The Information Retrieval Task: Ranking of Studies for Systematic Reviews - Real Use-Case

Jawaher A. Alghamdi
j.alghamdi@uqconnect.edu.au

The University of Queensland, Australia
Engineering and Information Technology Faculty

Abstract. The improvements of the *effectiveness* in search engines have led to a significant progress in many areas. There are many methods can be used to ensure the improvements of such effectiveness. In this work, we have considered two out of many ranking functions (namely, TF-IDF, BM25) and three rank fusion algorithms which are so-called Borda, CombSUM and CombMNZ; we, in addition to that, have implemented two types of different reduction methods which are so-called KLI and IDF-r. As summarised results, final findings calculated for all methods which include TF-IDF, BM25, Borda, CombSUM, CombMNZ. By applying evaluation measures on the aforementioned methods, we have come to the conclusion that BM25 outperforms TF-IDF , CombMNZ and CombSUM, in turn, surpass the Borda fusion algorithm. Comparing the obtained results between *(title queries and keyword queries)*, BM25 has more accurate results on title queries, and all fusion methods are more higher than that of Boolean queries. Moreover, CombMNZ and CombSUM have higher mean average precision than Borda and MAP shows BM25 ranking model best performance than TF-IDF ranking method. Regarding the reduction methods, the findings have shown that the KLI method is more effective than IDF-r. In this report, we firstly,in chapter 1, discuss both title and Boolean queries, we present an overview of the retrieval methods which we considered; then we stated some details about the empirical evaluation settings which have been carried out on our experiments; chapter 2 is devoted for the discussion of two reduction methods and their findings that we obtained; some details about the evaluation and analysis were also employed.

2

Query Extraction

1 Retrieval Methods Considered

Enhancement of IR effectiveness is challenge-able. Such effectiveness can be improved utilizing different techniques lead to an optimal ranking scores and therefore an optimal effectiveness. Information retrieval ranking techniques such as TF-IDF; BM25; fusion algorithms such as Borda, which is rank-based method; CombSUM and CombMNZ which are known to be score-based methods [8], are used to rank and score the documents in order to improve the search engine. Thankfully, we are given the collection and the relevance assessments which is the relationship between the documents and the queries (for those who are interested to know more about relevance judgments web-based tool, see [5]). Having that, we have to extract the queries and thus apply some of the ranking methods on the resulted files. *TF* stands for Term Frequency is one of the earliest ranking methods. One potential issue that can arise from using this method out of the box is the fact that it has a limitation of considering the same importance for all terms[8]. Fig. 1 shows the typical formula for TF [8].

$$score(D,Q) = \sum_{i=1}^{|Q|} f(q_i, D)$$

Fig. 1: A typical formula for TF[8].

Then followed by *IDF*(short for Inverse Document Frequency) which proposed to overcome the drawback of TF method[8]. That is, IDF gives higher weight to the terms which occur rarely in the collection, in other words, it raises the importance of the terms w.r.t the collection [8]. The limitation of this technique is that it can give the same score for all documents contain the same term. *TF-IDF* is a combination of the aforementioned methods above(see Fig.2) [8].

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t.$$

Fig. 2: A typical formula for TF-IDF[8].

Another ranking method to consider is *BM25* which extends the idea of TF-IDF, that is, it states the principle that less frequent terms in the collection can have higher weight compared to the common one[8]. This approach can be categorized into three components[8]:

1. saturation component.
2. within-document component.
3. within-query component.

$$\sum_{i \in Q} \log \frac{(r_i + 0.5)/(R - r_i + 0.5)}{(n_i - r_i + 0.5)/(N - n_i - R + r_i + 0.5)} \cdot \frac{(k_1 + 1)f_i}{K + f_i} \cdot \frac{(k_2 + 1)qf_i}{k_2 + qf_i}$$

Fig. 3: A typical formula for BM25[8].

To understand BM25, Fig. 3 shows the typical BM25 formula [8].

We also consider three types of fusion algorithms(see Fig-4) which is aiming at combining a set of original runs to obtain a final run with the goal of having higher ranking quality.

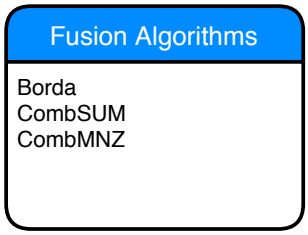

Fig. 4: considered rank-fusion methods

The objective of *Borda* method is to sum up the difference in rank position for each item in the list[8]. *CombSUM* aims at summing up the scores for each item in the list [8]. *CombMNZ*, is experimentally a reliable and an effective method[8].

1.1 Discussion

We first of all have created a parser to extract the queries and we have calling that parser along with BM25 and TF-IDF functions for both title and Boolean queries, next, files are generated for both data-sets for further processing. Evaluation measures along with statistical analysis are also applied on those resulted files(discussed in section 3), we evaluate the retrieval methods with respect to the mean average precision (MAP) using trec-eval; we sat the cut-off value to (-M ,i.e. the maximum number of documents per topic) to the number of documents to be re-ranked for each of the queries. We then applied the normalization in order to normalize the scores for fusion process. Three types of fusion have been used and applied on both data-sets (CLEF 2017 and 2018 eHealth).

All of the previously mentioned methods are evaluated using Evaluation measures(set-based measures and rank-based measures)in order to evaluate the performance, and the findings as well as the comparisons between these methods are demonstrated in section 3.

The tables below shows the average value for each run according to three evaluation measures(MAP,nDCG,P-10). We applied trec-eval tool using -q parameter in order to compare results across each individual query. In order to determine if any of two retrieval algorithms are significantly different(one is better than the other), statistical analysis is in urgent need[8]. Further insights of such evaluation results about effectiveness analysis can be seen in the Figures below. The numbers presented in the tables were obtained by using trec-eval tool.

In terms of the rank-fusion methods, we have tried all combinations in the fusion process, evaluating using MAP and nDCG. It turns out that CombSUM and CombMNZ achieved better results compared to the other fusion method and thus, it is the most effective fusion methods in this experiment.

To make relevance scores comparable, the normalization formula proposed by[7] is used:

$$Min\_Max = \frac{old\_sim - minimum\_sim}{maximum\_sim - minimum\_sim}$$

Fig. 5: MinMax furmula[7]

where old-sim is the relevance score that has to be normalized and minimum-sim, maximum-sim is respectively the minimum and the maximum relevance score in a specific run.

Then, we investigated which the effective method by means of comparing the estimated values of MAP and ndcg yielded from the evaluation process for all methods as reported in the output below.

In this section, we considered TF-IDF and BM25 methods as ranking functions and three types of rank fusion methods, namely, CombSUM,, CombMNZ, which is also score-based fusion algorithm, and Borda, which is voting and rank-based approach[8]. We also apply query reduction method on the Boolean queries.

1.2 Evaluation: Results

Finding a good ranking method

- **Title Queries**
  It can be clearly seen from Table 1 that BM25 method outperforms TF-IDF. Both methods have the same P@10. Further insights for the distribution can be seen in the following figures. As shown in the figures , BM25 seems to be better than TF-IDF.

  By comparing those methods in both 2017 and 2018 data sets, both methods are providing higher scores in 2018 data set compered to 2017 one.

	MAP	nDCG	Rprec	P_5
BM25	0.1629	0.5393	0.1843	0.2667
TF_IDF	0.1625	0.5389	0.1843	0.2667

Table 1: Evaluation measures for 2017 Testing Title Queries

  Utilizing Table 3 statistics, we can infer that TF-IDF is somehow outperforms BM25 and therefore, identifying which the best can be somewhat complex. According to P@5 measure is the same in both methods. Further insights could be seen in the figures below.

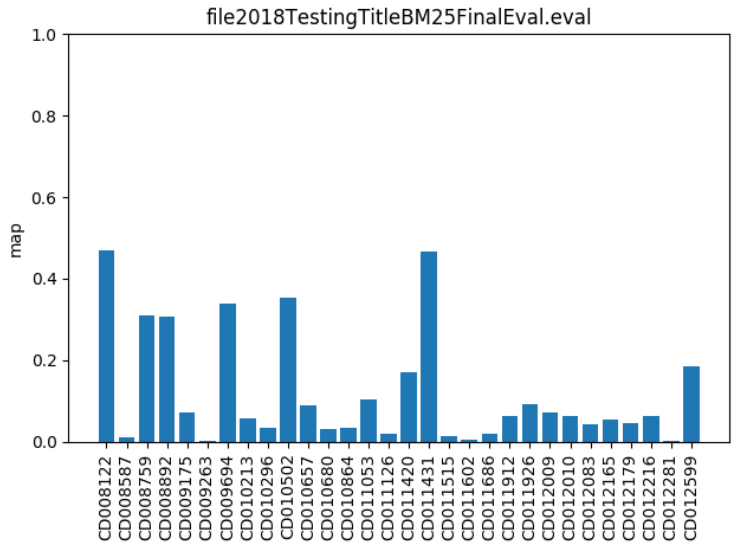

Fig. 6: BM25.

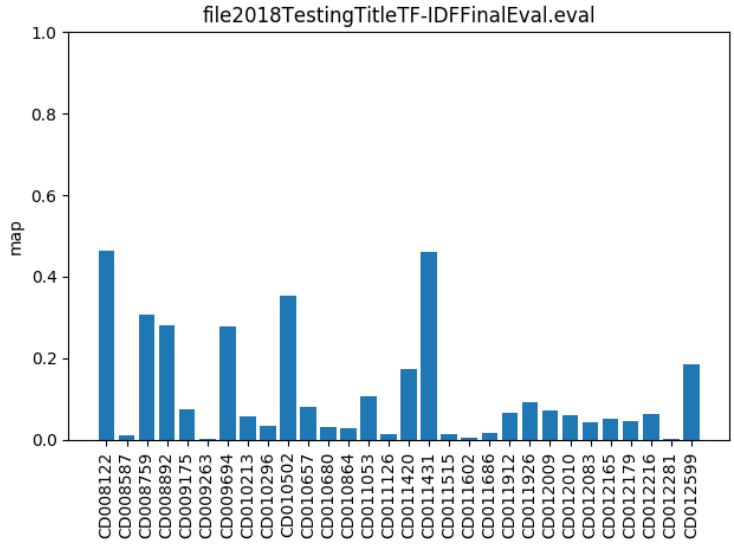

Fig. 7: TF-IDF.

In the same manner, we can make the same considerations between both methods in training datasets.

	MAP	nDCG	Rprec	P_5
BM25	0.1462	0.5415	0.1704	0.2524
TF_IDF	0.1458	0.5412	0.1692	0.2619

Table 2: Evaluation measures for 2018 Training Title Queries

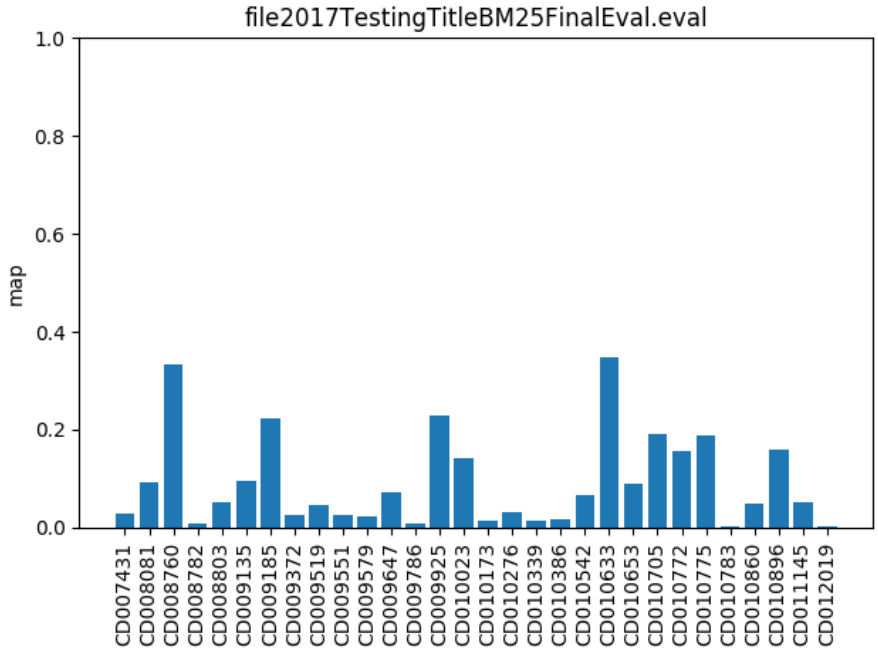

Fig. 8: BM25.

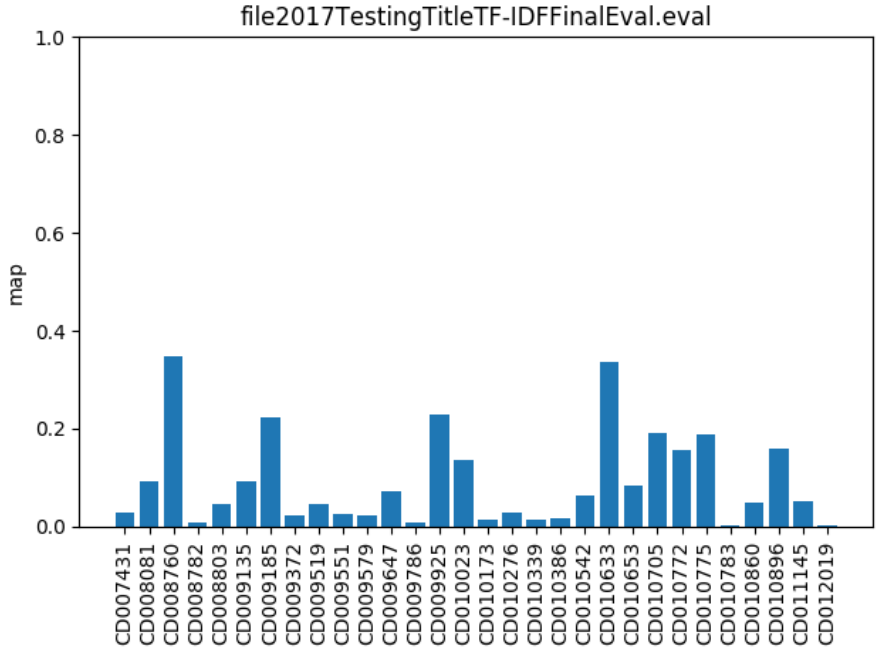

Fig. 9: TF-IDF.

By looking at Table 6, as previously mentioned, BM25 has higher scores and thus, it is a robust and an effective ranking method. Of course it shows that both of them have the same P@10. Further insights can be seen in [Fig. 10 and Fig. 11].

	MAP	nDCG	Rprec	P_5
BM25	0.1700	0.5933	0.1959	0.2200
TF_IDF	0.1705	0.5936	0.1976	0.2200

Table 3: Evaluation measures for 2018 Testing Title Queries

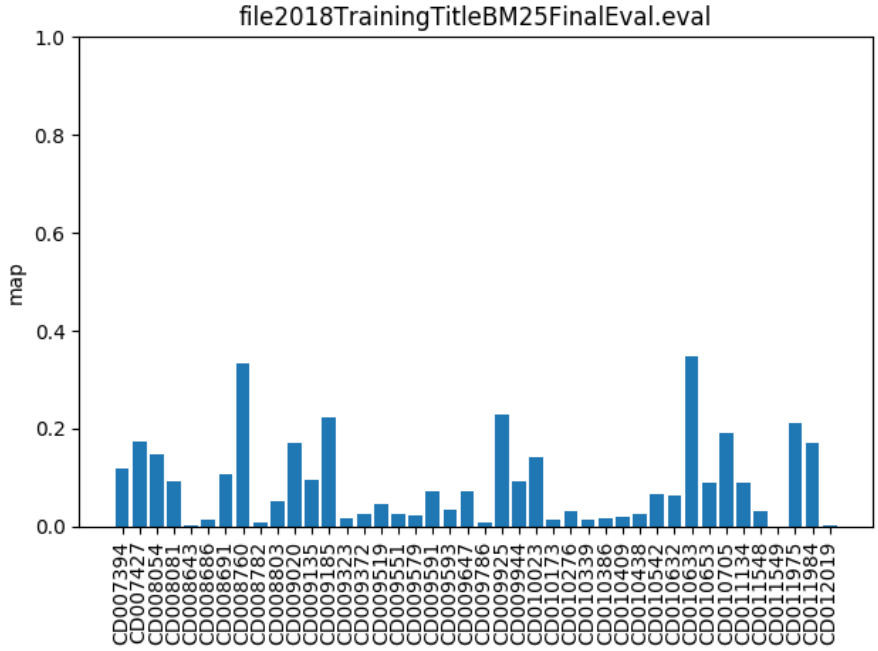

Fig. 10: BM25.

Here in this table , it turns out that BM25 is better than TF-IDF.
By observing Table 3 , BM25 does outperform TF-IDF according to these specified measures, map,Rprec,ndcg and p@10 respectively. Further insights can be observed from the following plots.
Now, we can observe training 2017 dataset with titles plots.

	MAP	nDCG	Rprec	P_5
BM25	0.1138	0.4817	0.1513	0.2200
TF_IDF	0.1134	0.4813	0.1514	0.2400

Table 4: Evaluation measures for 2017 Training Title Queries

It can be seen from the following tables that after fuse existing runs with the provided runs, we obtained less results. Also we can infer from gain/loss

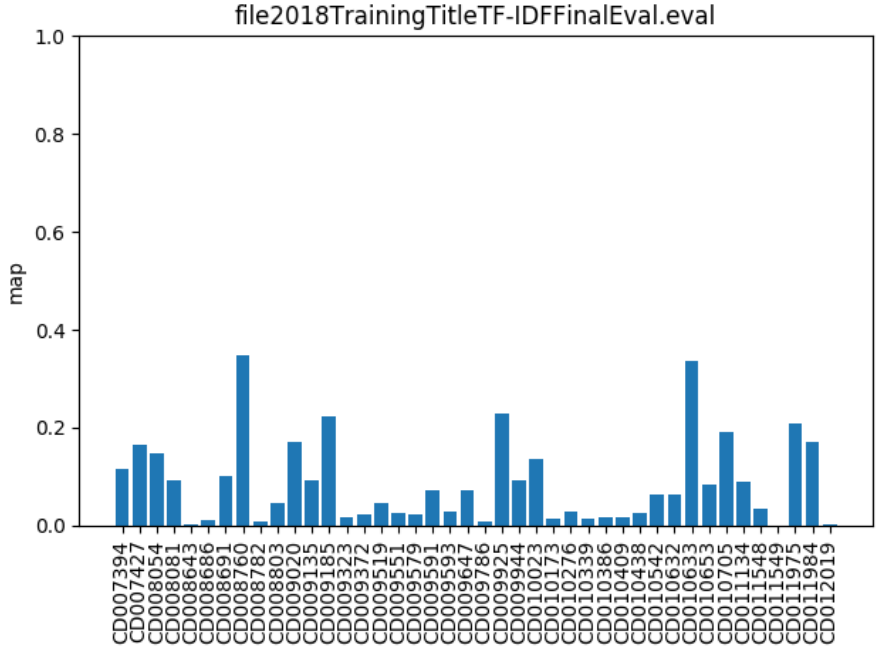

Fig. 11: TF-IDF.

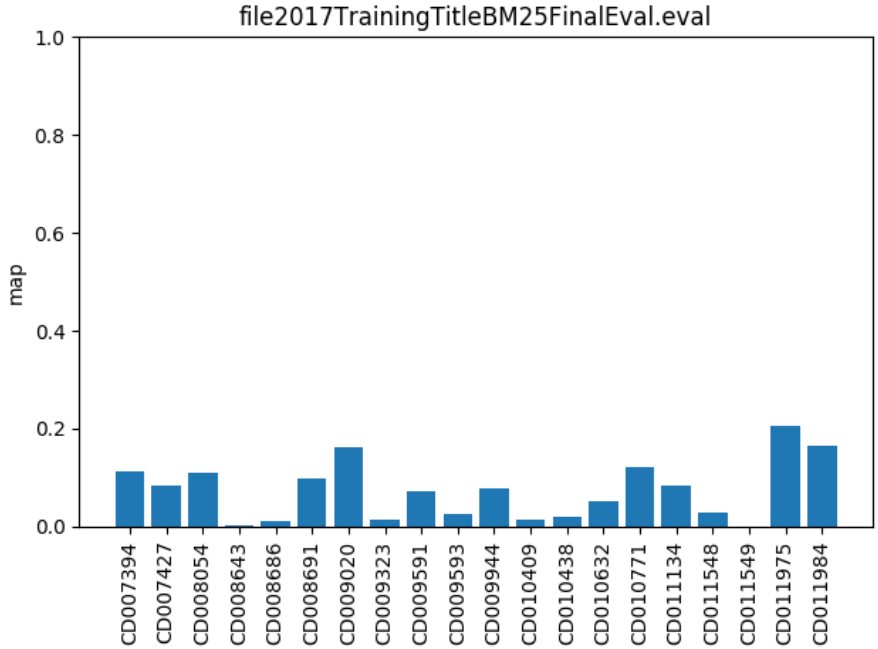

Fig. 12: BM25.

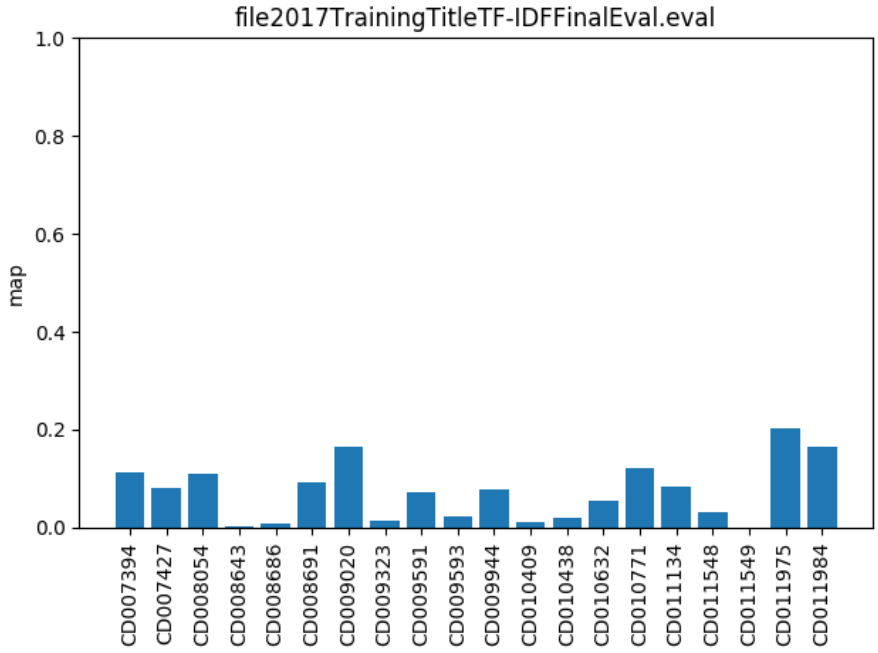

Fig. 13: TF-IDF.

