield_DATASET 1400 documents have been indexed and 222 Queries have been queried. For Time_DATASET 423 documents have been indexed and 83 Queries have been queried.
The Search Engine evaluation for Cranfield dataset is performed by using the 110 queries contained in the ”cran_Ground_Truth.tsv” file. Instead, for Time dataset the Search Engine evalutation has been performed by using the 80 queries contained into the ”time_Ground_Truth.tsv” file.

1.3 Search Engine Configuration

In order to perform the Search Engine evaluation assignment, it has been decided to evaluate 24 different search engine configurations for both documents datasets. Eight different text analyzers and three different scoring functions have been chosen. The 24 configurations are reported in the table below. However, for parsing the queries in Time documents dataset the QueryParser() module has been chosen because only the body field has been taken into account when a query is performed. This is due to the fact that the title field contains only noise, and therefore it has been discarded and it has neither been included into the Schema configuration. On the other hand, for parsing the queries in Cranfield documents dataset the MultifieldParser() module has been used because both title and body fields contain relevant information.

SE_ID	Analyzer	Scoring Function
01	StemmingAnalyzer	TF_IDF
02	StemmingAnalyzer	Frequency
03	StemmingAnalyzer	BM25F(B=0.75,K1=1.2)
04	StandardAnalyzer	TF_IDF
05	StandardAnalyzer	Frequency
06	StandardAnalyzer	BM25F(B=0.75,K1=1.2)
07	RegexAnalyzer	TF_IDF
08	RegexAnalyzer	Frequency
09	RegexAnalyzer	BM25F(B=0.75,K1=1.2)
10	SimpleAnalyzer	TF_IDF
11	SimpleAnalyzer	Frequency
12	SimpleAnalyzer	BM25F(B=0.75,K1=1.2)
13	FancyAnalyzer	TF_IDF
14	FancyAnalyzer	Frequency
15	FancyAnalyzer	BM25F(B=0.75,K1=1.2)
16	NgramAnalyzer(4)	TF_IDF
17	NgramAnalyzer(4)	Frequency
18	NgramAnalyzer(4)	BM25F(B=0.75,K1=1.2)
19	KeywordAnalyzer	TF_IDF
20	KeywordAnalyzer	Frequency
21	KeywordAnalyzer	BM25F(B=0.75,K1=1.2)
22	LanguageAnalyzer('en')	TF_IDF
23	LanguageAnalyzer('en')	Frequency
24	LanguageAnalyzer('en')	BM25F(B=0.75,K1=1.2)

Table 1: 24 Search Engine configuration summary

1.4 Mean Reciprocal Rank (MRR)

MRR doesn't consider the all good retrieved documents but only the first relevant document. Table below shows:

- MRR results for 24 search engines configuration used for Cranfiled documents dataset
- MRR results for 24 search engines configuration used for Time documents dataset

The yellow highlighted rows represent the top 5 search engine for both datasets. The first two columns are refered to the Cranfield dataset and the last two columns for Time dataset.

SE_ID (Cranfield)	MRR (Cranfield)	SE_ID (Time)	MRR (Time)
SE_24	0.521	SE_24	0.715
SE_6	0.518	SE_3	0.696
SE_15	0.518	SE_6	0.679
SE_3	0.508	SE_15	0.678
SE_9	0.476	SE_12	0.669
SE_12	0.476	SE_18	0.639
SE_21	0.428	SE_22	0.564
SE_18	0.419	SE_4	0.536
SE_22	0.418	SE_13	0.536
SE_1	0.405	SE_1	0.503
SE_4	0.388	SE_23	0.462
SE_13	0.388	SE_5	0.461
SE_23	0.348	SE_14	0.461
SE_16	0.329	SE_2	0.426
SE_2	0.321	SE_16	0.421
SE_14	0.319	SE_17	0.347
SE_5	0.314	SE_10	0.243
SE_17	0.264	SE_11	0.157
SE_7	0.168	SE_7	0.043
SE_10	0.168	SE_9	0.029
SE_19	0.155	SE_8	0.027
SE_8	0.061	SE_19	0.001
SE_11	0.061	SE_20	0.001
SE_20	0.048	SE_21	0.001

Table 2: MRR for Cranfield and Time documents datasets

1.5 R-Precision distribution table with data from the Top-5 search engine configurations according to MRR table

By using this metric, it is possible to skip the issue of choosing a value for k, however there is the problem that it does not penalize results that are not in the first positions (i.e. it doesn't consider the order of the retrieved documents, as it also happens in P@K). Table 3 refers to the R-Precison for Cranfield Search engines whereas Table 4 refers to the R-Precision for Time Search engines.

