Homework 1

Data Mining Technology for Business and Society

Part 1

Search Engine Evaluation:

We have a dataset of **1400** documents, and a set of **225** queries.

We created 18 search engine using Whoosh. We changed the configuration based on the weighting function and the analyzer, we used three analyzers (Simple, Standard, and Stemming) and for each one we created the index of the documents, and then when executing the search - for scoring- we used the algorithms: (frequency, tf_idf, bm25f: with 4 different parameters).

Search Engine	Config		
	Analyzer	Scoring Algo	Parameters
SE_1	Simple	frequency	-
SE_2	Simple	Tf_idf	-
SE_3	Simple	bm25f	B = 0.35, K1 = 0.7
SE_4	Simple	bm25f	B = 0.75, K1 = 1.2
SE_5	Simple	bm25f	B = 0.75, K1 = 2.3
SE_6	Simple	bm25f	B = 0.9, K1 = 1.1
SE_7	Standard	frequency	
SE_8	Standard	Tf_idf	
SE_9	Standard	bm25f	B = 0.35, K1 = 0.7
SE_10	Standard	bm25f	B = 0.75, K1 = 1.2
SE_11	Standard	bm25f	B = 0.75, K1 = 2.3
SE_12	Standard	bm25f	B = 0.9, K1 = 1.1
SE_13	Stemming	frequency	
SE_14	Stemming	Tf_idf	
SE_15	Stemming	bm25f	B = 0.35, K1 = 0.7
SE_16	Stemming	bm25f	B = 0.75, K1 = 1.2
SE_17	Stemming	bm25f	B = 0.75, K1 = 2.3
SE_18	Stemming	bm25f	B = 0.9, K1 = 1.1

R-Precision distribution table:

SE	Mean	Min	1 Q	Median	3 Q	Max
SE_1	0.006168	0	0	0	0	0.25
SE_2	0.022130	0	0	0	0	1
SE_3	0.054868	0	0	0	0	1
SE_4	0.059041	0	0	0	0	1
SE_5	0.059409	0	0	0	0	1
SE_6	0.058831	0	0	0	0	1
SE_7	0.037460	0	0	0	0	1

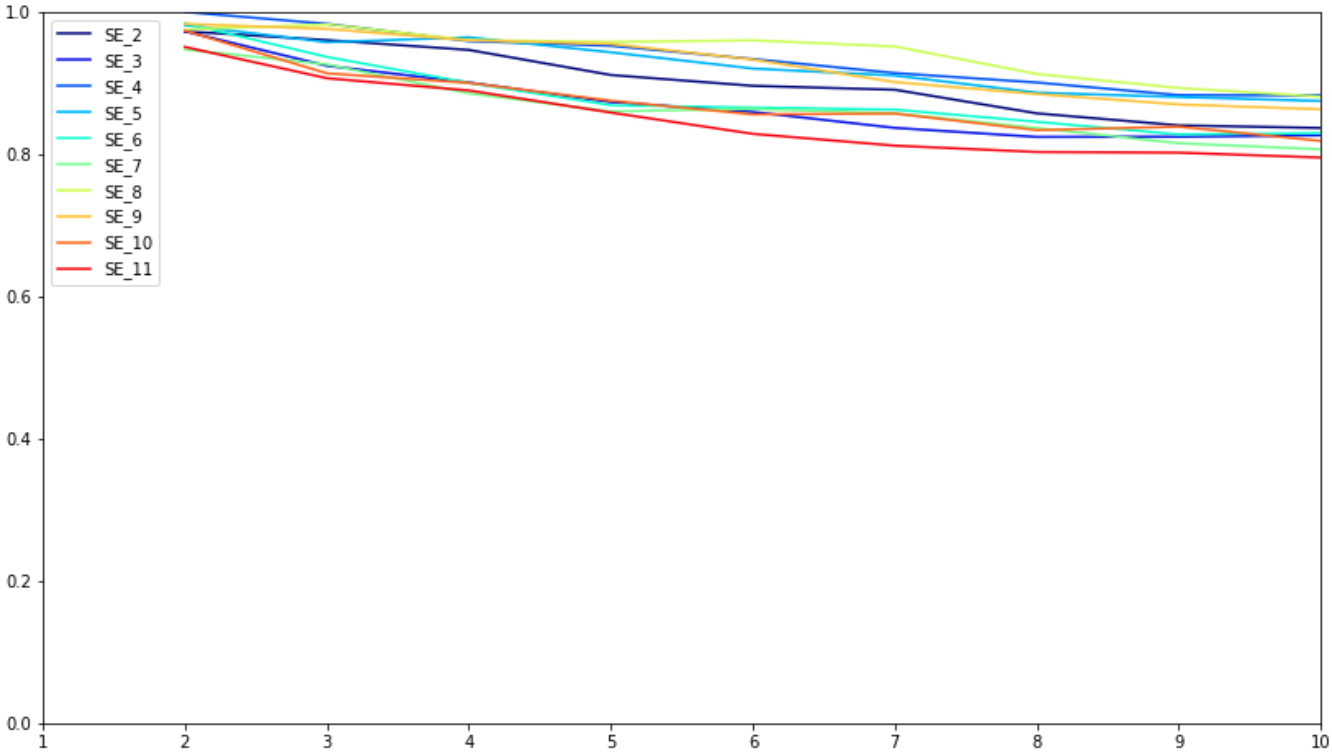
SE_8	0.038472	0	0	0	0	1
SE_9	0.056892	0	0	0	0	1
SE_10	0.059242	0	0	0	0	1
SE_11	0.061139	0	0	0	0	1
SE_12	0.062725	0	0	0	0	1
SE_13	0.031753	0	0	0	0	0.5
SE_14	0.044781	0	0	0	0	0.5
SE_15	0.050809	0	0	0	0	1
SE_16	0.062174	0	0	0	0	1
SE_17	0.063526	0	0	0	0	1
SE_18	0.065168	0	0	0	0	1

NDCG plot:

For the plot we plotted only 12 search engines, removing the last two configurations from the list.

```
plt.figure(figsize=(14,8))
jet= plt.get_cmap('jet')
colors = iter(jet(numpy.linspace(0,1,11)))
ranks = [i for i in range(2,11)]

for se_ in range(1,len(se)):
    ndcgs=[]
    for rank in ranks:
        ndcgs.append(se[se_].ndcg2(rank))
    if se[se_].valid: plt.plot(ranks,ndcgs,color=next(colors), label = 'SE_'+str(se_))
    plt.axis([1, 10, 0, 1])
plt.legend()
plt.show()
```



We notice that the search engine was not plotted because it failed the MRR constaint