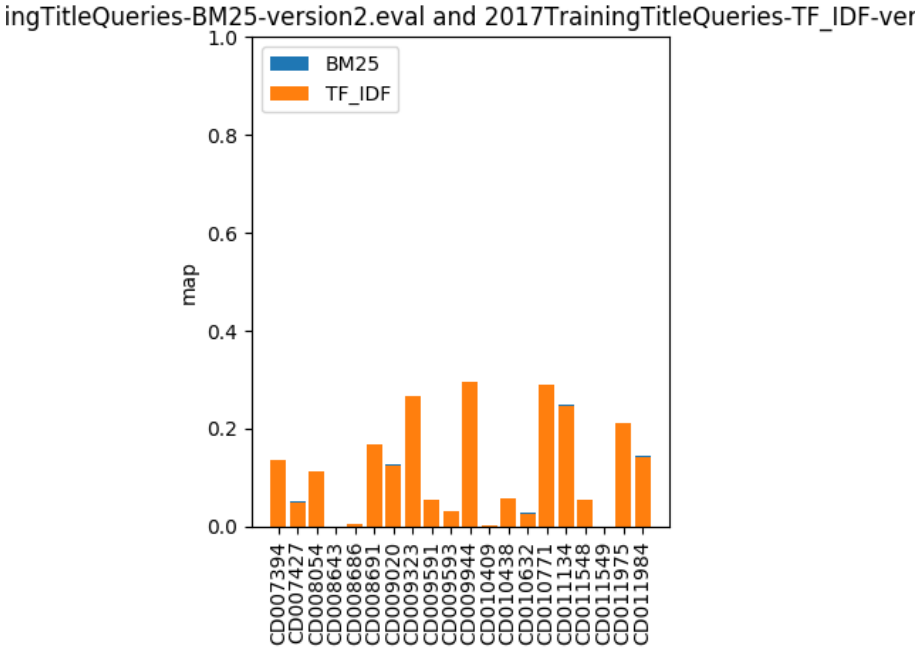

Fig. 14: BM25.

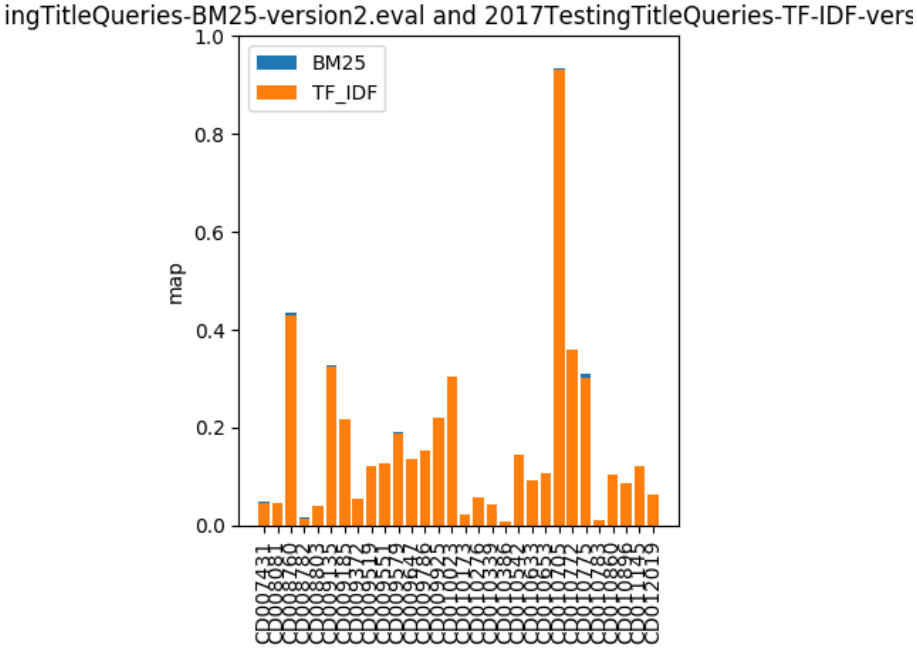

Fig. 15: TF-IDF.

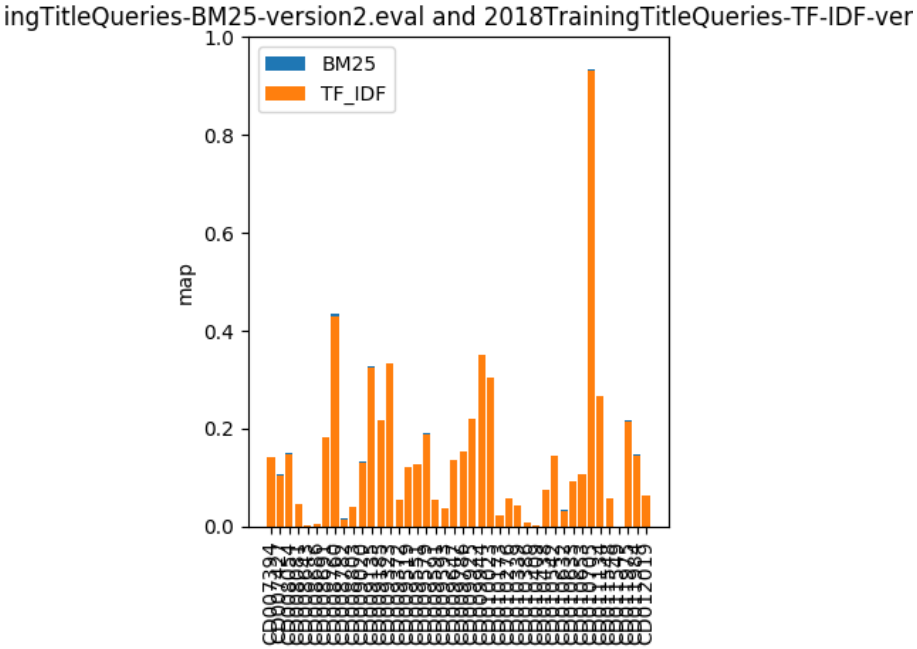

Fig. 16: BM25.

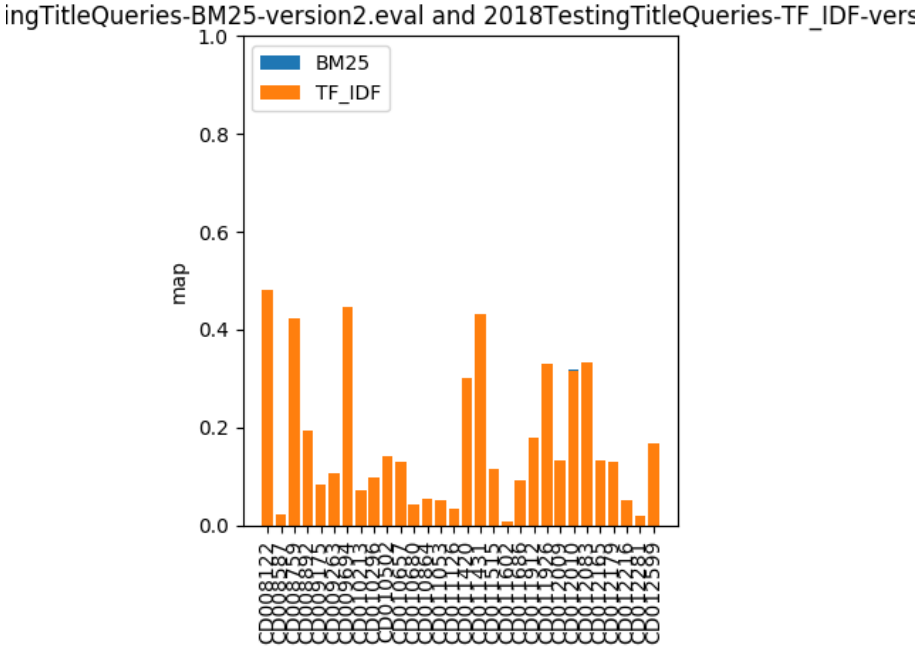

Fig. 17: TF-IDF.

	MAP	Rprec	nDCG	P@10
Borda	0.1175	0.1251	0.5284	0.1633
CombSUM	0.1181	0.1242	0.5287	0.1700
CombMNZ	0.1181	0.1242	0.5287	0.1700

Table 5: Evaluation measures for the fusion algorithms.

	MAP	***Rprec***
Borda	0.1629	0.1843
CombSUM	0.1628	0.1843
CombMNZ	0.1628	0.1843

Table 6: Fusion Results for 2017 Testing Title Queries data-set.

	MAP	***Rprec***
Borda	0.1702	0.1970
CombSUM	0.1703	0.1973
CombMNZ	0.1703	0.1973

Table 7: Fusion Results for 2018 Testing Title Queries data-set.

figures that there is no much difference among the methods after and before doing the fusion. Both CombSUM and CombMNZ obtained same results. We can apply the same considerations on 2018 Testing data-set.

	MAP	***Rprec***
Borda	0.1607	0.1767
CombSUM	0.1378	0.1686
CombMNZ	0.1378	0.1686

Table 8: Fusion Results for 2018 Testing Title Queries data-set with all other runs.

	MAP	***Rprec***
Borda	0.1333	0.1521
CombSUM	0.1489	0.1627
CombMNZ	0.1489	0.1627

Table 9: Fusion Results for 2017 Testing Title Queries data-set with all other runs.

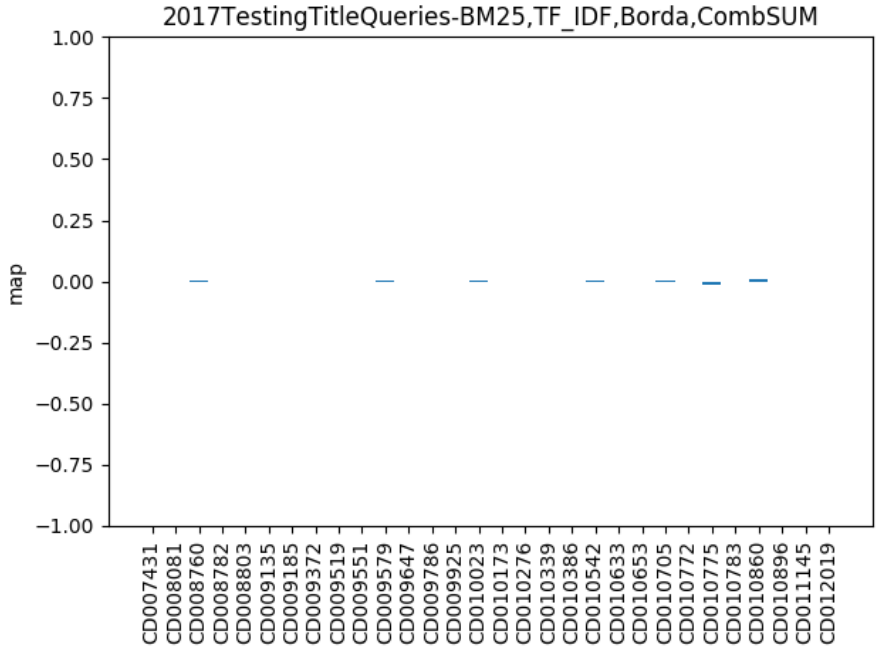

Fig. 18: Gain/Loss

MAP		
	2017 Testing Data set	2018 Testing Data set
BM25	0.0285	0.4684
TF-IDF	0.0293	0.4625
Borda	0.1175	
CombSUM	0.1181	
CombMNZ	0.1181	

Table 10: Summary MAP measure for both datasets.

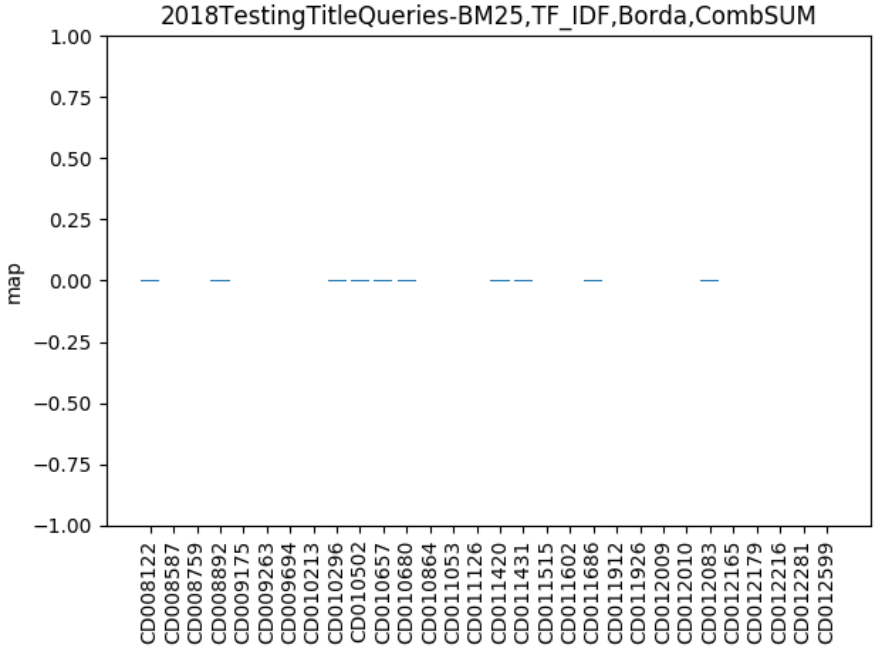

Fig. 19: Gain/Loss

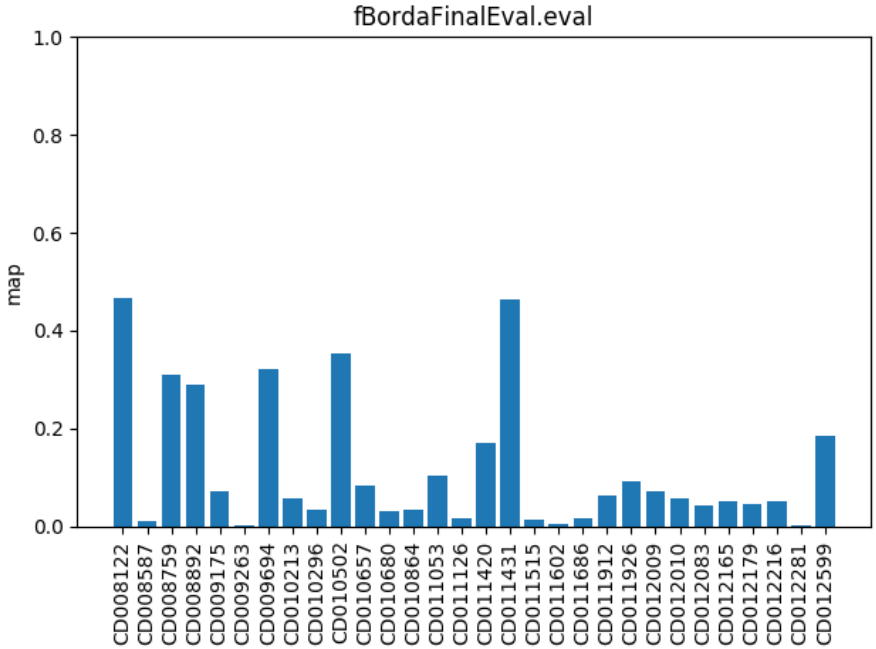

Fig. 20: Borda

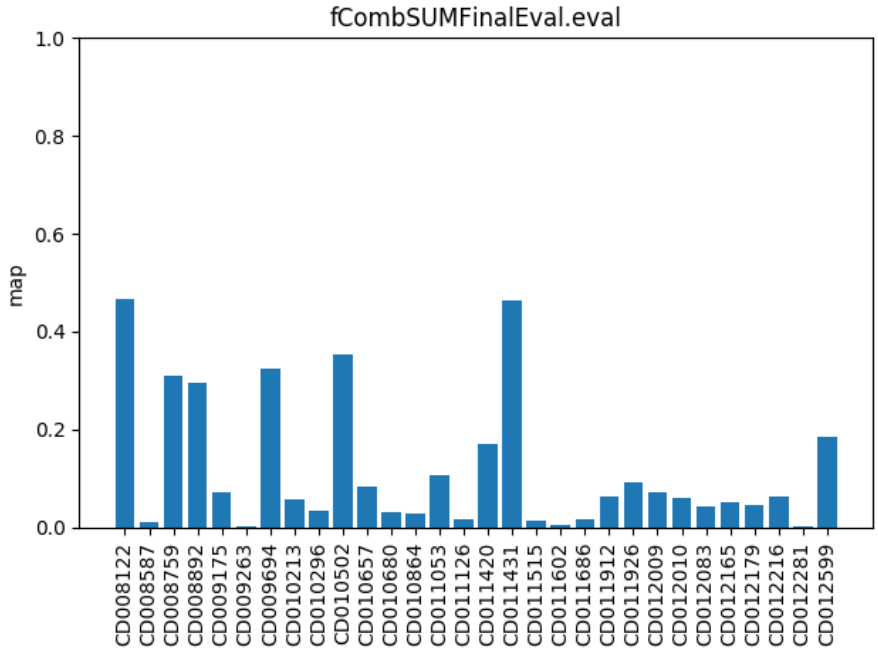

Fig. 21: CombSUM

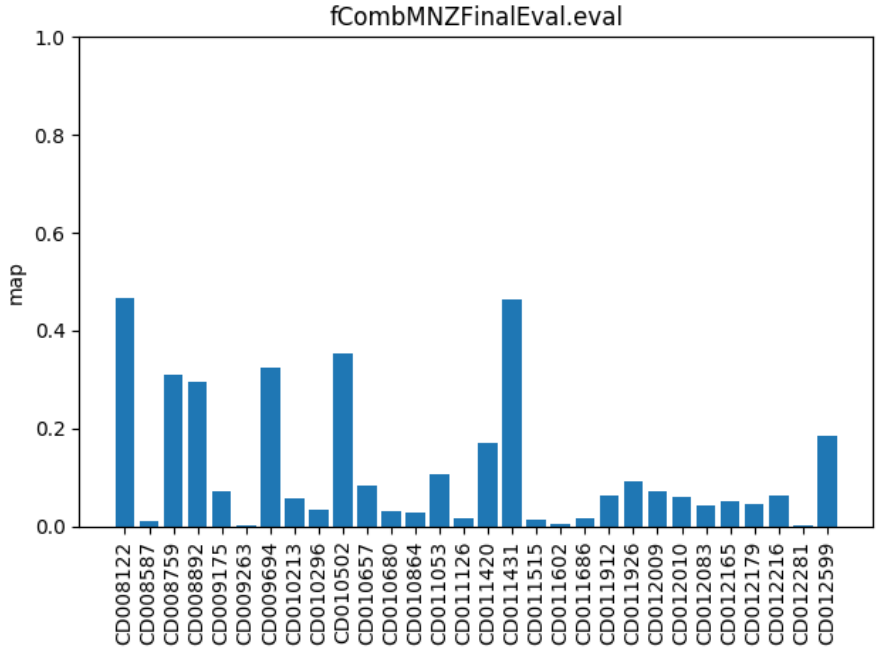

Fig. 22: CombMNZ

Rprec		
	2017 Testing Data set	2018 Testing Data set
BM25	0.1065	0.1263
TF-IDF	0.1047	0.1239
Borda	0.1175	
CombSUM	0.1181	
CombMNZ	0.1181	

Table 11: Summary Rprec measure for both datasets.

- Gain/Loss.
  Gain/loss for both data-sets can be seen in the following figures.

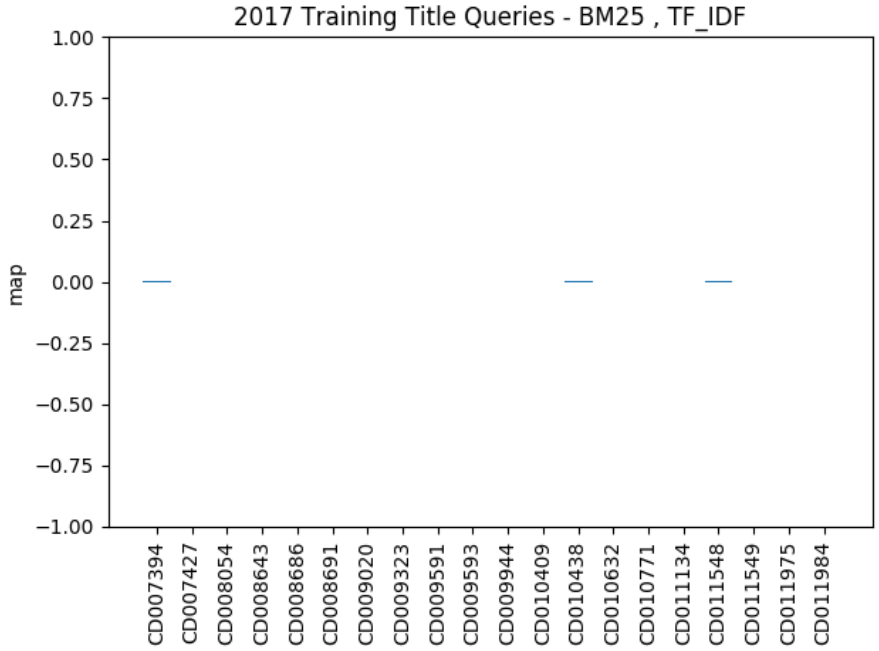

Fig. 23: Gain/Loss

  Gain/Loss figures indicate that there is no much difference in the gain/loss among BM25 and TF-IDF.