SE_ID	mean	min	Q1	Median	Q3	max
SE_24	0.2726	0	0	0.25	0.4533	1
SE_3	0.2655	0	0	0.25	0.4286	1
SE_6	0.2582	0	0	0.25	0.4286	1
SE_15	0.2582	0	0	0.25	0.4286	1
SE_9	0.2457	0	0	0.25	0.4286	0.67

Table 3: R-Precison Cranfield

SE_ID	mean	min	Q1	Median	Q3	max
SE_24	0.5556	0	0.1917	0.5584	1	1
SE_15	0.5487	0	0.3214	0.5	0.8889	1
SE_6	0.5476	0	0.3214	0.5	0.8889	1
SE_12	0.5420	0	0.3214	0.5	0.8889	1
SE_3	0.5273	0	0.1917	0.5	0.8889	1

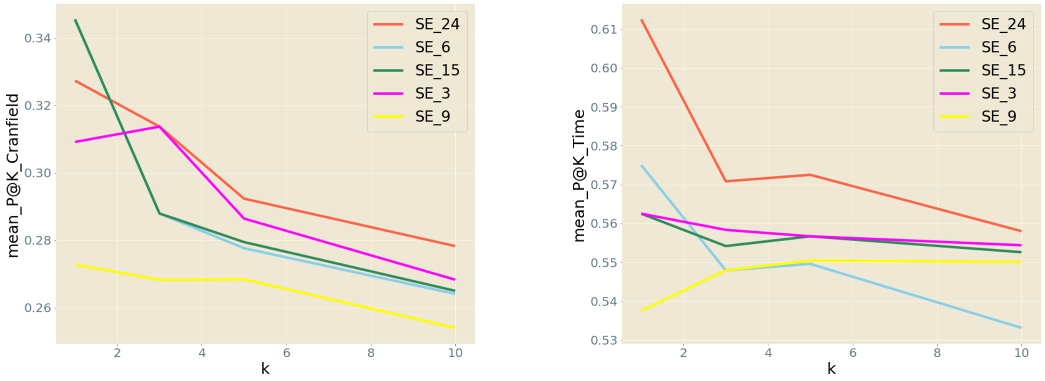
Table 4: R-Precison Time

1.6 The P@k-plots with data from the Top-5 search engine configurations according to the MRR table

P@K is computed by dividing the number of k relevant documents by the minimum value among k and the query Ground Truth length(|GT(q)|). The drawbacks of this metric are:

- it doesn't take into account the order of the top-k relevant documents.
- the issue of choosing a value k.

The left plot shows the top 5 Search Engine configurations curves according to the mean P@K for Cranfield documents dataset whereas the right plot refers to the top 5 Search Engines configurations curves according to the mean P@k for Time documents dataset. As it is possible to see from the two plots, when k increases, the P@K of the 5 Search Engines tends to decrease. Moreover, SE_24 seems still to be the top Search Engine configuration for both documents datasets even if it has an accentuated descent when it goes from k=1 to k=3, this is verifiable in particular for Time documents dataset.



(a) Average P@K Cranfield

(b) Average P@k Time

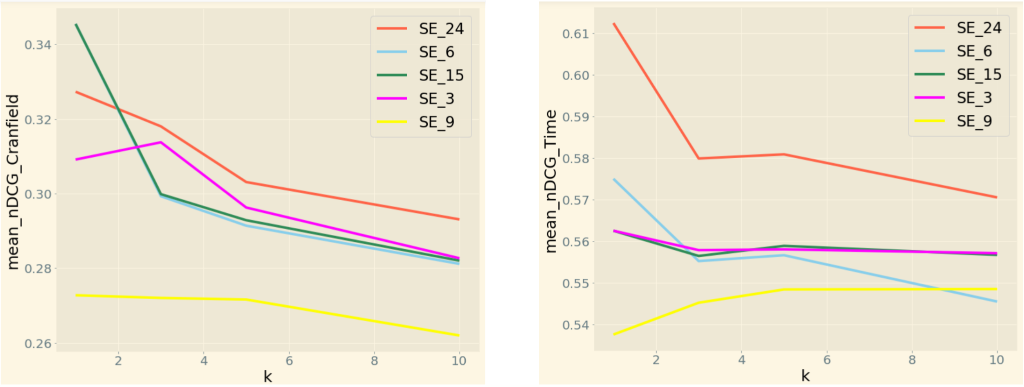
Figure 1: Average P@K for Cranfield and Time

1.7 The nDCG@k-plots with data from the Top-5 search engine configurations according to the MRR table

The nDCG tries to overcome the drawbacks of previous metrics, and it actually partially does it, because:

- it takes into account more than 1 result.
- it takes into account the position of the documents, i.e. it penalizes much more relevant docs that are not in the first positions.