From the box plots, at a first glance, it seems that 2018 data-sets performs better than 2017 data-sets.

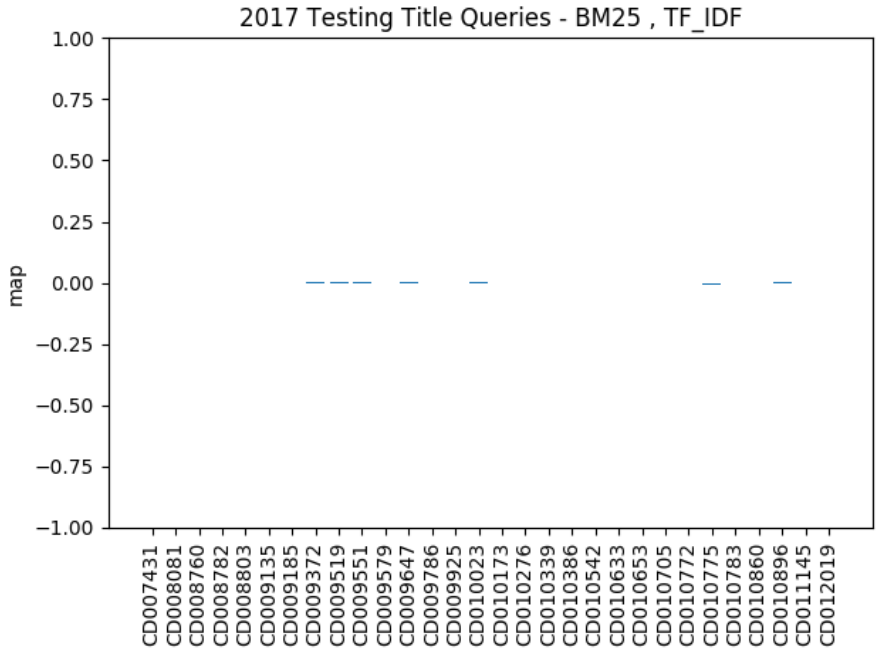

Fig. 24: Gain/Loss

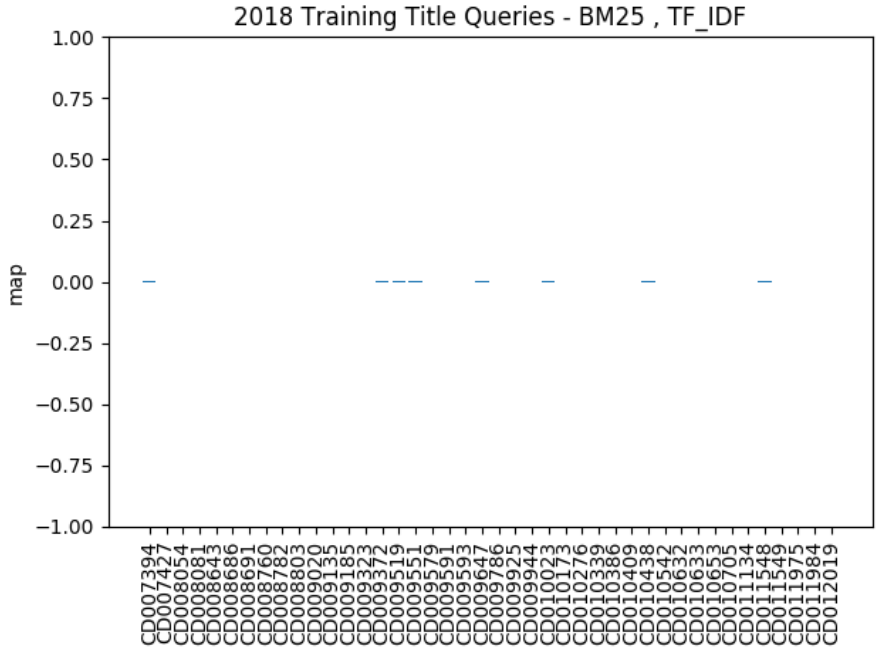

Fig. 25: Gain/Loss

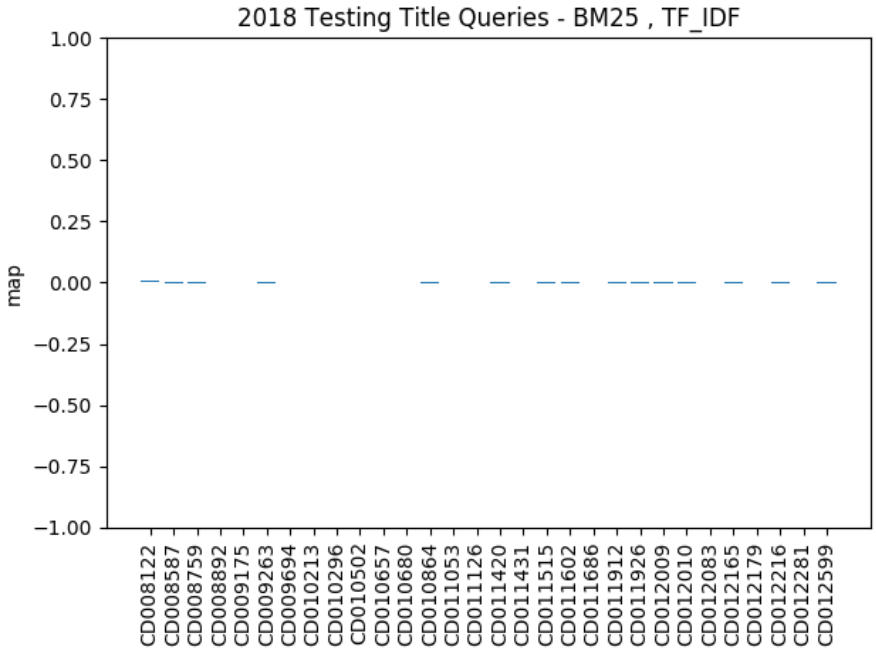

Fig. 26: Gain/Loss

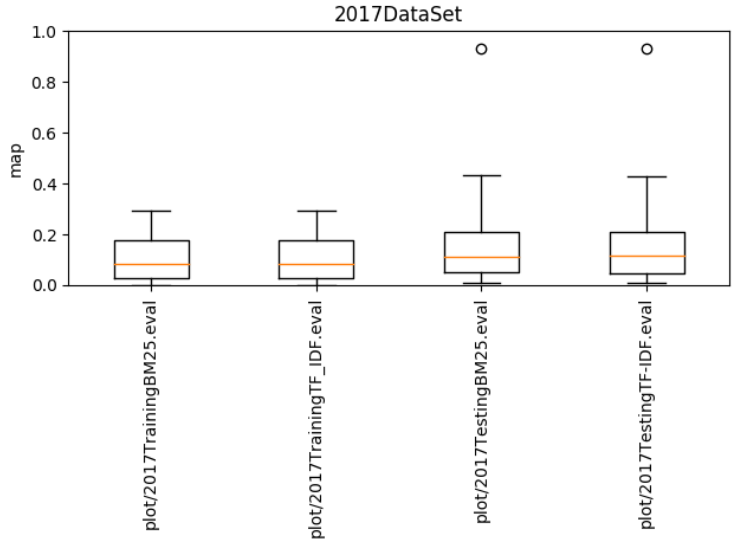

Fig. 27: BoxPlot

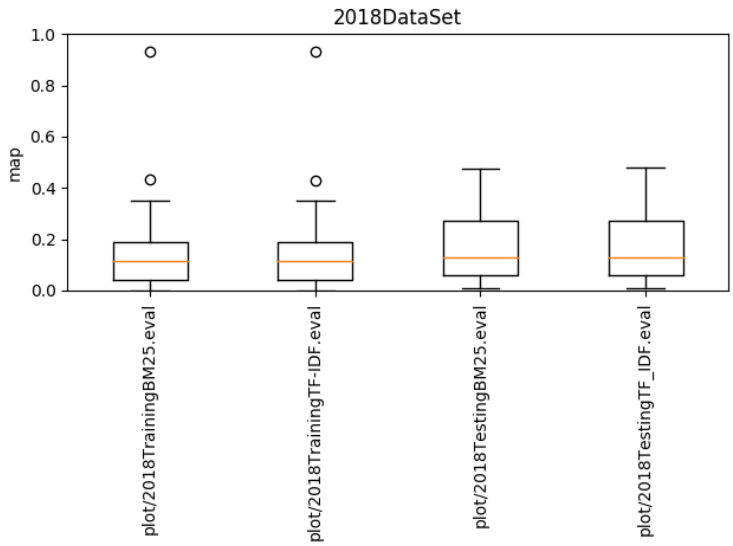

Fig. 28: BoxPlot

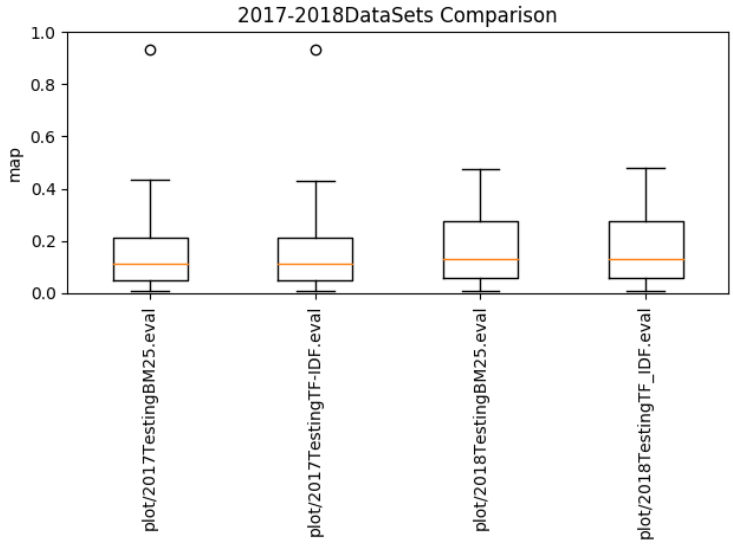

Fig. 29: BoxPlot-2018-2017TestingDataSets

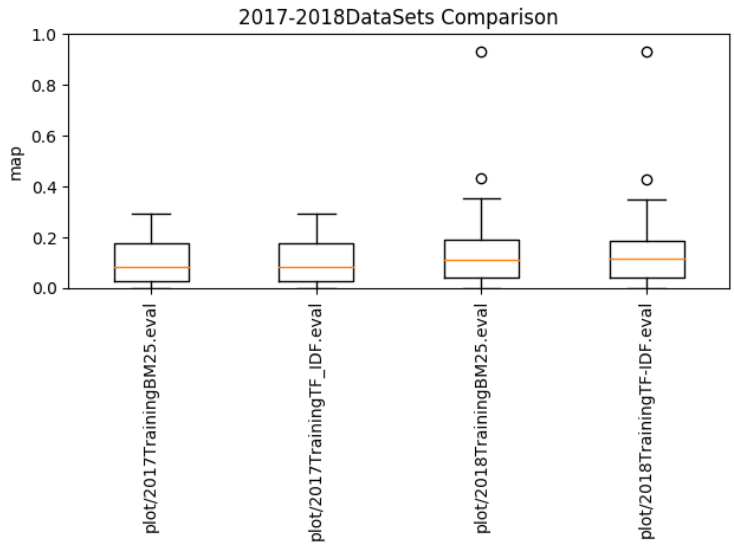

Fig. 30: BoxPlot-2018-2017TrainingDataSets

– **Boolean Queries**

In this context, the queries are boolean expressions and the system will retrieve only the documents that satisfy that query. As it can be observed, BM25 still more effective than TF-IDF.

	MAP	nDCG	Rprec	P_10
BM25	0.0833	0.4522	0.1101	0.1050
TF-IDF	0.0816	0.4487	0.1044	0.1000

Table 12: Evaluation measures for 2017 Training Boolean Queries.

	MAP	nDCG	Rprec	P_10
BM25	0.1071	0.4763	0.1182	0.1367
TF-IDF	0.1054	0.4727	0.1170	0.1333

Table 13: Evaluation measures for 2017 Testing Boolean Queries.

	MAP	nDCG	Rprec	P_10
BM25	0.1316	0.5325	0.1411	0.1733
TF-IDF	0.1299	0.5297	0.1354	0.1667

Table 14: Evaluation measures for 2018 Testing Boolean Queries.

	MAP	nDCG	Rprec	P_10
BM25	0.0949	0.4921	0.1128	0.1167
TF-IDF	0.0920	0.4865	0.1084	0.1095

Table 15: Evaluation measures for 2018 Training Boolean Queries.

Further insights between BM25 and TF-IDF can be observed form the following figures. We plot the difference between both methods using "map"

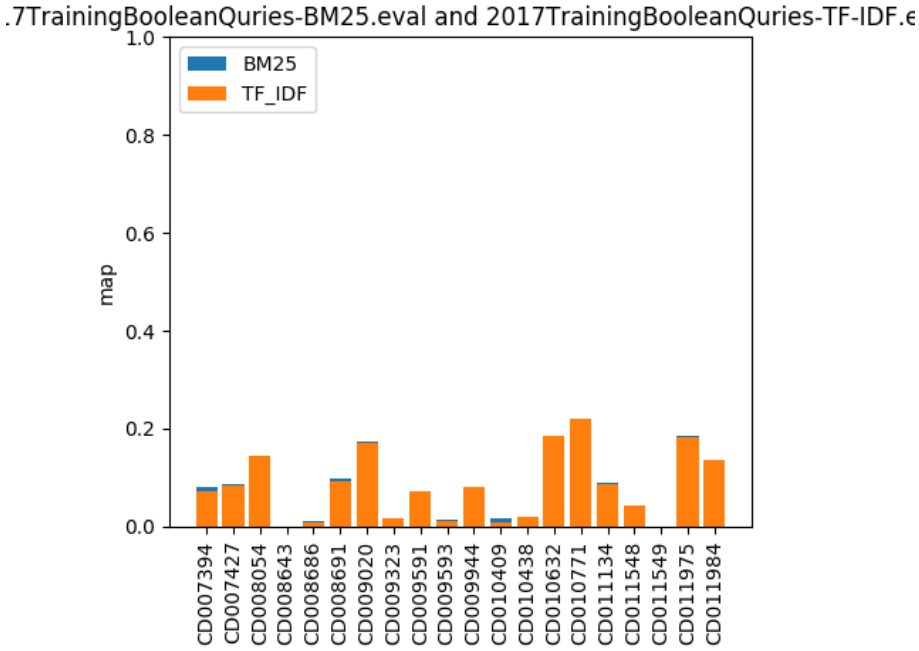

Fig. 31: 2017Training-Boolean Queries-BM25 and TF-IDF

measure and as it can be seen from the plots that BM25 outperforms TF-IDF. In terms of the fusion methods, by looking at Table 17, we can infer that both CombSUM and CombMNZ are having the same effect (results) whereas Borda obtained small results compared to the aforementioned methods. Generally speaking, both title queries and boolean queries come to the same conclusion where CombSUM and CombMNZ have yielded comparable values as opposed to Borda, and BM25, TF-IDF values are slightly different from each other, where BM25 surpasses TF-IDF as discussed above.

	MAP	***Rprec***
Borda	0.1305	0.1367
CombSUM	0.1308	0.1380
CombMNZ	0.1308	0.1380

Table 16: Fusion Results for 2018 Testing Boolean Queries data-set.

	MAP	***Rprec***
Borda	0.1062	0.1171
CombSUM	0.1071	0.1172
CombMNZ	0.1071	0.1172

Table 17: Fusion Results for 2017 Testing Boolean Queries data-set.

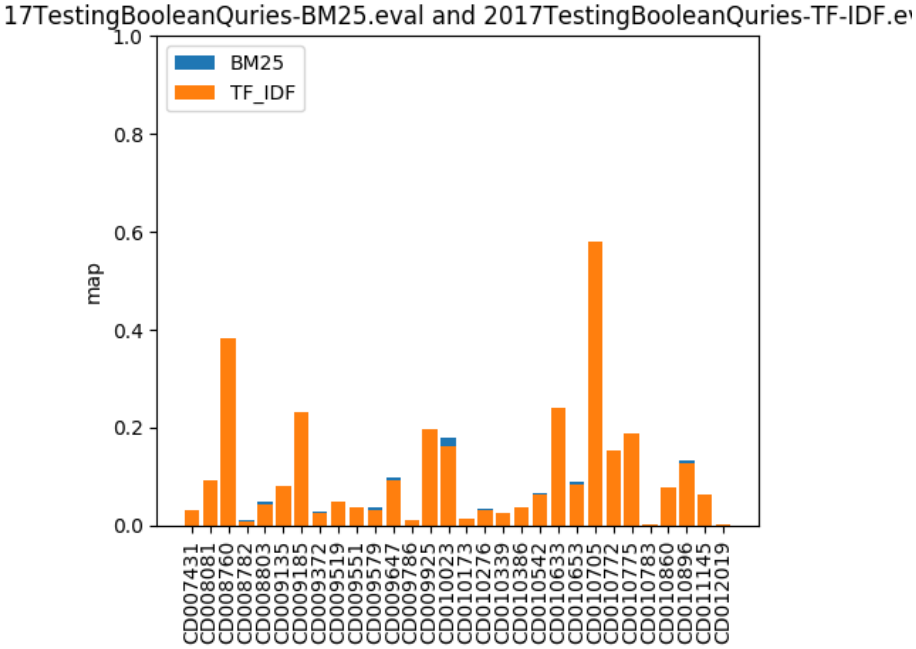

Fig. 32: 2017Testing-Boolean Queries-BM25 and TF-IDF

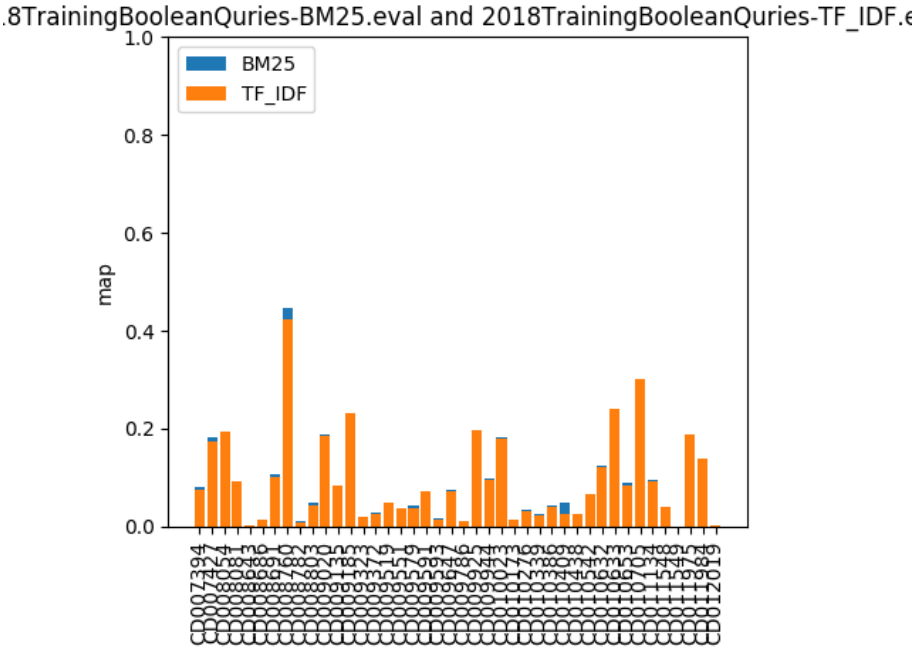

Fig. 33: 2018Training-Boolean Queries-BM25 and TF-IDF

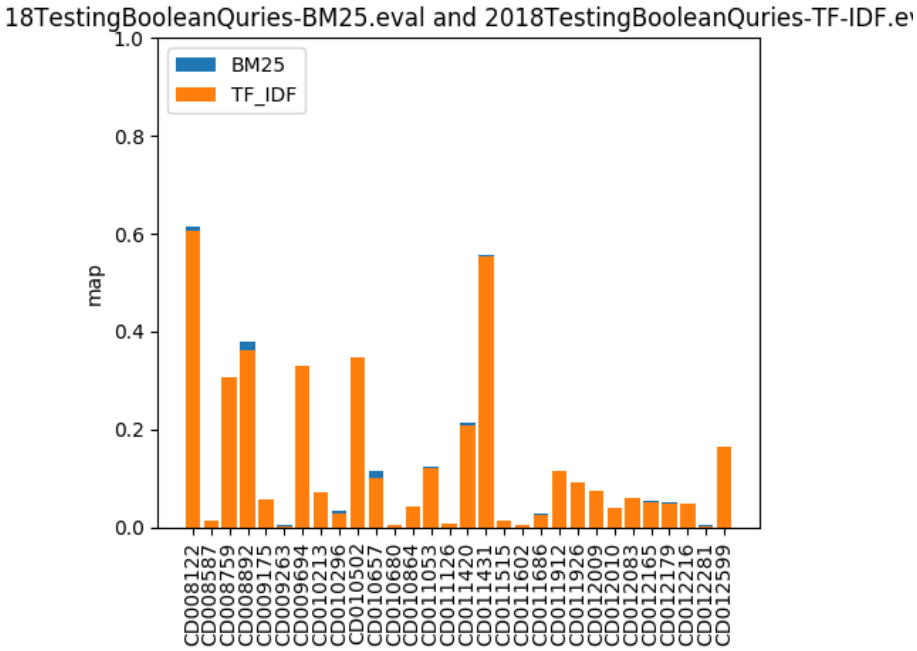

Fig. 34: 2018Testing-Boolean Queries-BM25 and TF-IDF

- Gain/Loss.
  By looking at the following figures, we can infer that there is no much difference in the gain/loss between both methods.

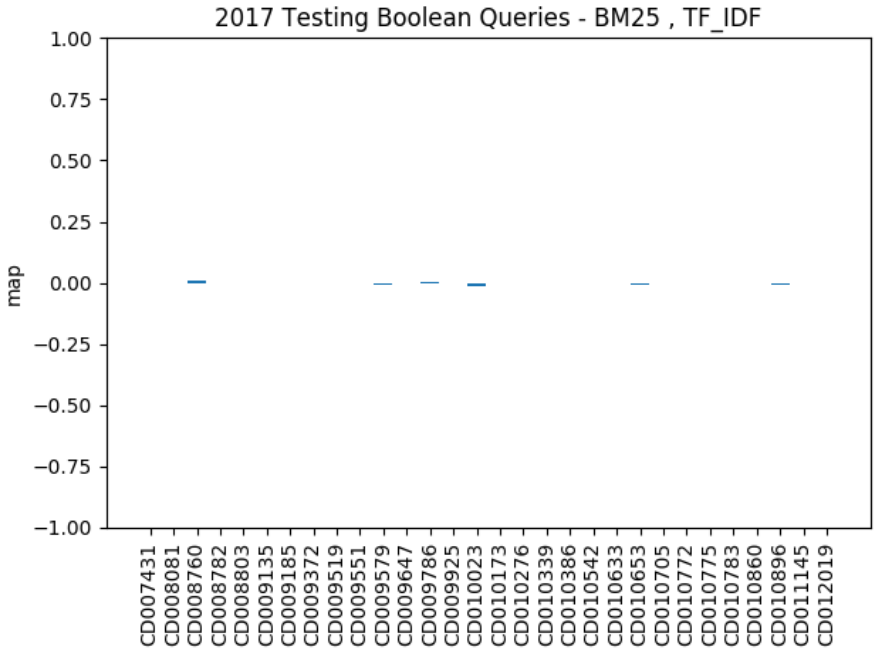

Fig. 35: Gain/Loss

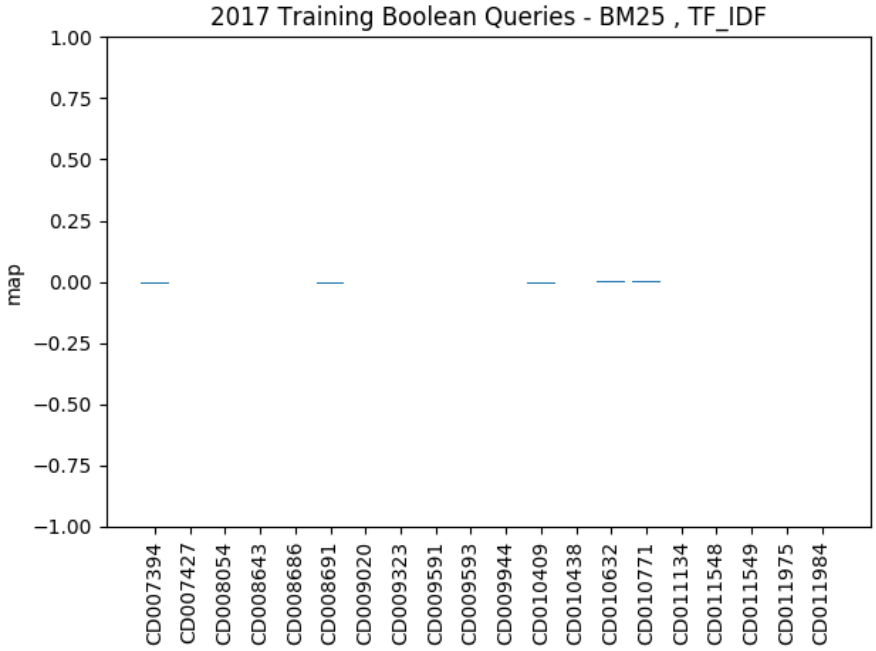

Fig. 36: Gain/Loss

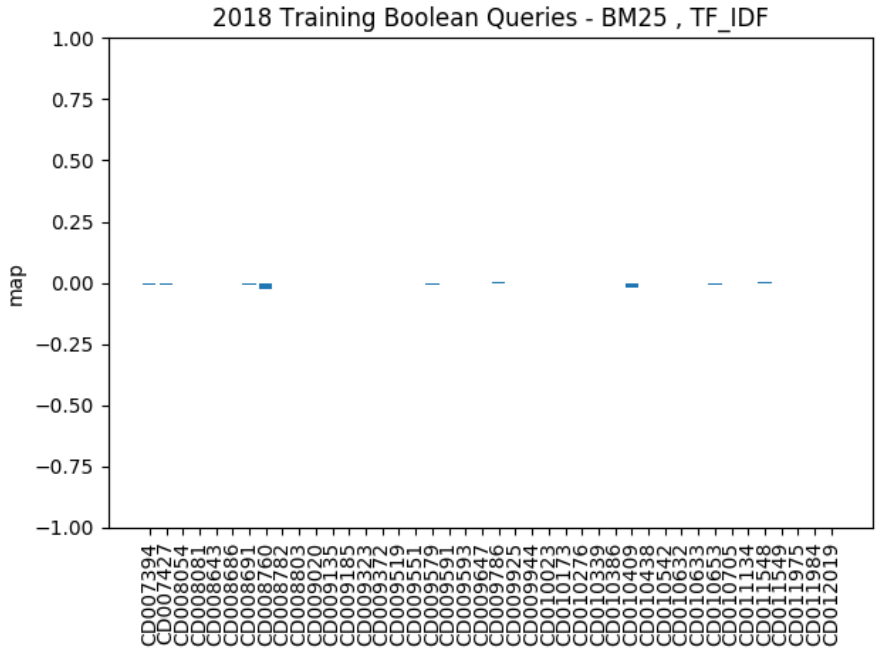

Fig. 37: Gain/Loss

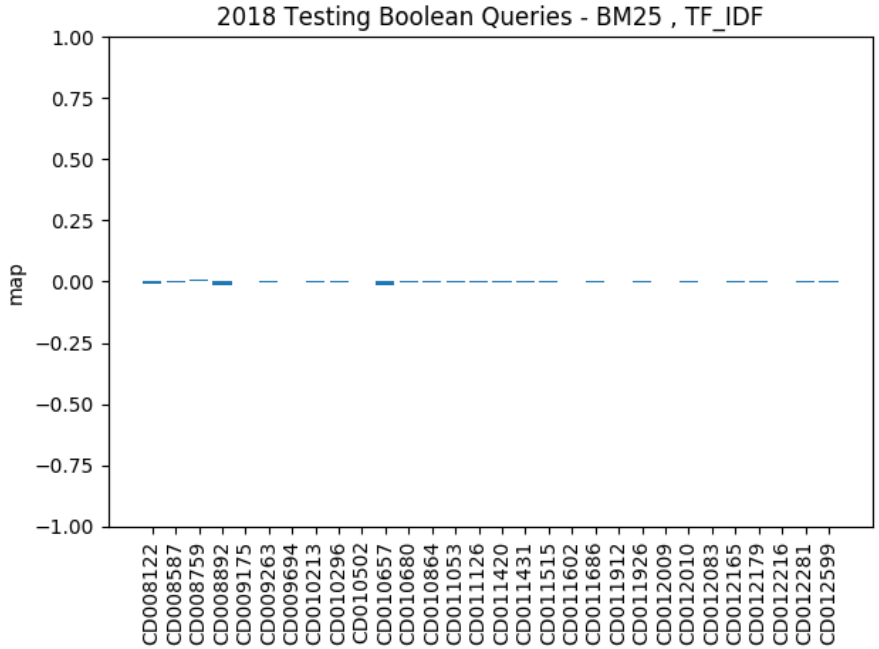

Fig. 38: Gain/Loss

– **Query Reduction**

In this section we applied the IDF-r on queries datasets and we limited the number of terms to 3. Surprisingly, the obtained results are not better than that of the original queries. In other words, term-reduction process is not adding any value to our retrieval. Further insights can be observed from the tables and figures. One thing should be noted is that both methods have somewhat same results.

	MAP	nDCG	Rprec	P_10
BM25	0.0531	0.4016	0.0496	0.0433
TF_IDF	0.0532	0.4017	0.0496	0.0433

Table 18: 2017 Testing Query-Reduction.

	MAP	nDCG	Rprec	P_10
BM25	0.0476	0.4006	0.0606	0.0450
TF_IDF	0.0476	0.4007	0.0610	0.0450

Table 19: 2017 Training Query-Reduction.

	MAP	nDCG	Rprec	P_10
BM25	0.0537	0.4290	0.0545	0.0524
TF_IDF	0.0537	0.4291	0.0547	0.0524

Table 20: 2018 Training Query-Reduction.

	MAP	nDCG	Rprec	P_10
BM25	0.0625	0.4626	0.0629	0.0633
TF_IDF	0.0625	0.4626	0.0630	0.0633

Table 21: 2018 Testing Query-Reduction.

It can be seen that after doing the term-reduction process, TF-IDF somewhat surpasses BM25 in 2017 testing data.
By comparing the map and other evaluation measures between the data-set before and after the term-reduction process, we infer that the values of these measures are better without doing the reduction.

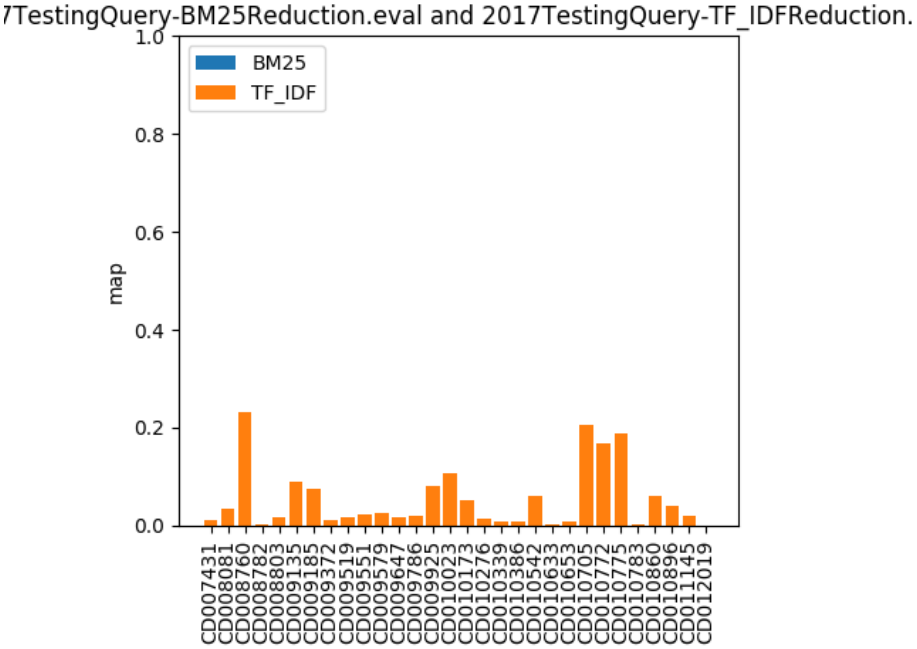

Fig. 39: 2017Testing-QueriesReduction-BM25 and TF-IDF

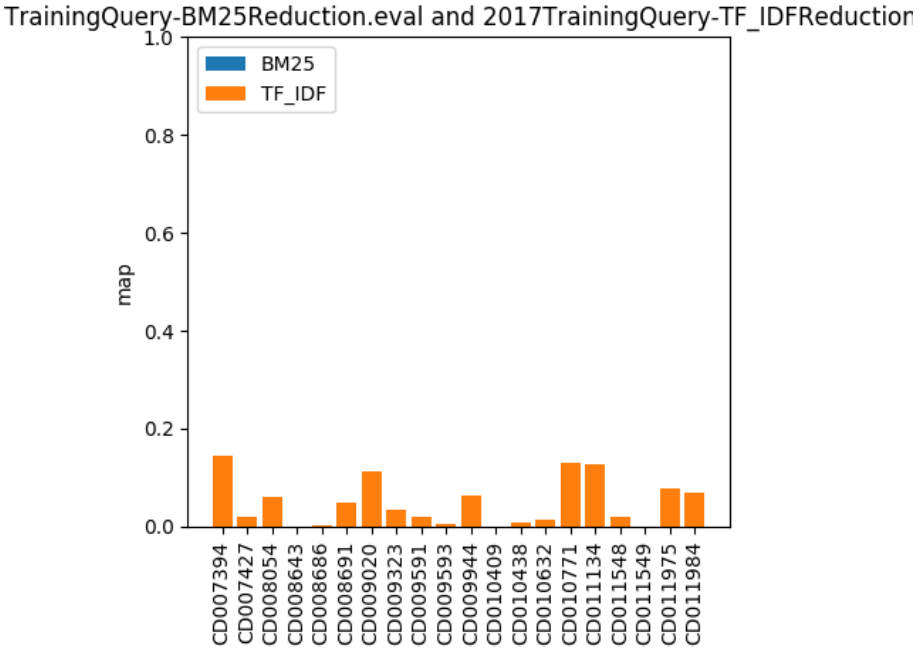

Fig. 40: 2017Training-QueriesReduction-BM25 and TF-IDF

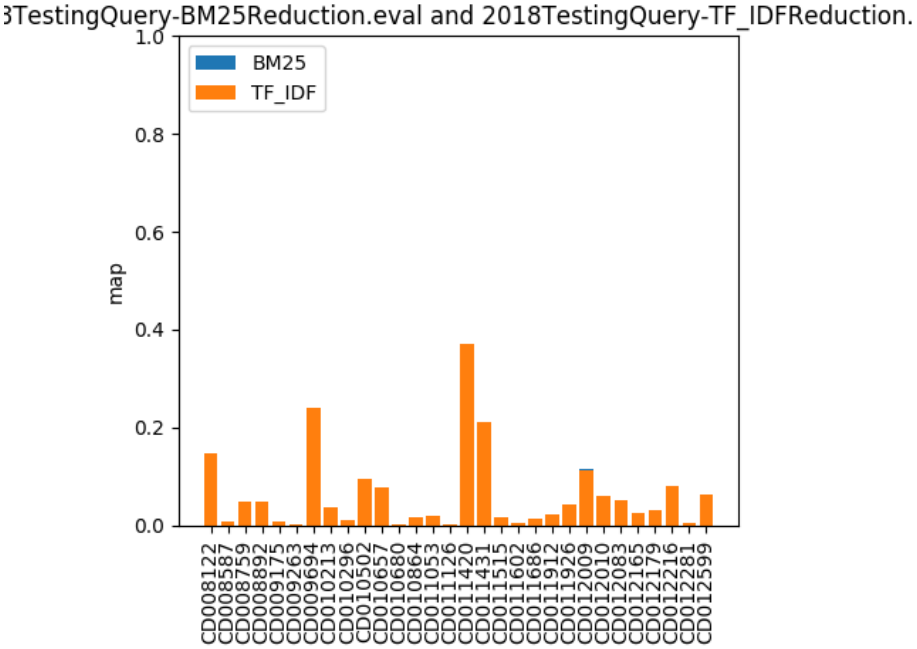

Fig. 41: 2018Testing-QueriesReduction-BM25 and TF-IDF

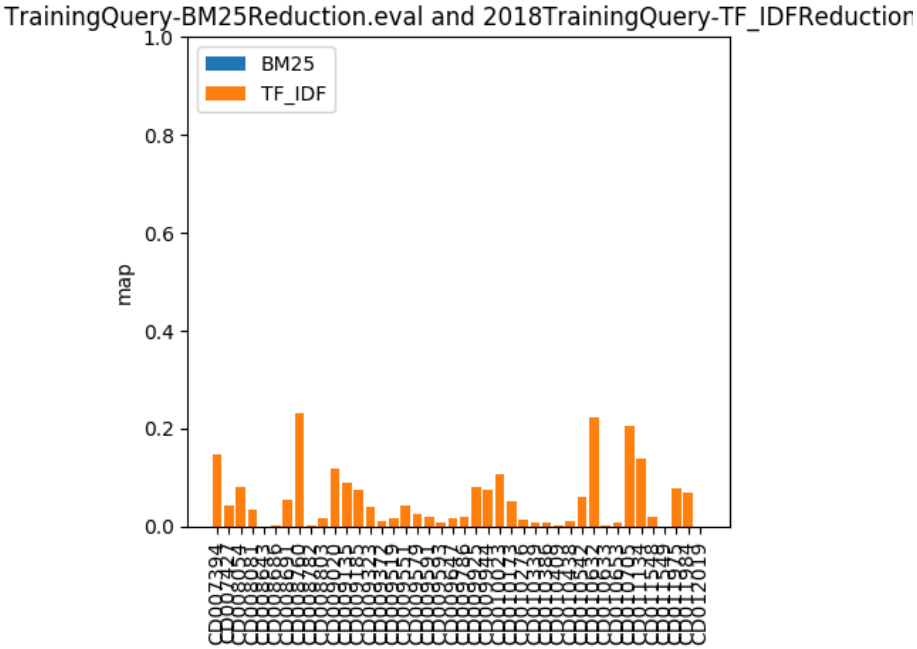

Fig. 42: 2018Training-QueriesReduction-BM25 and TF-IDF

– **BM25 Parameter Tuning**

We have tried many values, each of which has a different value. According to the "map" measure, we come to the conclusion that BM25 with the value of b = 0.25 is the best.

	map	ndcg
0.25	0.1194	0.4916
0.5	0.1177	0.4880
1.0	0.1111	0.4781
1.25	0.1031	0.4688

Table 22: 2017 Training Data-set.

	map	ndcg
0.25	0.1718	0.5487
0.5	0.1710	0.5488
1.0	0.1508	0.5246
1.25	0.1357	0.5074

Table 23: 2017 Testing Data-set.

By applying the chosen value, which is 0.25, to the testing set, we find that it achieves a better result. Therefore, BM25 with value of 0.25 achieves better.

	map	ndcg
0.25	0.1555	0.5531
0.5	0.1538	0.5509
1.0	0.1374	0.5314
1.25	0.1225	0.5145

Table 24: 2018 Training Data-set.

	map	ndcg
0.25	0.1696	0.5966
0.5	0.1688	0.5935
1.0	0.1679	0.5889
1.25	0.1539	0.5736

Table 25: 2018 Testing Data-set.

1.3 Statistical significance test

Statistical significance test is commonly used to evaluate the performance of two systems. Recalling that `( from the lecture)` "the significance test is represented as a number between 0 and 1, where smaller p (when p less than 0.05 or 0.01) indicates stronger evidence to reject the null hypothesis,we can infer that there is a strong statistical significance.

P-value			
	MAP	nDCG	P_5
2017 Training Title Queries	0.17392392246807886	0.4305460288403863	0.16254999902972722
2017 Testing Title Queries	0.2390969474748348	0.3001400326453135	NaN
2018 Training Title Queries	0.10731193007559892	0.33755901667724	0.1597858784099142
2018 Testing Title Queries	0.12139847744806853	0.2962209323980247	NaN

Table 26: P-value for both data-sets Title-Queries (BM25,TF_IDF)

According to the observations (see Table 26), it turns out that there is no a difference between the methods since p-value is larger than 0.05, 0.01 in both data-sets. Generally speaking, there was a dramatic increase in p value which means the chances of no significance difference occurs.

P-value			
	MAP	nDCG	P-20
2017 Training Boolean Queries	0.023668075119543462	0.044674991490528386	0.5770324017264925
2017 Testing Boolean Queries	0.019687479985319584	0.008076808721681381	0.18414134229197232
2018 Training Boolean Queries	8.637450754862341E-4	0.001329680176953616	1.0
2018 Testing Boolean Queries	0.07111156499515059	0.010139554760400333	0.7450142227061849

Table 27: P-value for both data-sets with BM25,TF-IDF methods"Boolean-Query".

Table 27 shows the P-value over the data sets using Boolean queries. It can be clearly seen that, when we use nDCG to compare the values, both *2017Testing data and 2018 Training data* have a value less than 0.01 which indicates that there is a strong power. Using Boolean queries provides more insights of the significance difference among the methods than the title queries. by looking at ndcg measure, it can be inferred that there is a strong evidence that a difference exists, the p-value was slightly less than 0.05.

P-value			
	MAP	nDCG	P-10
2017 Training Query Reduction	0.18641143545847794	0.1625499990297401	NaN
2017 Testing Query Reduction	0.47621308371163706	0.5985448217956946	NaN
2018 Training Query Reduction	0.20781736434535603	0.18431374600615835	NaN
2018 Testing Query Reduction	0.19935961429586913	0.35944447768620646	NaN

Table 28: P-value for both data-sets after term-reduction process BM25,TF-IDF methods.

Regarding p-value after reduction process, there is a considerable increase in the values which means no difference exists.

2 Conclusion

`Effectiveness` is one of the most important requirements in Information Retrieval. We have seen that the best effective solution is BM25. Unsurprisingly, **BM25** outperforms **TF-IDF** due to the fact that it has experimentally shown as a robust method. One of the major advantages that BM25 has over other ranking algorithms is that it is an effective, flexible, and robust algorithm. **CombMNZ** and **CombSUM** have the same values and they both outperforms **Borda** in this project. By utilizing query reduction technique *IDF-r* on Boolean queries, we obtained worse results.