However, the drowback of this metric is that there is still the issue of choosing a value for k. The left plot shows the top 5 Search Engine configurations curves according to the average nDCG@k for Cranfield documents dataset whereas the right plot refers to the top 5 Search Engines configurations curves according to the average nDCG@k for Time documents dataset. As it is possible to see from the two plots, when k increases, the nDCG decreases. Moreover, SE_24 is still the top Search Engine configuration for both documents datasets. In fact, by using the 4 metrics, it is the Search Engine configuration that performs better when k is greater than 1.



(a) Average nDCG@k Cranfield

(b) Average nDCG@k Time

Figure 2: Average nDCG@k for Cranfield and Time

2 Near Duplicates Detection (Part 2)

2.1 Software Information

SW_1 NDD: it imports the songs, creates the set of shingles and store everything into .tsv file.
SW_2 NDD: it imports the found near duplicate couples and plots the bar plot.

2.2 Number of rows (r) and number of bands (b)

Number of rows (r) and number of bands (b) give information about the Min Hashing Sketches length. In fact r represents the number of rows (elements) that are in one band, whereas b represents the number of bands. The length of the Min Hashing Sketches is given by the multiplication between r and b. Choosing r and b is not a straightforward issue in fact, by changing them is possible to reduce/increase the probability of having FP and FN. FP and FN are inversely correlated, for that reason is necessary to find a trade-off among them, remembering always that if FN are excluded from the set of candidate pairs they are lost forever and cannot be retrieved. For that reason it is important to maintain the probability of FN quite low. However, having a quite low FN probability implies that the probability of having FP (in particular for pairs that are close to the Jaccard Threshold) is very high. Having a high FP probability implies that the probability of having a bigger set of candidate pairs is very high, and having a bigger set of candidate pairs, through which browsing, implies more computations and therefore more time to retrieve near duplicates. For that reason it is necessary to choose them carefully and taking into consideration also case restrictions.
In this scenario, r = 10 and b = 10 have been chosen as parameters. Many different configurations have been taken into account, but these two values allow to reduce the probability of having FN at the expense of a not too high probability to have FP in the set of candidate pairs, in particular for values far from the Jaccard Threshold. Moreover, with r=10 and b=10 it is possible not to violate the first two constrains because the Min-Hashing sketch is < 300 and the probability of having near duplicate candidate pairs with a Jaccard=0.89 is 0.976 which is > 0.97.
The Execution Time required by the machine to execute the Near-Duplicates-Detection tool, that returns the near duplicates with the approximate Jaccard similarity, is of 2minutes and 29seconds.

2.3 False Negative(FN) Probability

Given the number of rows(r) in a band, the number of bands(b) and a Jaccard similarity(J) among two documents, the probability that two documents are identical in 1 band is given by: p = J^r. So, the probability that the two documents are different in one band is: p = $(1 - J^r)$. Because of this, the probability that they will be different in all bands is: p = $(1 - J^r)^b$. This last probability gives us information about the probability that two documents, with a given Jaccard similarity, will not be included into the set of candidate pairs. Moreover, if their Jaccard Similarity is $\geq$ than a Jaccard Threshold, this probability can be used to detected the probability of having FN; i.e. the probability that two documents, with a real Jaccard similarity at least equal to the threshold, will not be included in the set of candidate pairs.
In the studied case the probability of having FN for Jaccard Similarities = 0.89,0.9,0.95 and 1 is:

FN(J=0.89)=$(1 - J^r)^b = (1 - 0.89^{10})^{10} = 0.024 = 2.4\%$

FN(J=0.9)=(1 - J^r)^b = (1 - 0.9^{10})^{10} = 0.014 = 1.4%

FN(J=0.95)=(1 - J^r)^b = (1 - 0.95^{10})^{10} = 0, 0001 = 0.01%

FN(J=1)=(1 - J^r)^b = (1 - 1^{10})^{10} = 0

2.4 False Positive(FP) in the set of candidate pairs

As reported in the previous subsection, the probability that two documents, with a real Jaccard similarity, are equal in 1 band is: p = J^r. Also from above, it is possible to remember that the probability that they will be different in all bands is p = (1 - J^r)^b. So, the probability that they will be identical, at least in 1 band, is equal to: p = 1 - (1 - J^r)^b. This last probability gives us information about the probability that two documents will be picked to be included in the set of candidate pairs. Now, if the Jaccard similarity among these two documents is below the Jaccard Threshold, this probability refers to the probability of having FP, i.e. the probability that two documents with a real Jaccard similarity below the threshold will be included in the set of candidate pairs.
In the studied case the probability of having FP for Jaccard Similarities = 0.85, 0.8, 0.75, 0.7, 0.65, 0.6, 0.55 and 0.5 is:

FN(J=0.85)=1 - (1 - J^r)^b= 1 - (1 - 0.85^{10})^{10} = 0.89 = 89%

FN(J=0.8)=1 - (1 - J^r)^b= 1 - (1 - 0.8^{10})^{10} = 0.68 = 68%

FN(J=0.75)=1 - (1 - J^r)^b= 1 - (1 - 0.75^{10})^{10} = 0.44 = 44%

FN(J=0.7)=1 - (1 - J^r)^b= 1 - (1 - 0.7^{10})^{10} = 0.25 = 25%

FN(J=0.65)=1 - (1 - J^r)^b= 1 - (1 - 0.65^{10})^{10} = 0.13 = 13%

FN(J=0.6)=1 - (1 - J^r)^b= 1 - (1 - 0.6^{10})^{10} = 0.06 = 6%

FN(J=0.55)=1 - (1 - J^r)^b= 1 - (1 - 0.55^{10})^{10} = 0.03 = 3%

FN(J=0.5)=1 - (1 - J^r)^b= 1 - (1 - 0.5^{10})^{10} = 0.01 = 1%

2.5 Near-Duplicates couples

By running the Near-Duplicates-Detection tool, that took 2minutes and 29seconds to be executed, the number of all found Near Duplicate couples is 39567.
The Bar Plot below shows the number of Near-Duplicate couples that have been found with an approximated Jaccard similarity value of at least 0.89, 0.90, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97,

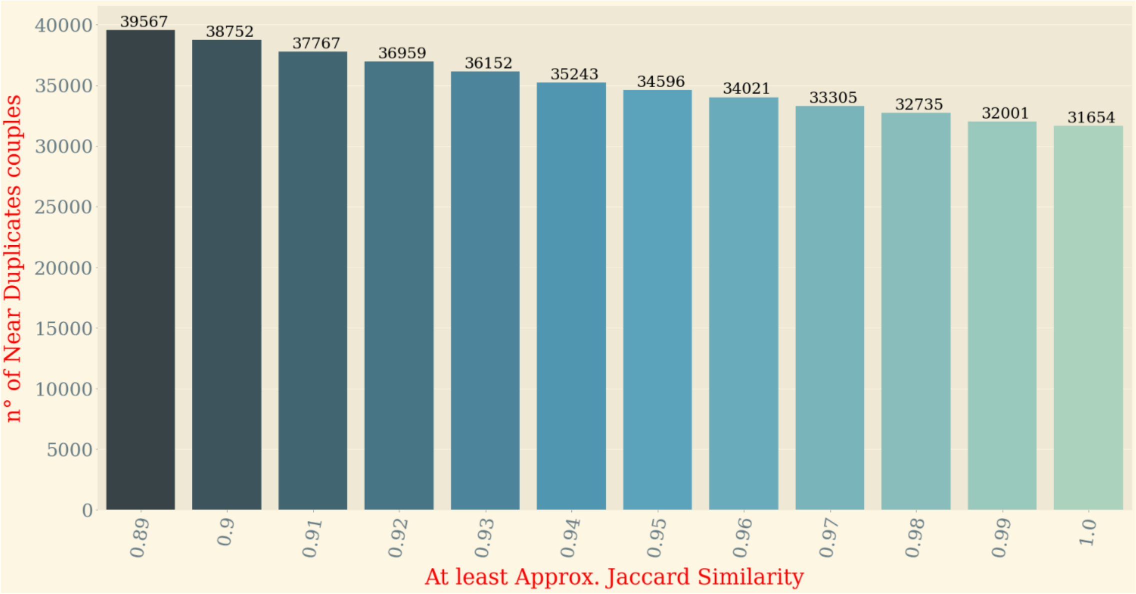


Figure 3: Number of Near-Duplicates couples that have been found with an approximated Jaccard similarity value of at least 0.89, 0.90, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97,0.98, 0.99, 1.