3

Query Reduction

3 Query Reduction methods

3.1 Review

So far we have seen how to score the documents and fuse these scores using some types of fusion algorithms and we have concluded that BM25 is the best ranking method compared to TF-IDF; CombMNZ and CombSUM were effective and comparable to each other in this experiment as opposed to Borda method. In this chaper we have introduced and implemented two types of query reduction methods which are so-called IDF-r and KLI. The work of [9] investigated the generation of queries, one of the most studied subjects in Natural Language Processing (NLP). The work compared different automated methods for common keyword extraction, such as the Kullback-Leibler divergence for informativeness (KLI), explored in the work of [11], the parsimonious language models (PLM), proposed in the work of [3], as well as the proportional inverse document frequency (IDF-r), proposed in the work of [4]. The work compared those automated methods with collected queries obtained by legal experts, through the Boolean method of comparison and best-match method. The article used almost 64,000 documents as its dataset, used both for training and test. It's pointed out by the authors that the most important challenge in the legal information retrieval (IR) field is that there is no standard definition for test collection, which takes a lot of effort in order to obtain statistically relevant datasets to train models for this application. The work of [2] presents a method for improving IR by distributional composition with term order probabilities, which could be an interesting way to maintain relevant data, without needing large collections of documents. The manual query generation methodology is commented on by the authors. More than one query was created for each topic and Boolean queries were performed manually by identification of important keywords from each of the topics, based on its citations. Match queries using Elasticsearch, which is a Lucene-based search server. It provides a full-text search engine and it's an important open-source library for NLP applications. Sentences and paragraphs are automatically explored using the different methods mentioned before. For all the models applied in the work, a list of terms included in an information object was used to create a subset list by ranking the terms according to each method, giving each one a score metric. Those scores identify relevant terms and then ranking them by relevance in order to extract the n most important terms. Regarding the manual queries, Boolean queries are reported as presenting significantly better results in the non-Boolean ones. Furthermore, the best-match method was

reported as outperforming all other manual queries. Regarding the automatically generated queries, for the topic paragraphs, the KLI model presents better results than any other methods, as well as the sentences and paragraphs baselines. The IDF-r method is the method presenting the lowest accuracy results than all other methods. All methods of reduction from topic sentences can only obtain higher accuracy rates at higher proportions, next to the totality of the sentence and half of the paragraphs for the topic paragraphs. When compared to the manual queries made by the legal experts, the automatic methods presented considerably lower accuracy rates and query expansion is recommended as a possible theme for exploration for better results. The work of [4] investigated automatic query generation methods to clinical queries, based on the interview data obtained from different patients narratives. The work intended to evaluate the efficiency in reducing the verbose characteristics of the patient report narrative into the ad-hoc query and how this automated query compares with the human-generated ones. It is pointed out that, in order to correctly obtain an automated query generated algorithm it is important to identify the main points of an effective query. This is also pointed out by the work of [1], applied to the Emerging Topic Tracking System (ETTS) for web security purposes, detecting the changes in the area of interest and preparing reports the changes periodically. The general applied proportional inverse document frequency (IDF-r) model is tested in comparison with a model based on medical related terms on patient narrative, identified through the UMLS medical thesaurus. This QuickUMLS model, proposed in the work of [10], is an information extraction platform to set up and present the results based on the mapping text to UMLS concepts. A third approach, using the QuickUMLS model to reduce the original patient report into only medical terms and then apply the IDF-r model. Effort is dedicated on the article to discuss the importance of reaching an optimal condition for the reduction proportion based on the query. This is achieved by determining a set of Query Performance Predictors, based on the work of [6]. Different predictors metrics were employed for each model. In addition, the QPP's used in the work as features in a model to predict the value of the query reduction proportion parameter, determined by given a particular topic (patient narrative), determining what query reduction proportion should be applied to it in order to maximize the effectiveness of the retrieval obtained. The training data was obtained by the selection of the best settings of the query reduction proportion parameter. In total, more than 1200 topic, query pairs were used to train the model. K-fold cross-validation was used to test the Generalized Linear Model used to predict the query reduction proportion model for overfitting. The model was compared with empirical evaluation using clinical trial tests from more than 200,000 trial documents available public. The narratives were summarized and those represented a human benchmark against automated methods in order to produce a comparison and test base. The results showed that the shorter human-generated summaries presented better fitting than the narrative, which can, in the author's point of view, motivate the development and investment in research of query reduction methods. The human ad-hoc queries showed better results overall. The

IDF-r model showed good results, but dependent on the value of the query reduction proportion parameter. As a summary, human-generated queries presents varied effectiveness and IDF-r method presented better results when used to reduced data containing just medical terms. Both works presented the accuracy and robustness of the **IDF-r** and **KLI** methods in prediction text-based queries from different applications. For that reason, the above mentioned methods will be employed in our present work.

3.2 IDF-r

In the previous chapter, we discussed this method and we have applied it on queries data-sets, and in order to compare the data sets before and after applying this method, we evaluated the results using Evaluation measures and surprisingly utilizing this method has not added any useful insights. In this section, we also applied this method (*IDF-r*) ,with retantion rates where the original query terms would be reduced at all values of r (in this case, 0.3,0.5, and 0.85 respectively), on the title queries and unsurprisingly the methods (**BM25** and **TF-IDF**) obtained more effective results. To support such claim, by looking at Table 2 , BM25 has a map value of 0.1462 and nDCG value of 0.5415; whereas in Table 31 "taking into account the best obtained value 0.85", the results of the method are more effective, 0.1474 and 0.5429 respectively. We can make the same considerations when comparing 2017 Training data-set results after and before performing IDF-r function (see Table 4 and Table 29). However, the results of both testing data-sets are less than the previous(before doing the reduction process), see table 1 and 3 , table 30 and 32 respectively. **Query reduction** *is useful in case of long queries that have multiple terms so that it makes queries more focused; less terms for which to iterate through the postings and therefore, faster query processing* . The basic idea behind this method, is that queries will be reduced at a reduction of 1-r, where r represents three retantion rates as mentioned above. We utilized ceiling function (see the following formula) to round the number of query terms to retain to an integer number. $Math.ceil((retantionRate) * titleNumberofWords)$,where titleNumberofWords refers to the total number of terms in the title; retantionRate refers to the three rates (we created a loop over them); we basically multiplied each of these rates by the number of terms in the title. According to Table 29, results show that BM25 performs better than

	MAP			*Pprec*			*nDCG*		
	0.3	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**
BM25	0.0850	0.0882	0.1174	0.1168	0.1188	0.1565	0.4475	0.4641	0.4864
TF_IDF	0.0791	0.0848	0.1133	0.1083	0.1120	0.1517	0.4469	0.4574	0.4804

Table 29: 2017 Training Title Queries using IDF-r method.

TF-IDF. We chose the value of 0.85 as the best obtained value,since utilizing this value yielded to better results, and we applied it on testing data-set (see Table 30). Utilizing the retantion rate at 0.85, achieves higher results. Generally speaking, both methods (BM25,TF-IDF) obtain better results on testing data

compared to training data-sets. Similarly, same considerations can be applied to 2018 data-sets.

	MAP			*Pprec*			*nDCG*		
	0.3	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**
BM25	0.1155	0.1216	0.1612	0.1312	0.1337	0.1733	0.4903	0.5032	0.5386
TF_IDF	0.1005	0.1040	0.1552	0.1233	0.1145	0.1688	0.4712	0.4813	0.5307

Table 30: 2017 Testing Title Queries using IDF-r method.

	MAP			*Pprec*			*nDCG*		
	0.3	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**
BM25	0.1002	0.1080	0.1474	0.1178	0.1202	0.1685	0.4935	0.5095	0.5429
TF_IDF	0.0935	0.0986	0.1406	0.1103	0.1092	0.1600	0.4848	0.4960	0.5350

Table 31: 2018 Training Title Queries using IDF-r method.

	MAP			*Pprec*			*nDCG*		
	0.3	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**
BM25	0.1068	0.1159	0.1584	0.1169	0.1368	0.1876	0.5384	0.5488	0.5879
TF_IDF	0.1046	0.1135	0.1616	0.1082	0.1285	0.1799	0.5329	0.5411	0.5866

Table 32: 2018 Testing Title Queries using IDF-r method.

As mentioned above, the best value was 0.85. Therefore, we applied this on testing data-sets and we obtained better results.

By looking at Table 2 and 31 "considering r= 0.85", we can infer that after performing the reduction function we have obtained better insights.

In above tables, we have come to the conclusion that BM25 still outperforms TF-IDF which is expected. The evaluation measures for 0.85 value show that this value is the best among other values. One thing should be noted that after applying IDF-r function, TF-IDF outperforms BM25 in some measures such as map measure(see Table 32, r=0.85).

– **Fusion Algorithms**

	MAP			***Rprec***		
	0.3	0.5	0.85	0.3	0.5	0.85
Borda	0.0817	0.0865	0.1164	0.1091	0.1142	0.1565
CombSUM	0.0809	0.0870	0.1166	0.1117	0.1185	0.1568
CombMNZ	0.0809	0.0870	0.1166	0.1117	0.1185	0.1568

Table 33: Fusion Results on 2017 Training data-set.

It can be clearly seen from the tables that both CombSUM and CombMNZ still have the same values; after performing the fusion on all values, we can observe that the results of each fusion method are better than the previous. In Table 39, for instance, we obtained higher result (i.e. map of CombSUM

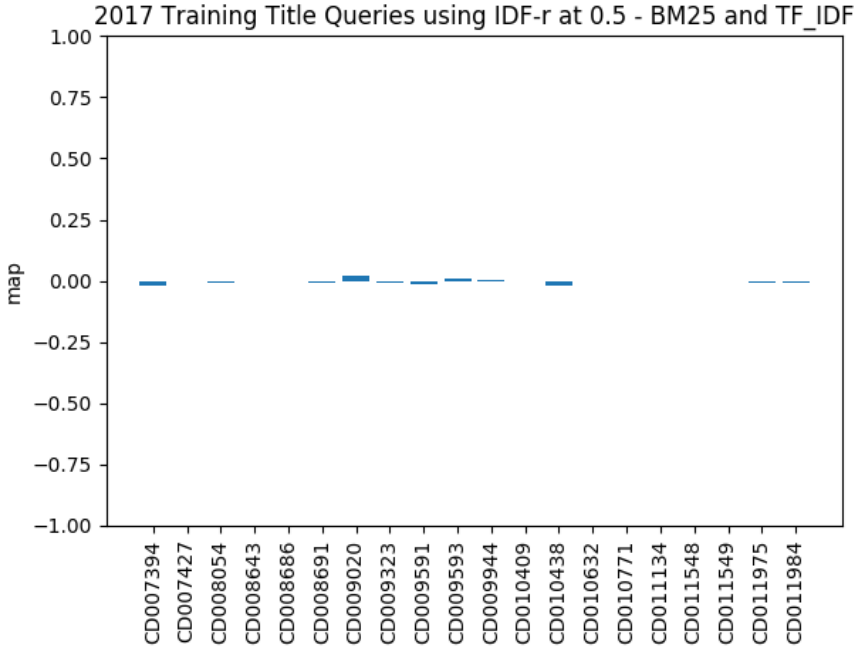

Fig. 43: Gain/Loss1

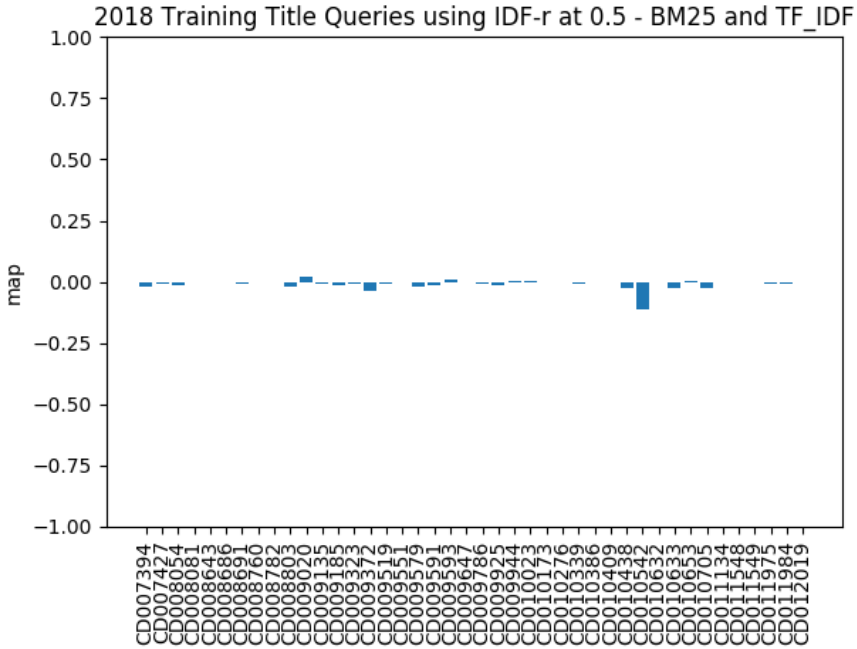

Fig. 44: Gain/Loss

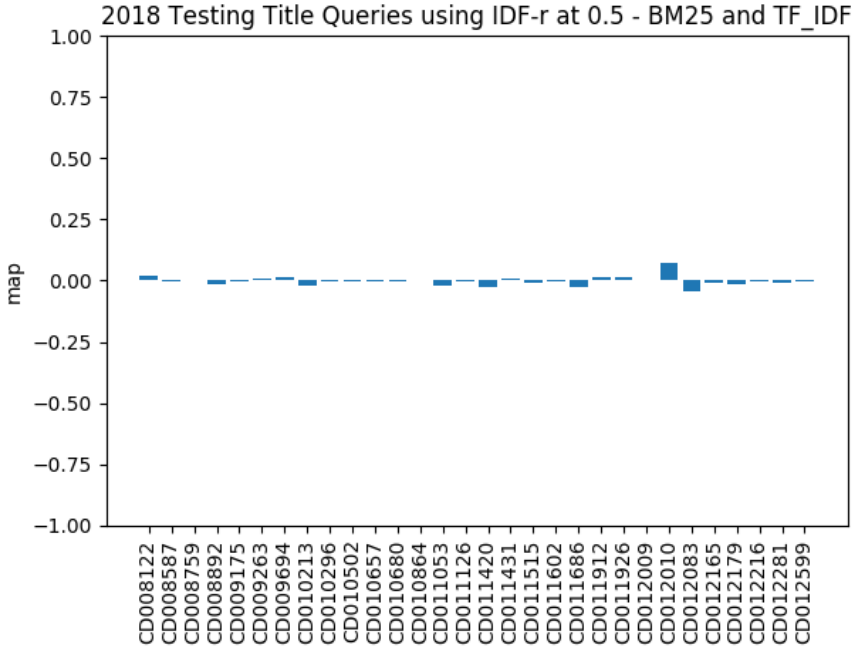

Fig. 45: Gain/Loss

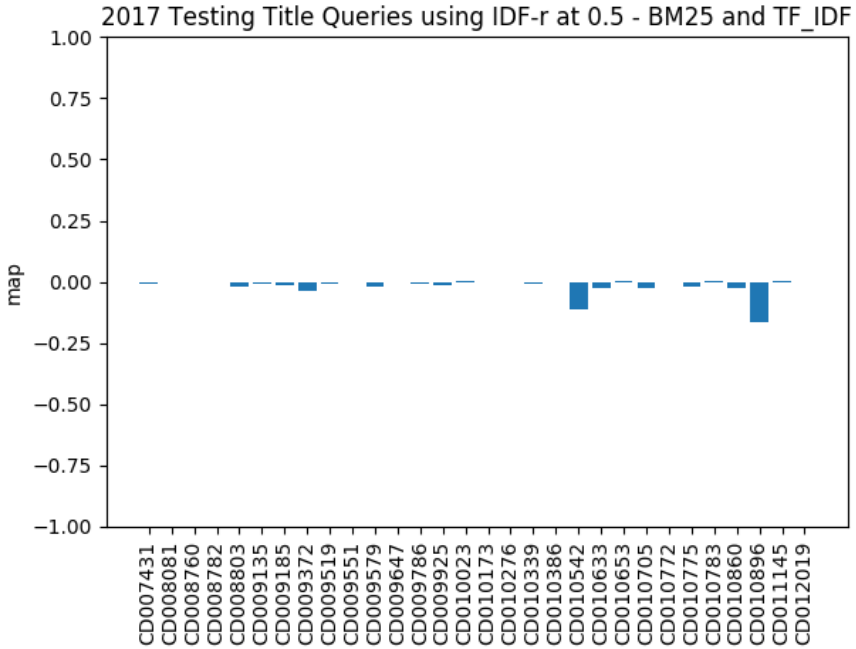

Fig. 46: Gain/Loss

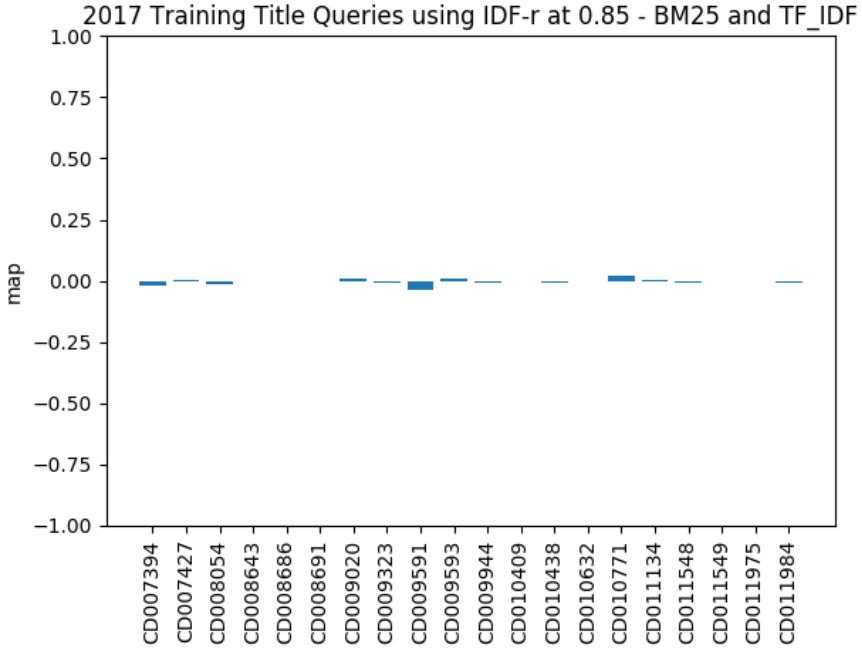

Fig. 47: Gain/Loss

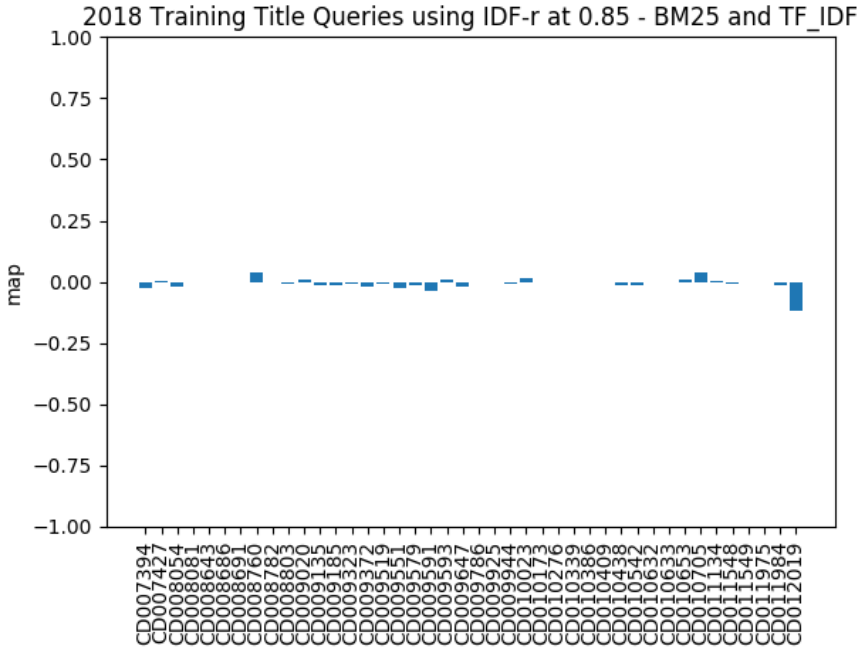

Fig. 48: Gain/Loss

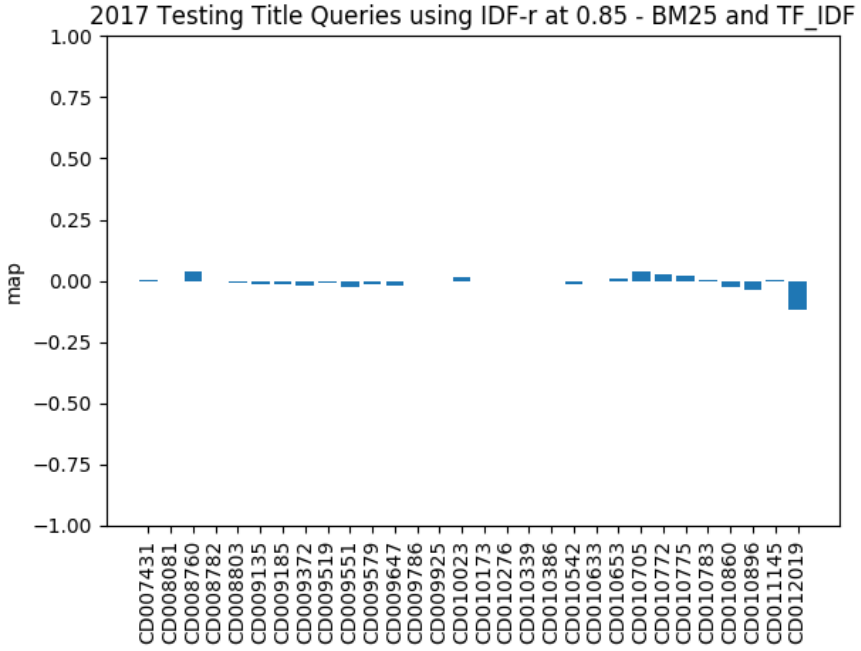

Fig. 49: Gain/Loss

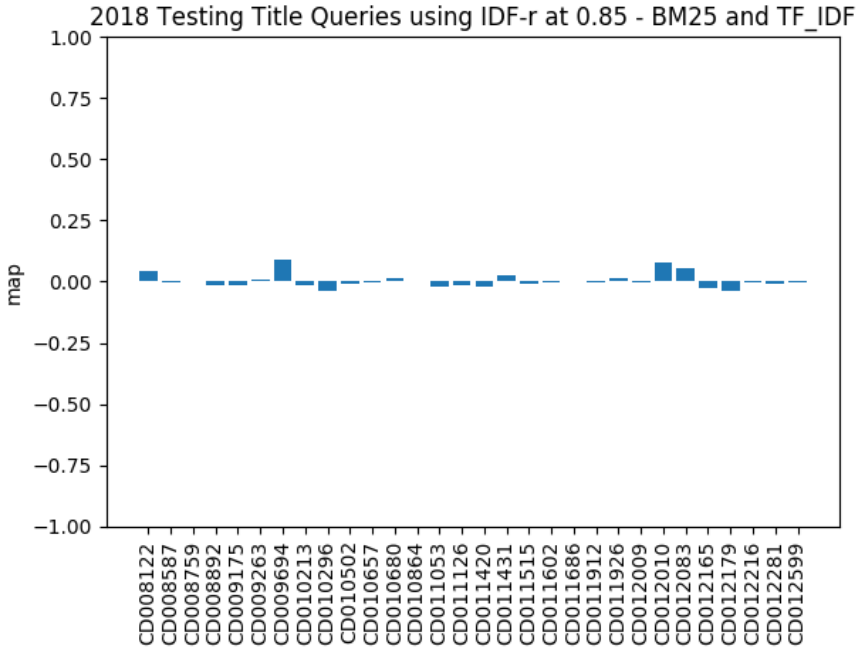

Fig. 50: Gain/Loss

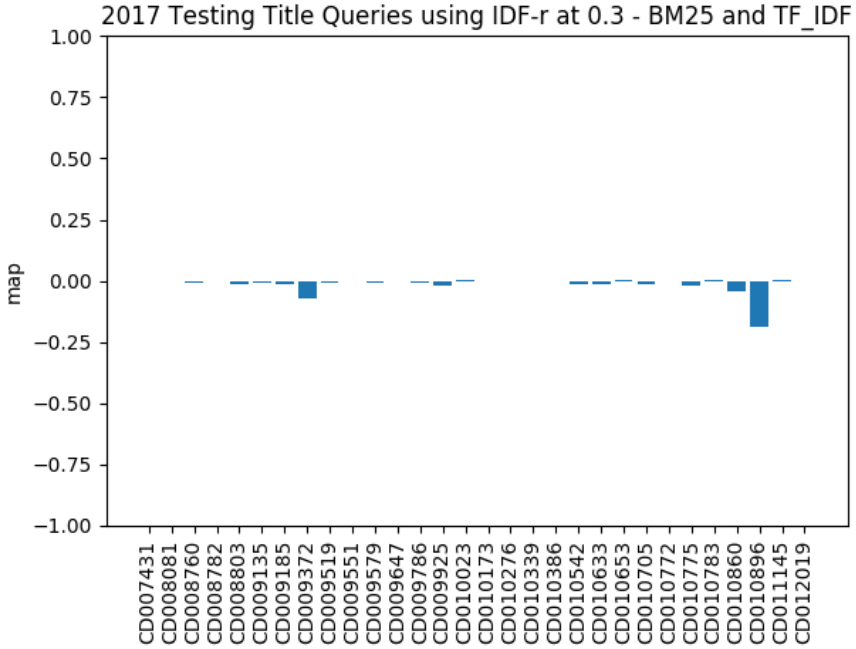

Fig. 51: Gain/Loss

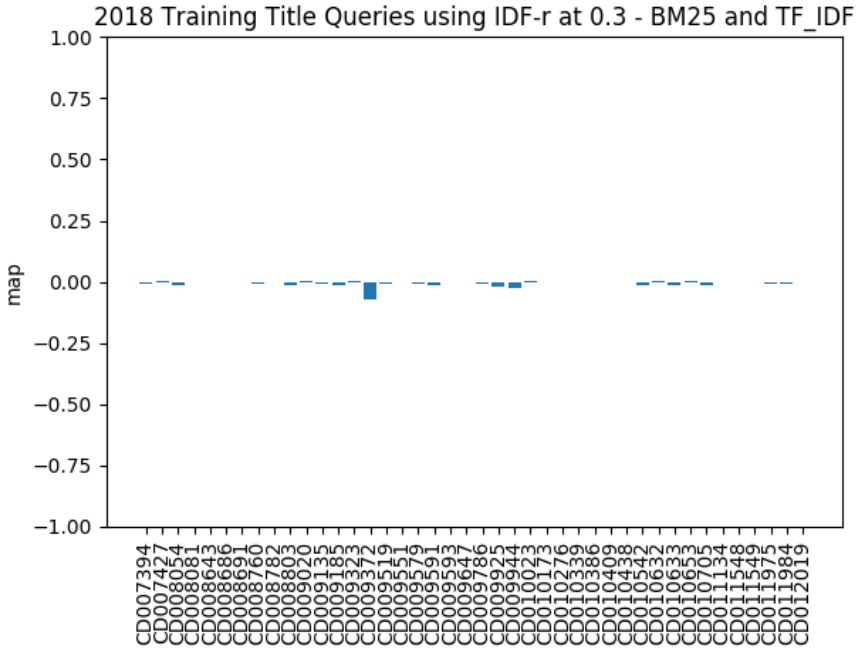

Fig. 52: Gain/Loss

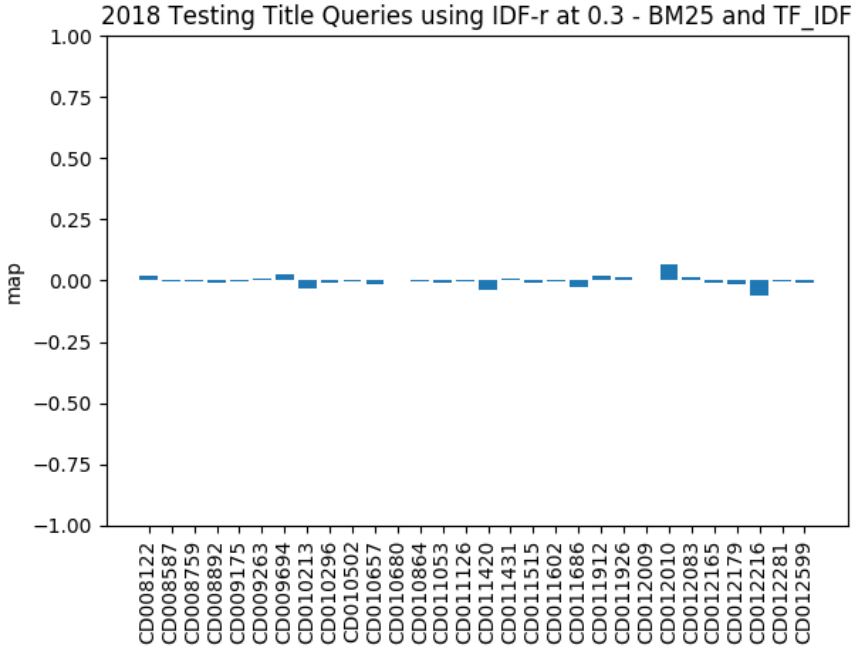

Fig. 53: Gain/Loss

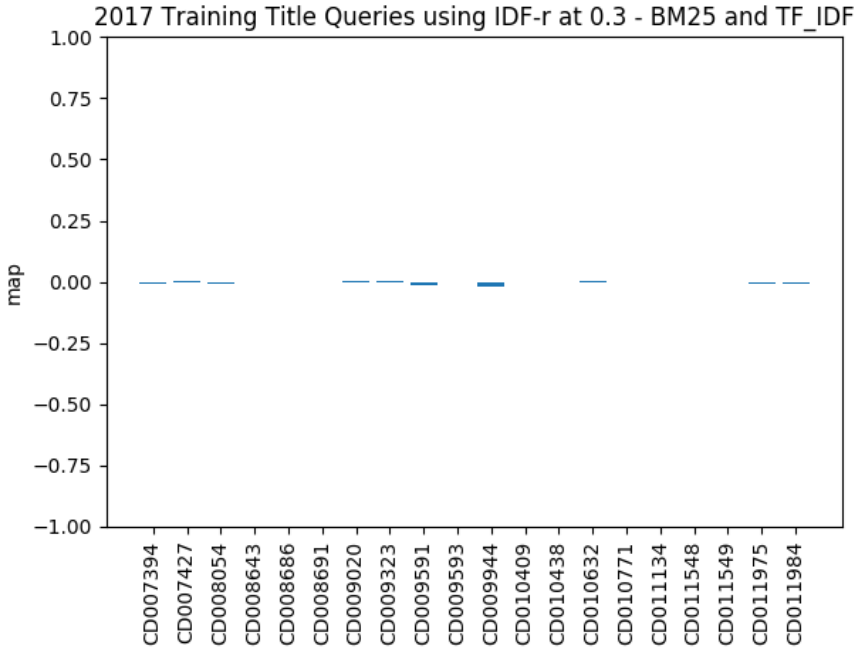

Fig. 54: Gain/Loss

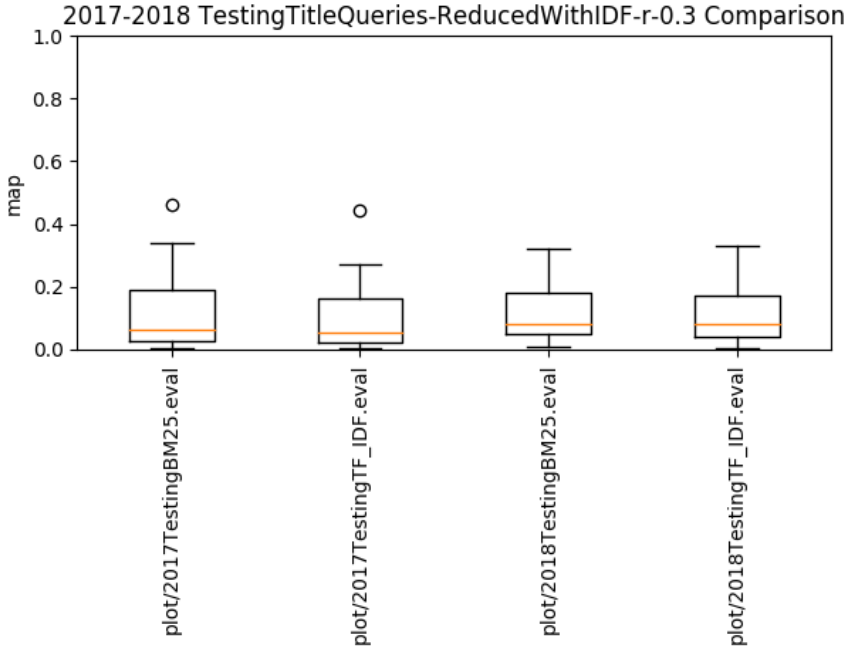

Fig. 55: Box-plot

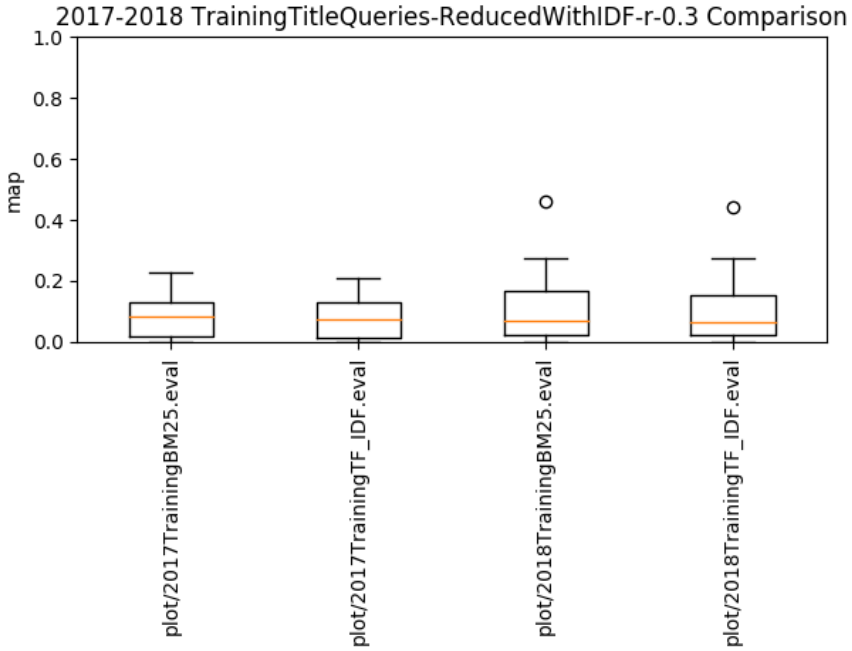

Fig. 56: Box-plot

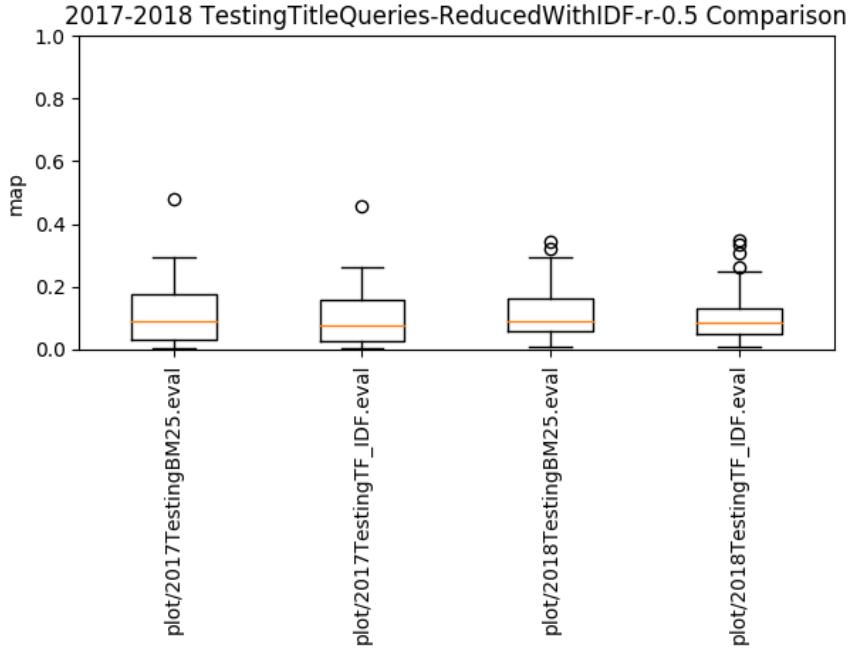

Fig. 57: Box-plot

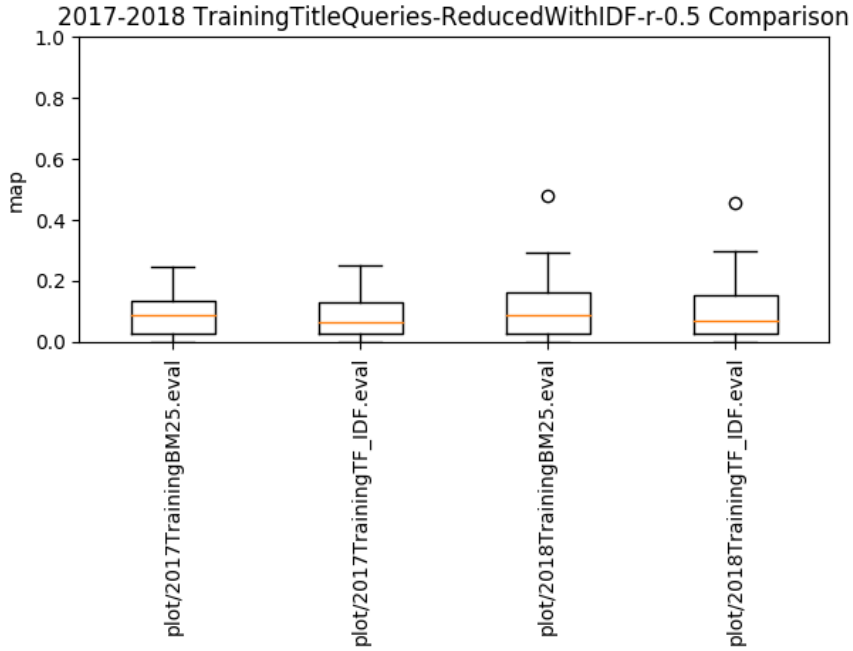

Fig. 58: Box-plot

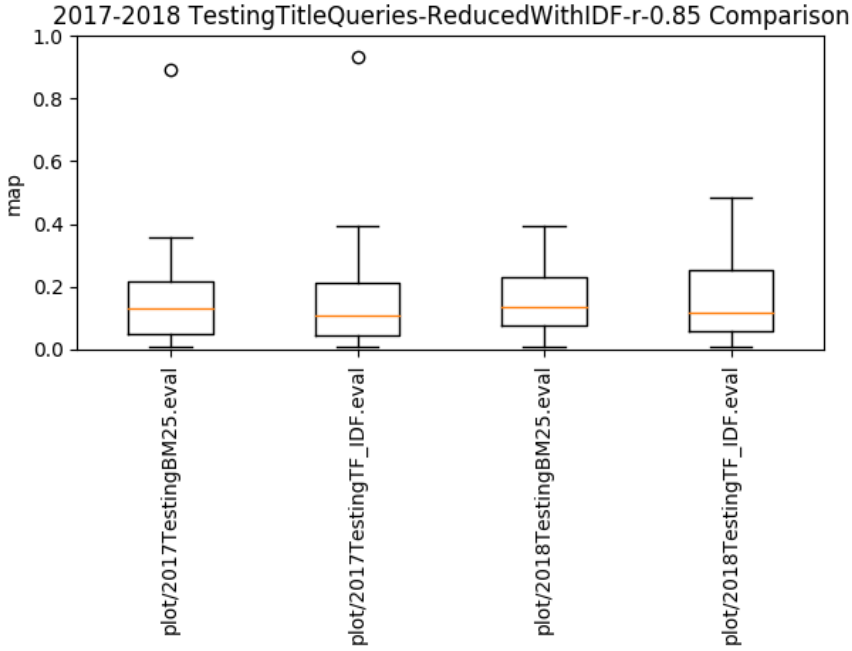

Fig. 59: Box-plot

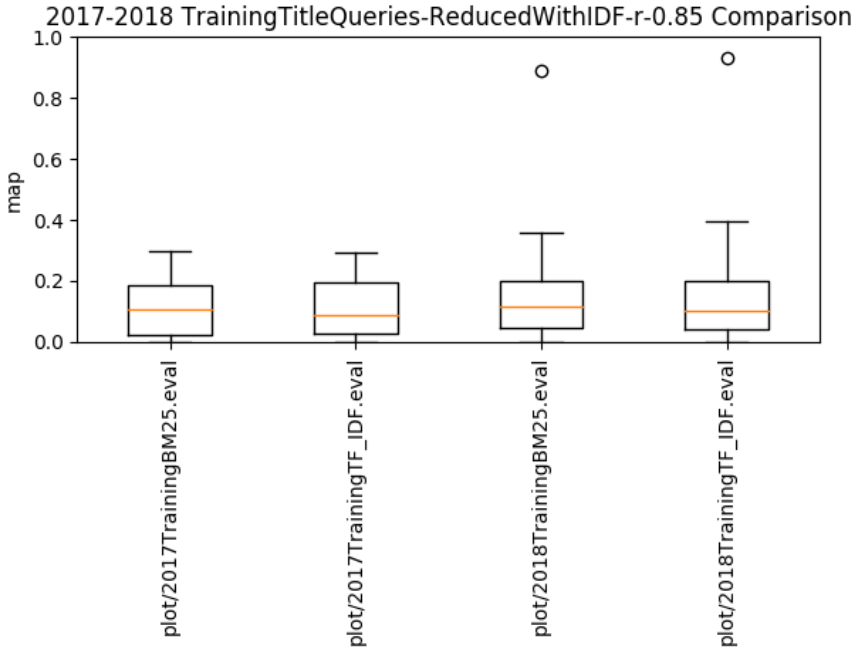

Fig. 60: Box-plot

	MAP			***Rprec***		
	0.3	0.5	0.85	0.3	0.5	0.85
Borda	0.1096	0.0865	0.1592	0.1260	0.1142	0.1799
CombSUM	0.1111	0.1203	0.1642	0.1310	0.1355	0.1840
CombMNZ	0.1111	0.1203	0.1642	0.1310	0.1355	0.1840

Table 34: Fusion Results on 2017 Testing data-set.

	MAP			***Rprec***		
	0.3	0.5	0.85	0.3	0.5	0.85
Borda	0.1053	0.1139	0.1593	0.1064	0.1276	0.1833
CombSUM	0.1078	0.1154	0.1596	0.1112	0.1338	0.1831
CombMNZ	0.1078	0.1154	0.1596	0.1112	0.1338	0.1831

Table 35: Fusion Results on 2018 Testing data-set.

	MAP			***Rprec***		
	0.3	0.5	0.85	0.3	0.5	0.85
Borda	0.0972	0.1045	0.1451	0.1146	0.1170	0.1699
CombSUM	0.0988	0.1063	0.1477	0.1173	0.1207	0.1740
CombMNZ	0.0988	0.1063	0.1477	0.1173	0.1207	0.1740

Table 36: Fusion Results on 2018 Training data-set.

	MAP	***Rprec***
	All values	*All values*
Borda	0.1002	0.1268
CombSUM	0.1001	0.1267
CombMNZ	0.1001	0.1267

Table 37: Fusion Results for all IDF-r values(0.3,0.5,0.85) 2017 Training data-set.

	MAP	***Rprec***
	All values	*All values*
Borda	0.1203	0.1311
CombSUM	0.1202	0.1340
CombMNZ	0.1202	0.1340

Table 38: Fusion Results for all IDF-r values(0.3,0.5,0.85) 2018 Training data-set.

	MAP	***Rprec***
	All values	*All values*
Borda	0.1333	0.1521
CombSUM	0.1402	0.1592
CombMNZ	0.1402	0.1592

Table 39: Fusion Results for all IDF-r values(0.3,0.5,0.85) 2017 Testing data-set.

= 0.1402 while in table 34, the value of the map measure for CombSUM equals 0.1111). From above discussion we can infer that, fusing all values together can be very effective. Comparing fusion methods in Table 7 with

	MAP	***Rprec***
	All values	*All values*
Borda	0.1235	0.1449
CombSUM	0.1183	0.1363
CombMNZ	0.1183	0.1363

Table 40: Fusion Results for all IDF-r values(0.3,0.5,0.85) 2018 Testing data-set.

	MAP	***Rprec***
	All values	*All values*
Borda	0.1607	0.1767
CombSUM	0.1378	0.1686
CombMNZ	0.1378	0.1686

Table 41: Fusion Results for all IDF-r values 2018 Testing data-set with other runs.

	MAP	***Rprec***
	All values	*All values*
Borda	0.1333	0.1521
CombSUM	0.1489	0.1627
CombMNZ	0.1489	0.1627

Table 42: Fusion Results for all IDF-r values 2017 Testing data-set with other runs.

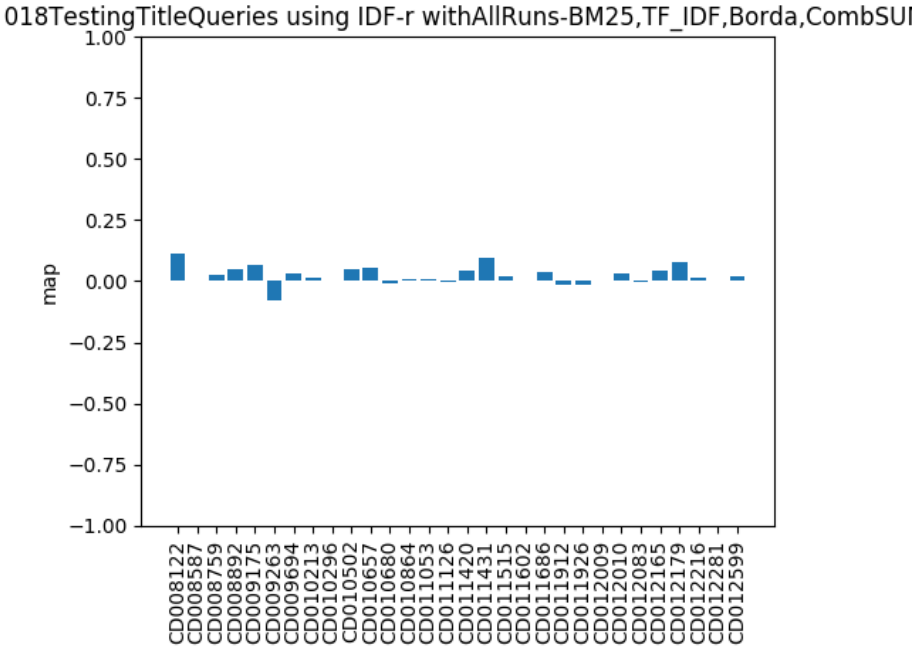

Fig. 61: Gain/Loss

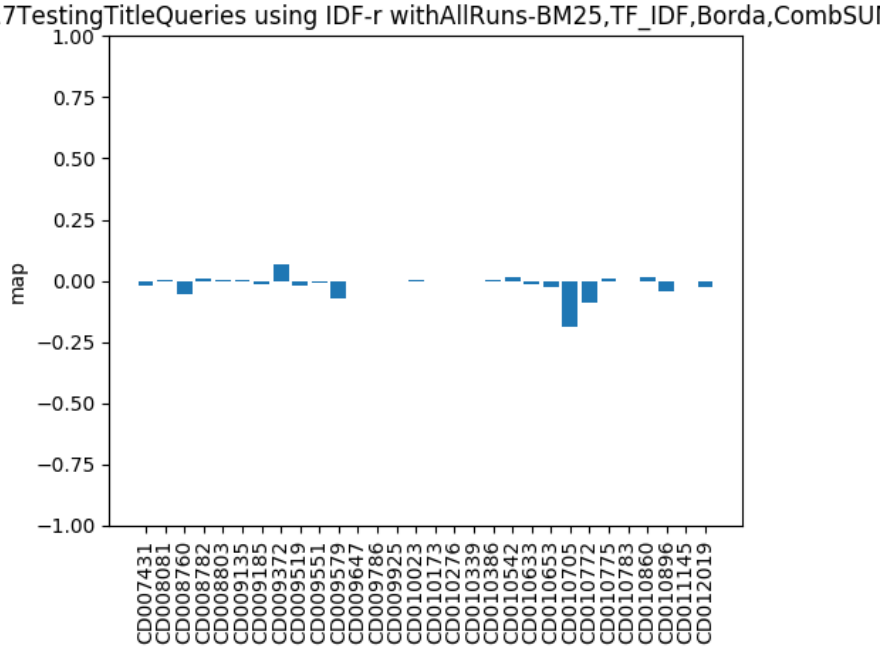

Fig. 62: Gain/Loss

that of table 35, for example, by taking the best value we obtained 0.85, it can be inferred that IDF-r presents less results.
When fuse the dataset with the provided runs, we obtained better results. In Table 42, for instance, we obtained higher result (i.e. map of CombSUM = 0.1489 while in table 39, map of the same method is equal to 0.1402).
According to the most effective value (namely, 0.3,0.5,0.85), we found that 0.85 is the best value where it provides higher results compared to the other values. Further insights could be seen in the tables and figures.

we sat BM25 parameter to 0.25 which yields the best value and then we applied both reduction methods on that data sets.

3.3 KLI

KLI stands for Kullback-Leibler divergence for informativeness. KLI is another proposed reduction method which based on the probability concept.The KLI of a term is formally defined in the Fig. 63, where D is the set of documents provided to rank, and C is the entire collection of documents. In this context, we have applied this method by approximating $P(t|D)$ as (number of documents d in D that contain t)/(number of documents in D).

$$KLI(t) = P(t|D) * log\frac{P(t|D)}{P(t|C)}$$

Fig. 63: KLI

	MAP			*Rprec*			*nDCG*		
	0.3	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**
BM25	0.0909	0.1037	0.1204	0.1150	0.1400	0.1605	0.4509	0.4682	0.4922
TF_IDF	0.0870	0.0974	0.1145	0.1096	0.1276	0.1513	0.4467	0.4621	0.4820

Table 43: 2017 Training Title Queries using KLI method.

	MAP			*Rprec*			*nDCG*		
	0.3	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**
BM25	0.1005	0.1346	0.1689	0.1032	0.1454	0.1941	0.4741	0.4987	0.5458
TF_IDF	0.0927	0.1238	0.1602	0.0895	0.1323	0.1729	0.4632	0.4885	0.5361

Table 44: 2017 Testing Title Queries using KLI method.

	MAP			*Rprec*			*nDCG*		
	0.3	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**
BM25	0.1035	0.1349	0.1539	0.1096	0.1524	0.1815	0.4919	0.5182	0.5513
TF_IDF	0.0964	0.1244	0.1448	0.1019	0.1410	0.1620	0.4831	0.5084	0.5395

Table 45: 2018 Training Title Queries using KLI method.

	MAP			*Rprec*			*nDCG*		
	0.3	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**
BM25	0.1241	0.1335	0.1678	0.1423	0.1646	0.1993	0.5427	0.5578	0.5943
TF_IDF	0.0784	0.1339	0.1691	0.0781	0.1529	0.1950	0.4828	0.5563	0.5917

Table 46: 2018 Testing Title Queries using KLI method.

As shown in the above tables, results show that BM25 outperforms TF-IDF. We chose the value of 0.85 as the best obtained value,since utilizing this value yielded to better results, and we applied it on testing data-sets (see Table 44). Utilizing the retantion rate at 0.85, achieves better results. Generally speaking, both methods (BM25,TF-IDF) obtained better results on testing data compared to training data-set. Similarly, same considerations can be applied to 2018 data-sets. From this, we can infer that both IDF-r and KLI performs very well on testing data compared to training one.

Comparing KLI to the other reduction method (IDF-r), we can infer that KLI performs better than IDF-r. To support such claim, by looking at Table 31(IDF-r on 2018 training data) and Table 45(KLI on the same data), BM25 performs much better in KLI than that of IDF-r. In addition, by looking at Table 4 and Table 43, KLI provides better results taking into account the best value 0.85.

As shown in the below figures, the gain/loss shows better results on KLI compared to that of the previous methods. KLI performs better on 2018 dataset compared to 2017 ones.

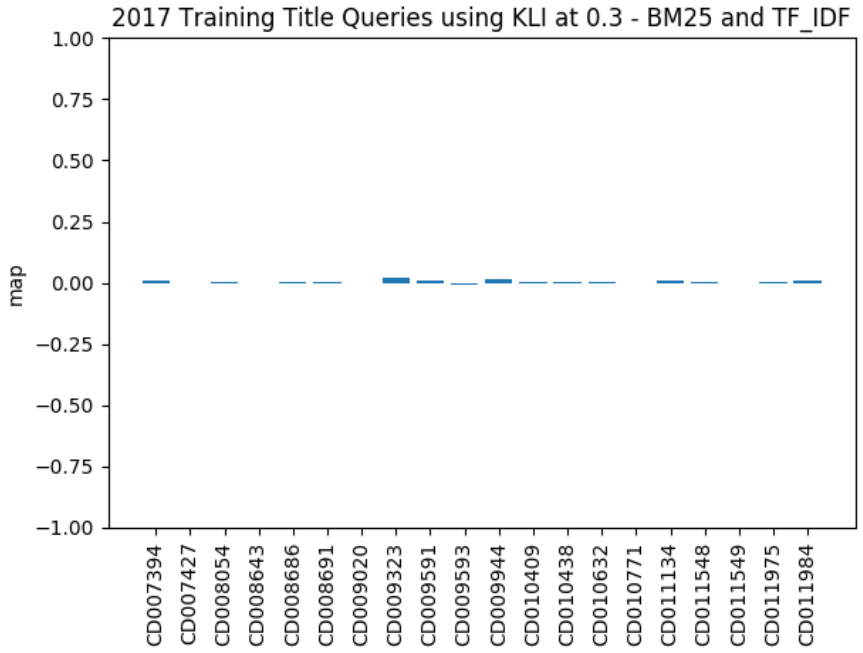

Fig. 64: Gain/Loss

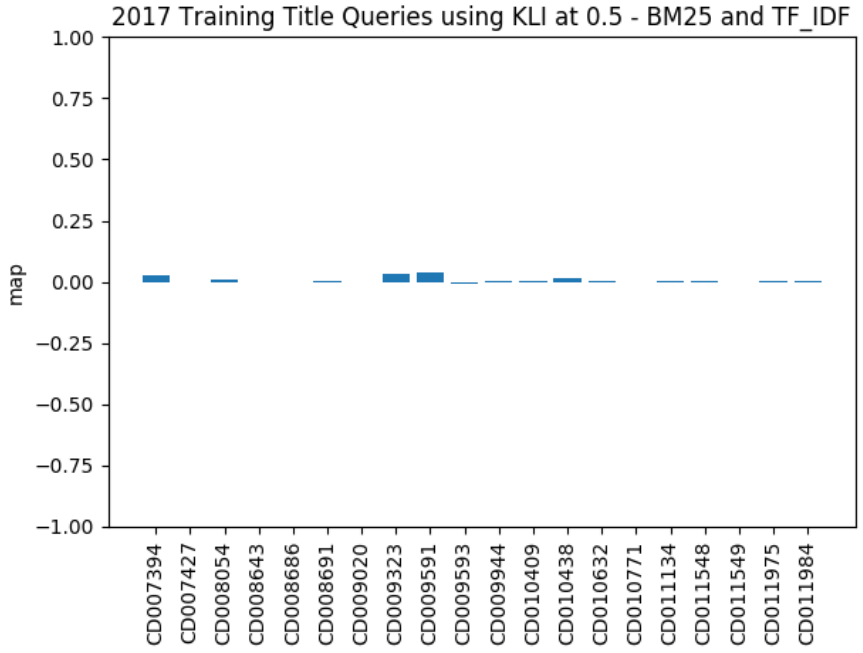

Fig. 65: Gain/Loss

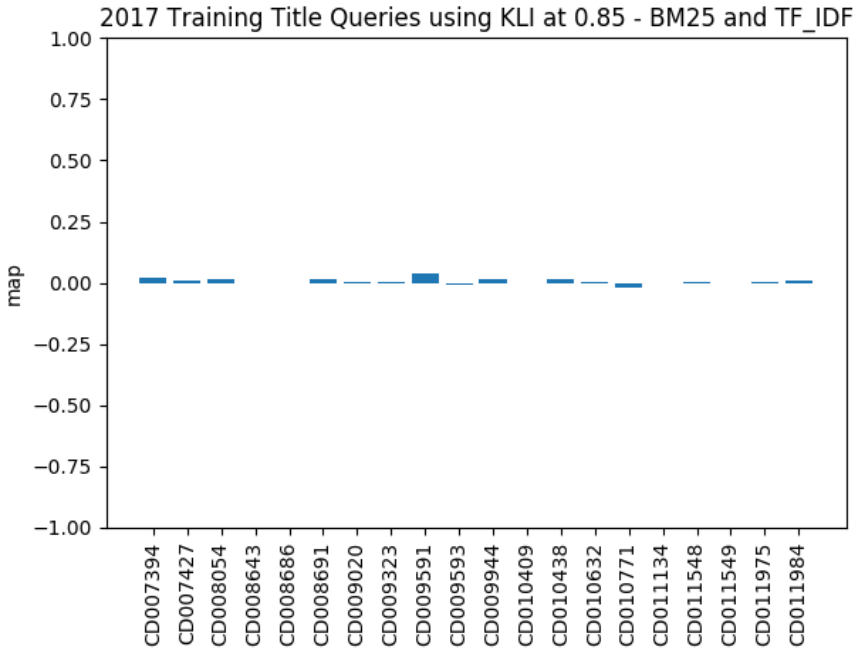

Fig. 66: Gain/Loss

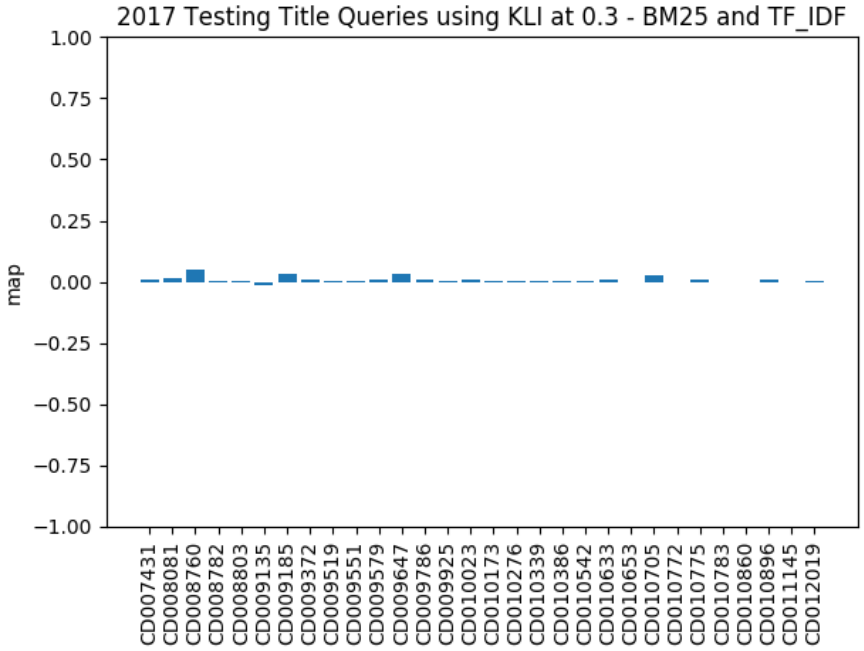

Fig. 67: Gain/Loss

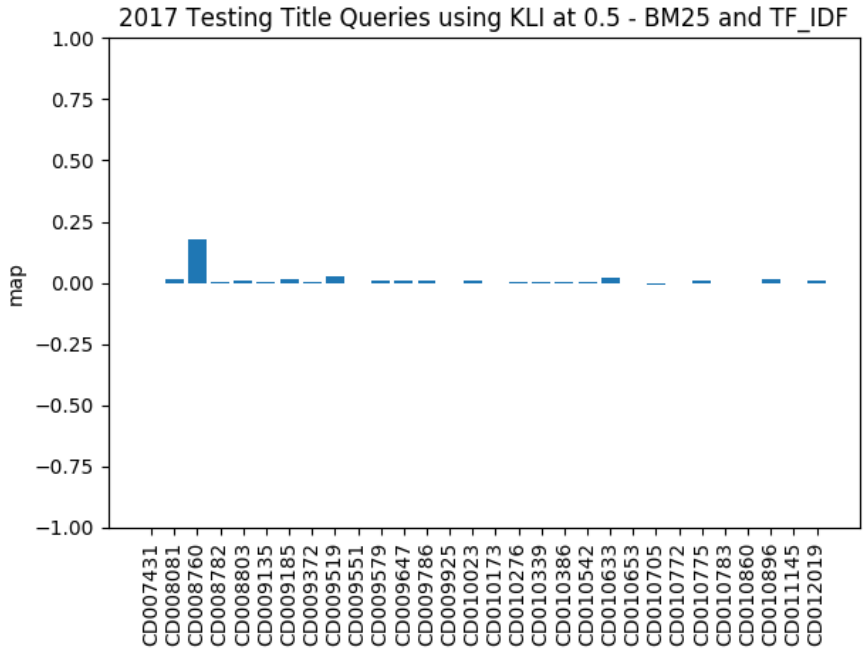

Fig. 68: Gain/Loss

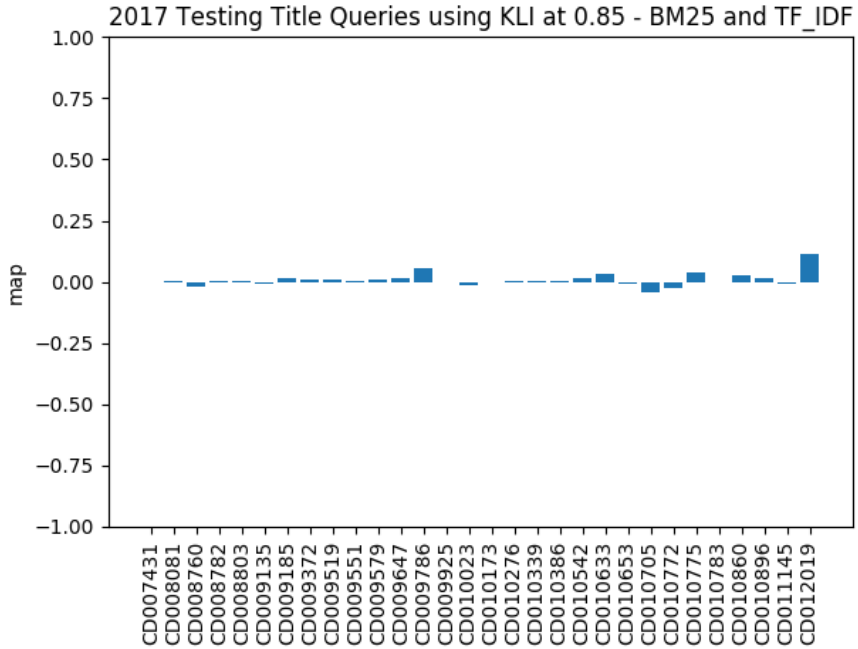

Fig. 69: Gain/Loss

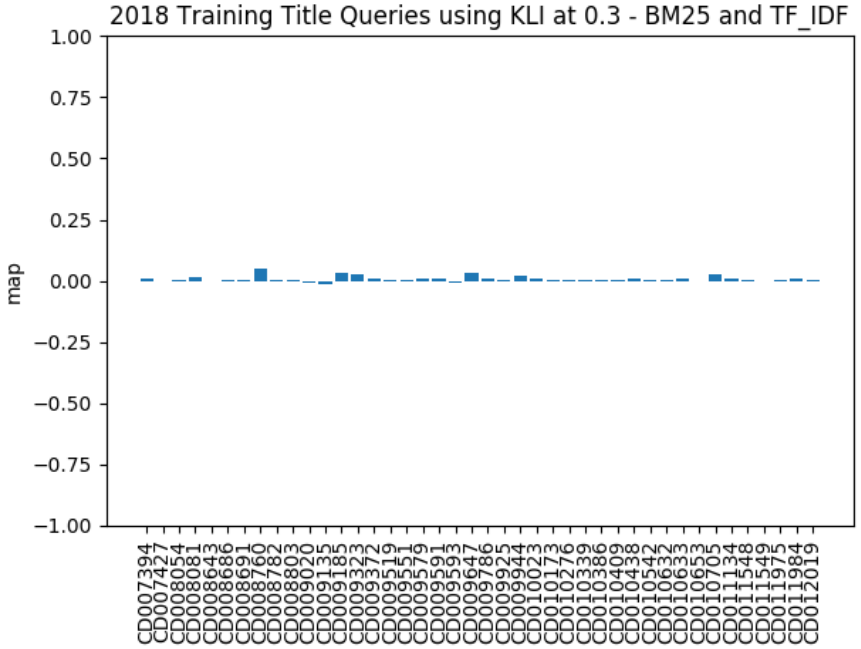

Fig. 70: Gain/Loss

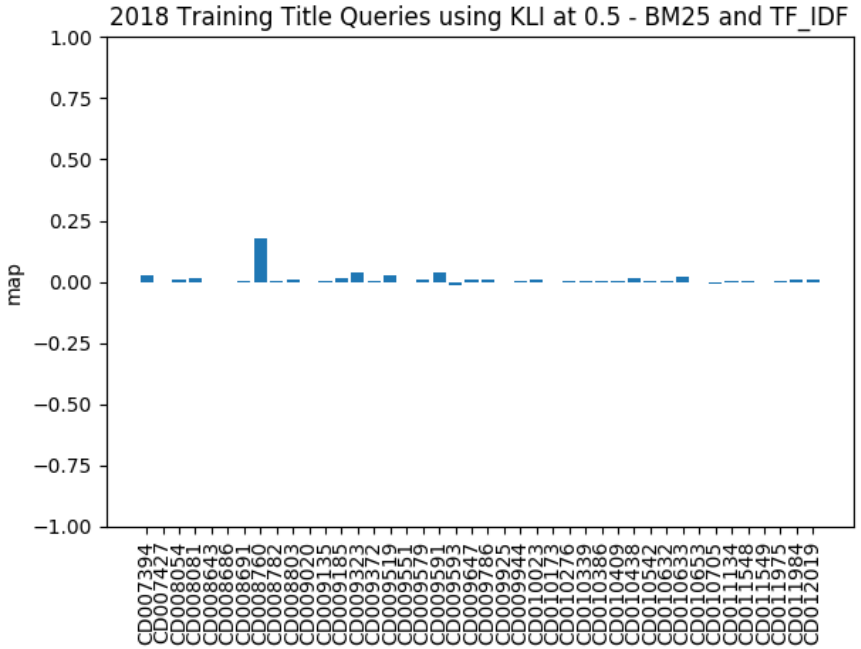

Fig. 71: Gain/Loss

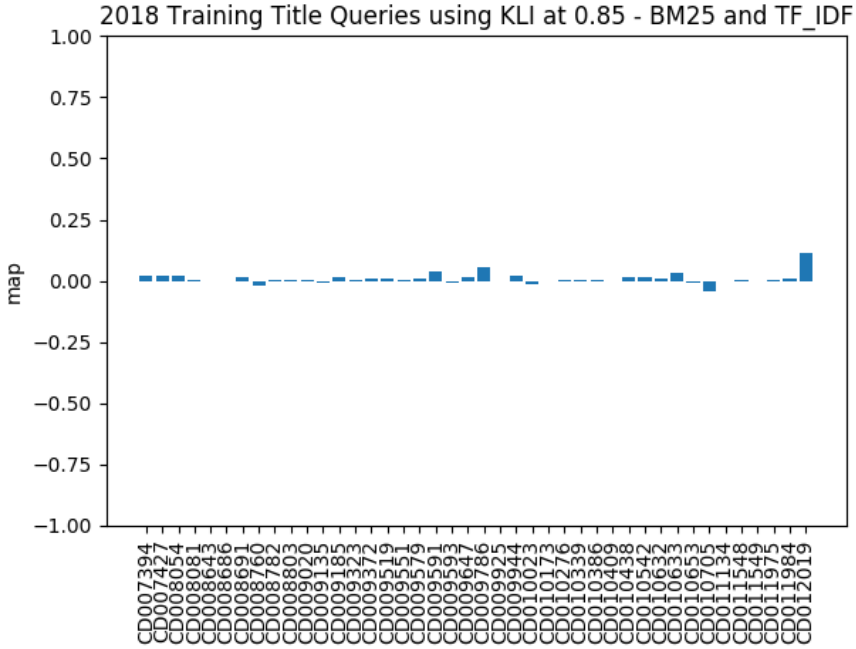

Fig. 72: Gain/Loss

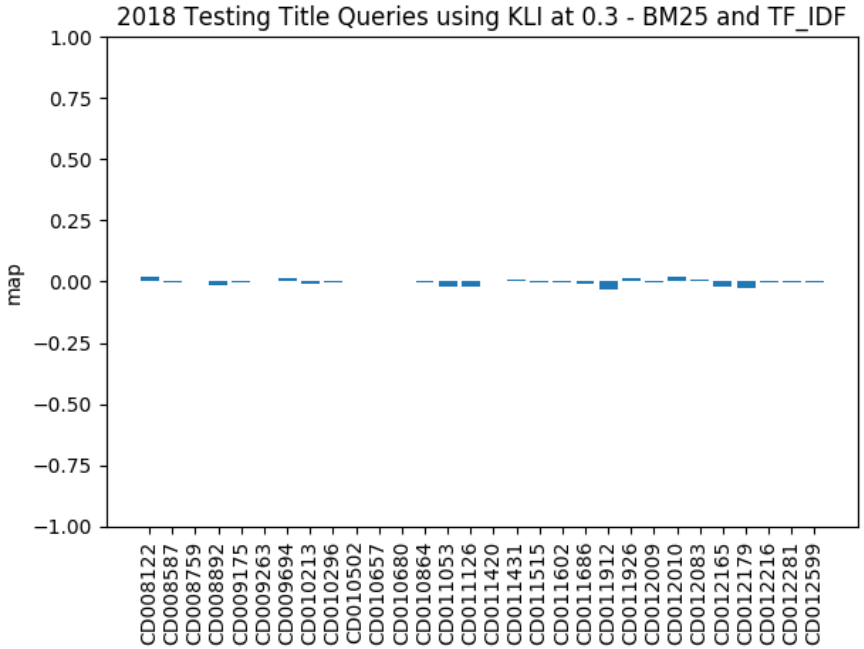

Fig. 73: Gain/Loss

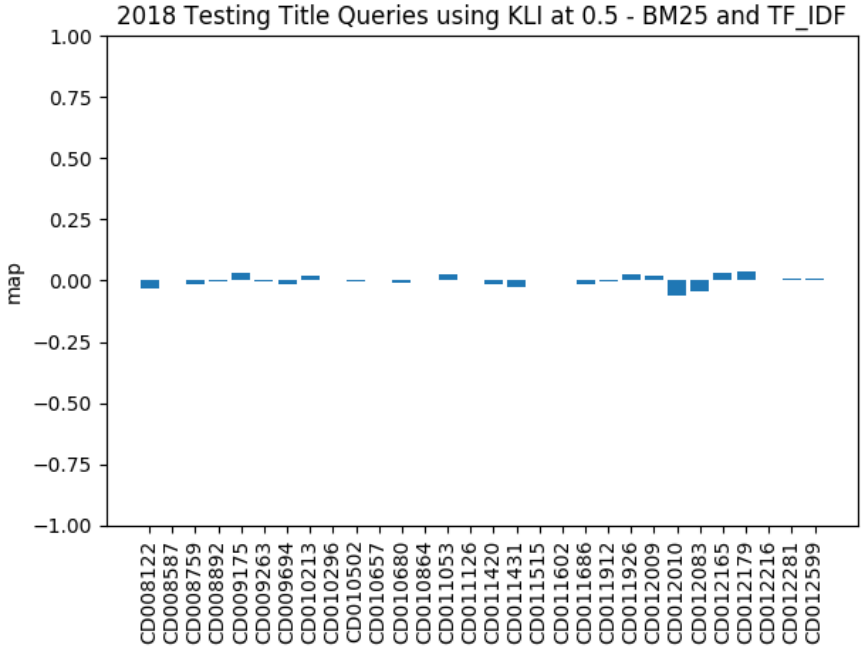

Fig. 74: Gain/Loss

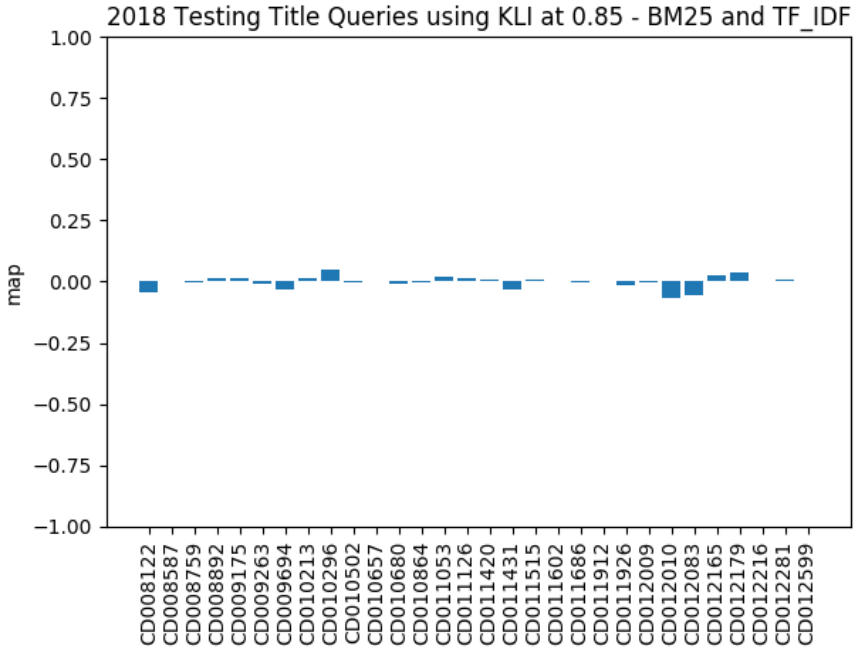

Fig. 75: Gain/Loss

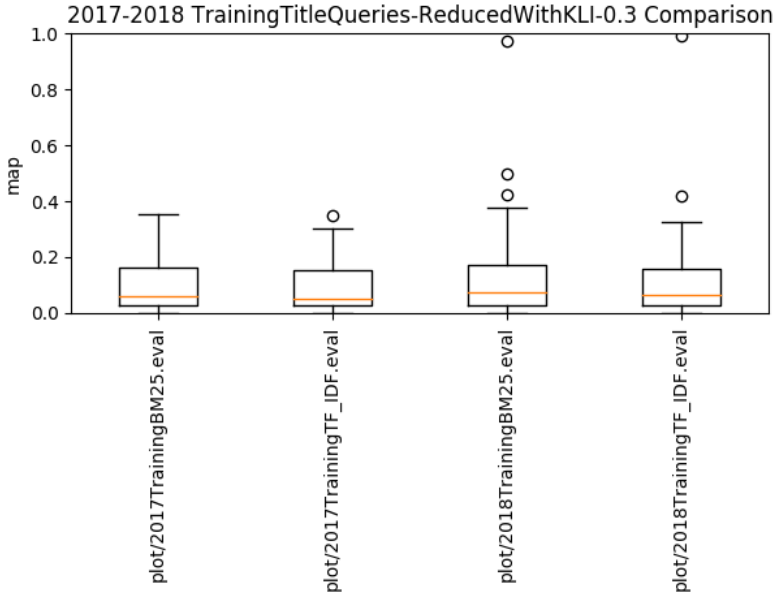

Fig. 76: Box-plot

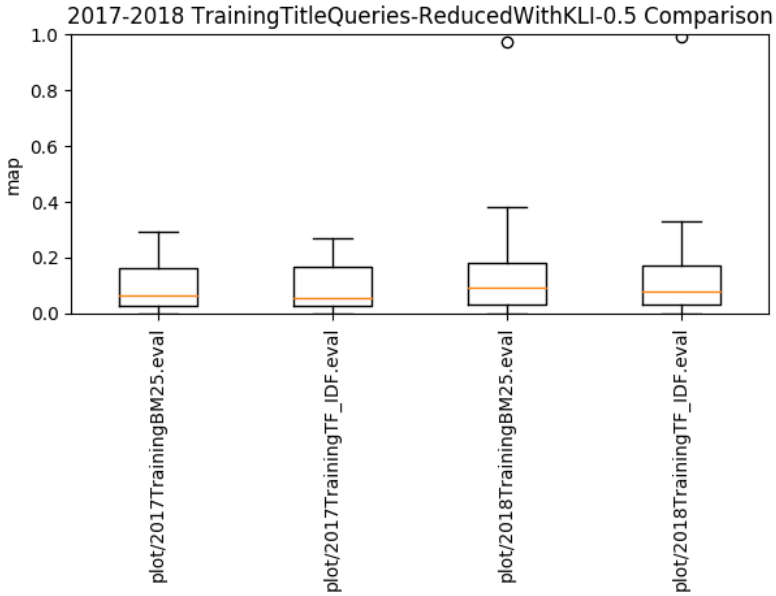

Fig. 77: Box-plot

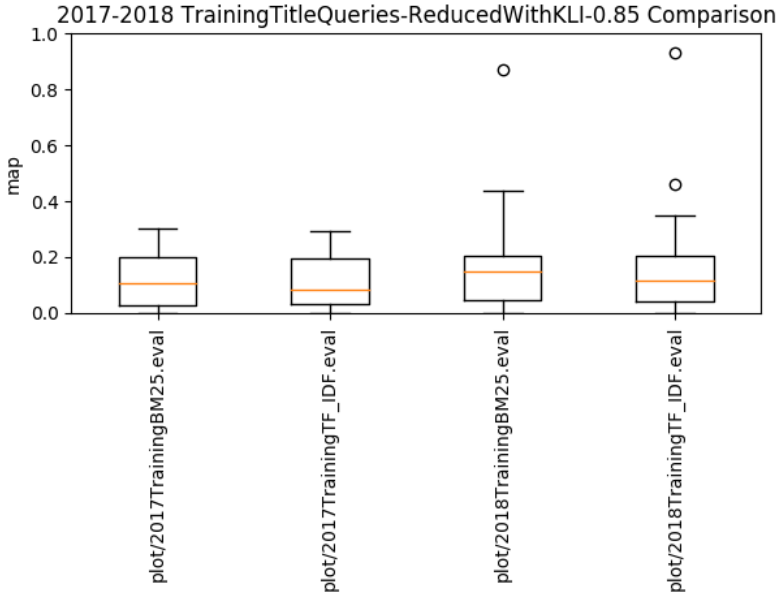

Fig. 78: Box-plot

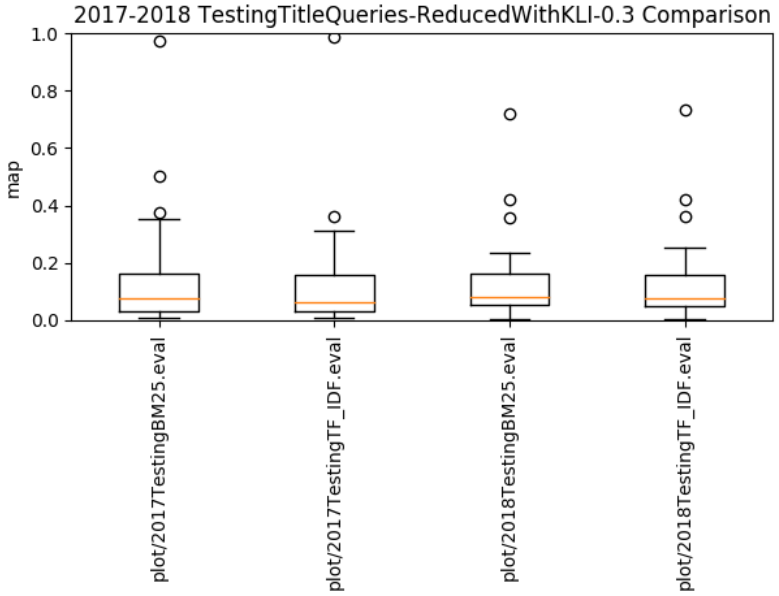

Fig. 79: Box-plot

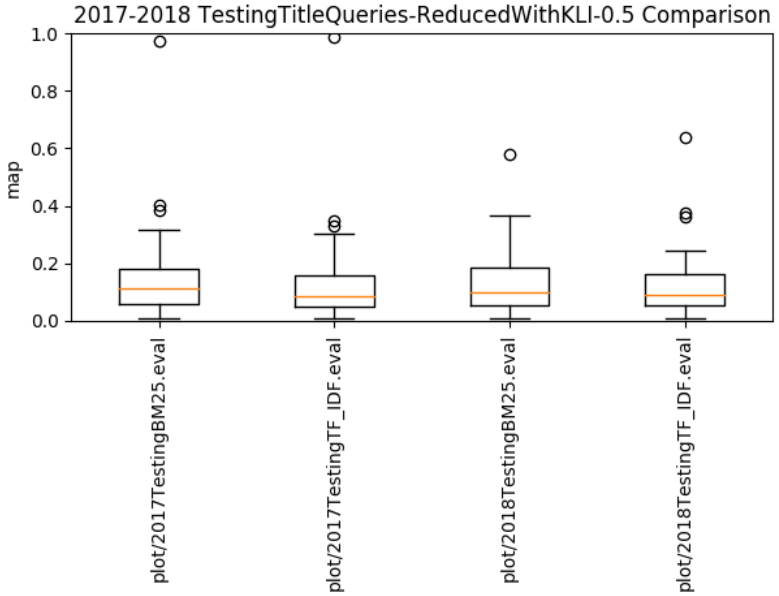

Fig. 80: Box-plot

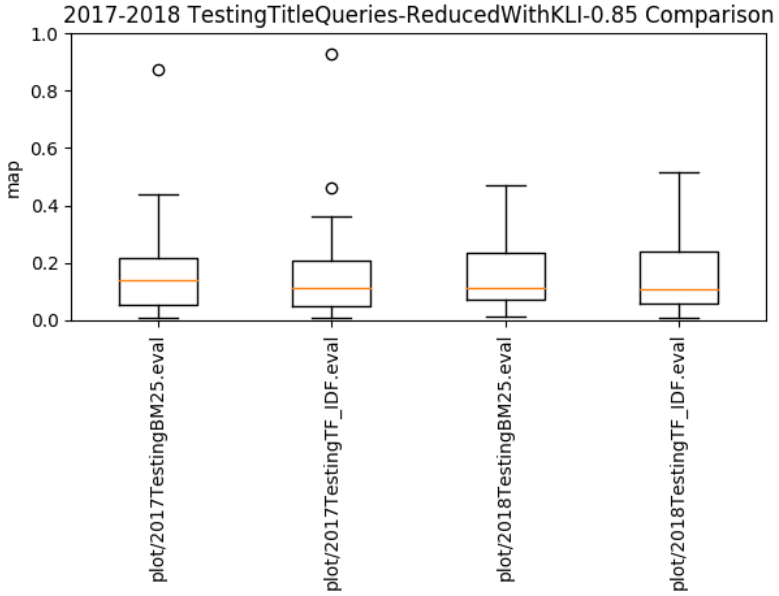

Fig. 81: Box-plot

3.4 Statistical Significance Tests

	MAP			*Pprec*			*nDCG*		
P-value	**0.3**	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**
2017 Training Title Reduction	0.0258	0.1167	0.1500	0.1235	0.0576	0.4758	0.4028	0.1270	0.1106
2017 Testing Title Reduction	0.0296	0.0102	0.2330	0.2721	0.1343	0.6795	0.0302	0.0058	0.2153
2018 Training Title Reduction	0.0010	0.0037	0.0544	0.0170	0.0041	0.2933	0.0047	0.0022	0.0854
2018 Testing Title Reduction	0.5913	0.5033	0.5570	0.1984	0.2852	0.2719	0.2898	0.0908	0.7968

Table 47: p-value for both data-sets using IDF-r method on Title Queries.

Table 47 and 48 show how these methods differ from each other, the observations showed that there is a difference between the methods, so by looking at "2018 Training data-set",for instance,we can infer that there is a significant difference between the methods in both tables since the results are less than 0.01 whereas "2018 testing data sets p-value" shows that there is no a difference since the obtained results are larger than 0.05.

	MAP			*Rprec*			*nDCG*		
	0.3	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**	**0.3**	**0.5**	**0.85**
2017 Training Title Reduction	0.0199	0.0365	0.0413	0.1212	0.0120	0.1530	0.0559	0.1785	0.0143
2017 Testing Title Reduction	0.0017	0.0800	0.0956	0.0949	0.0728	0.0480	0.0003	0.0113	0.1268
2018 Training Title Reduction	0.0003	0.0214	0.0141	0.1012	0.0039	0.0140	0.0002	0.0068	0.0161
2018 Testing Title Reduction	0.2089	0.6428	0.6796	0.2125	0.2886	0.2324	0.1409	0.2792	0.6943

Table 48: p-value for both data-sets using KLI method on Title Queries.

Generally speaking, it can be clearly seen from the tables that the two methods are significantly different particularly at value of 0.3. Therefore, observations indicate that as the retantion rate increases, the difference is decreased. Thus, the methods are different w.r.t some data-sets whereas the difference doesn't exist in some data such as 2018 testing data. KLI shows more interesting insights about the significant difference as opposed to IDF-r. Furthermore, comparing these p-values we obtained using IDF-r and KLI to p-values obtained in the previous chapter, we have come to the conclusion that using reduction methods provide considerable insights about the significant difference, that is, there is a significant difference among the methods which previous results could not produce(see Tables 47,48,and 26 respectively).

4 Conclusion

The findings have shown that the KLI method outperforms IDF-r and other methods and presents the best accuracy than IDF-r when compared to other methods,namely, BM25 and TF-IDF. As expected, IDF-r presents the lowest results than the other methods. IDF-r showed better results when compared to the previous results ,(without reduction), on training data-sets whereas presenting worse results on testing data-sets. Moreover, fusion algorithms performs very well on 2018 testing data while presenting worse on 2017 testing data. KLI and IDF-r perform very well on testing data-sets whereas they perform poorly on

training data.

References

1. Bun, K.K., Ishizuka, M.: Emerging topic tracking system. Web Intelligence: Research and Development Lecture Notes in Computer Science p. 125–130 (2001). https://doi.org/10.1007/3-540-45490-x$_1$3
2. Carvalho, D.S., Tran, V.D., Tran, V., Nguyen, L.: Improving legal information retrieval by distributional composition with term order probabilities. In: COLIEE 2017. 4th Competition on Legal Information Extraction and Entailment, held in conjunction with the 16th International Conference on Artificial Intelligence and Law (ICAIL 2017) in King's College London, UK. pp. 43–56 (2017), http://www.easychair.org/publications/paper/347227
3. Hiemstra, D., Robertson, S., Zaragoza, H.: Parsimonious language models for information retrieval. Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR 04 (2004). https://doi.org/10.1145/1008992.1009025
4. Koopman, B., Cripwell, L., Zuccon, G.: Generating clinical queries from patient narratives. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR 17 (2017). https://doi.org/10.1145/3077136.3080661
5. Koopman, B., Zuccon, G.: Relevation! : an open source system for information retrieval relevance assessment. In: ACM SIGIR 2014 : The 37th Annual ACM Special Interest Group on Information Retrieval. pp. 1243–1244. Gold Coast Convention and Exhibition Centre, Queensland, Australia (2014). https://doi.org/10.1145/2600428.2611175, https://eprints.qut.edu.au/72102/
6. Kumaran, G., Carvalho, V.R.: Reducing long queries using query quality predictors. Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval - SIGIR 09 (2009). https://doi.org/10.1145/1571941.1572038
7. Lee, J.: Combining multiple evidence from different properties of weighting schemes. In: Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval. pp. 180–188. SIGIR '95, ACM (1995)
8. Levene, M.: Search engines: Information retrieval in practice. The Computer Journal **54**(5), 831–832 (2011)
9. Locke, D., Zuccon, G., Scells, H.: Automatic query generation from legal texts for case law retrieval. Information Retrieval Technology Lecture Notes in Computer Science p. 181–193 (2017). https://doi.org/10.1007/978-3-319-70145-5$_1$4
10. Soldaini, L.: Quickumls : a fast , unsupervised approach for medical concept extraction (2016)
11. Tomokiyo, T., Hurst, M.: A language model approach to keyphrase extraction. Proceedings of the ACL 2003 workshop on Multiword expressions analysis, acquisition and treatment - (2003). https://doi.org/10.3115/1119282.1119287