Chapter 4

Result Analysis

In this section, we analyse the experiment results that we obtained and analyse the underlying reasons for the observation.

4.1 Overall performance

As mentioned before, we applied both binary judgement method, precision at 10, as well as graded judgement, NDCG at 10 to examine the overall performance of different algorithms. We used graded judgement system in our experiment, so the scores given to each document with respect to a query is in the range from 0 to 5. Documents given score zero are irrelevant to the topic while documents whose scores are larger than zero are relevant to the topic regardless of the degree of relevance. So the overall precision at 10 turned out to be shown in Figure 4.1. The observation shows that the differences between the precsion@10 distributions of these algorithms are not that significant. We conducted a pared t-test between the baseline algorithm *LRI*(Local Re-indexing), with other algorithms. The result is displayed in Table **??**.

The largest **t** we got is 6.5063.

Because the system execute merging algorithms based on documents that is already ranked highly by the original search engines, that is top 10 documents of the original

	GDS_TSS	GDS_TS	GDS_SS	GDS_DTSS	LMS	SRR	PRR	SPRR	MW	LRI
GDS_TSS	—	NH	t=2.8873, df=43	NH	NH	t=4.7466, df=43	t=4.4734, df=43	t=3.7726, df=43	NH	NH
GDS_TS		—	t=3.2307, df=43	NH	NH	t=4.699, df=43	t= 4.4148, df = 43	t= 3.8922, df= 43	NH	NH
GDS_SS			—	t=-3.0256, df = 43	t=-3.1842, df =43	NH	NH	NH	NH	t=-3.4549, df=43
GDS_DTSS				—	NH	t=4.8382, df=43	t=4.5063, df=43	”t=3.8844, df=43	NH	NH
LMS					—	t=4.7543, df=43	t=4.4829, df=43	t=3.9019, df=43	NH	NH
SRR						—	NH	t=-3.51, df=43	t=-3.7406, df=43	t=-7.2133, df=43
PRR							—	t=-2.6575, df=43	t=-3.7114, df=43	t =-6.6232, df=43
SPRR								—	t=-2.8994, df=43	t=-5.9037, df=43
MW									—	NH
LRI										—

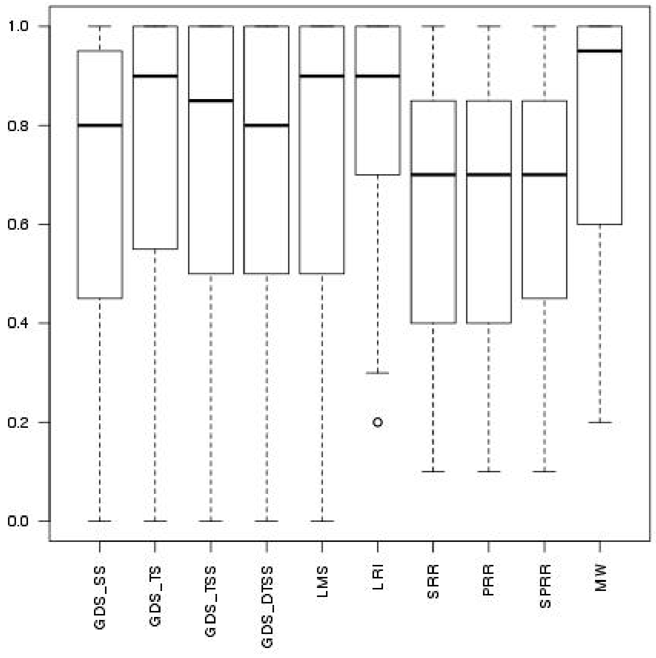
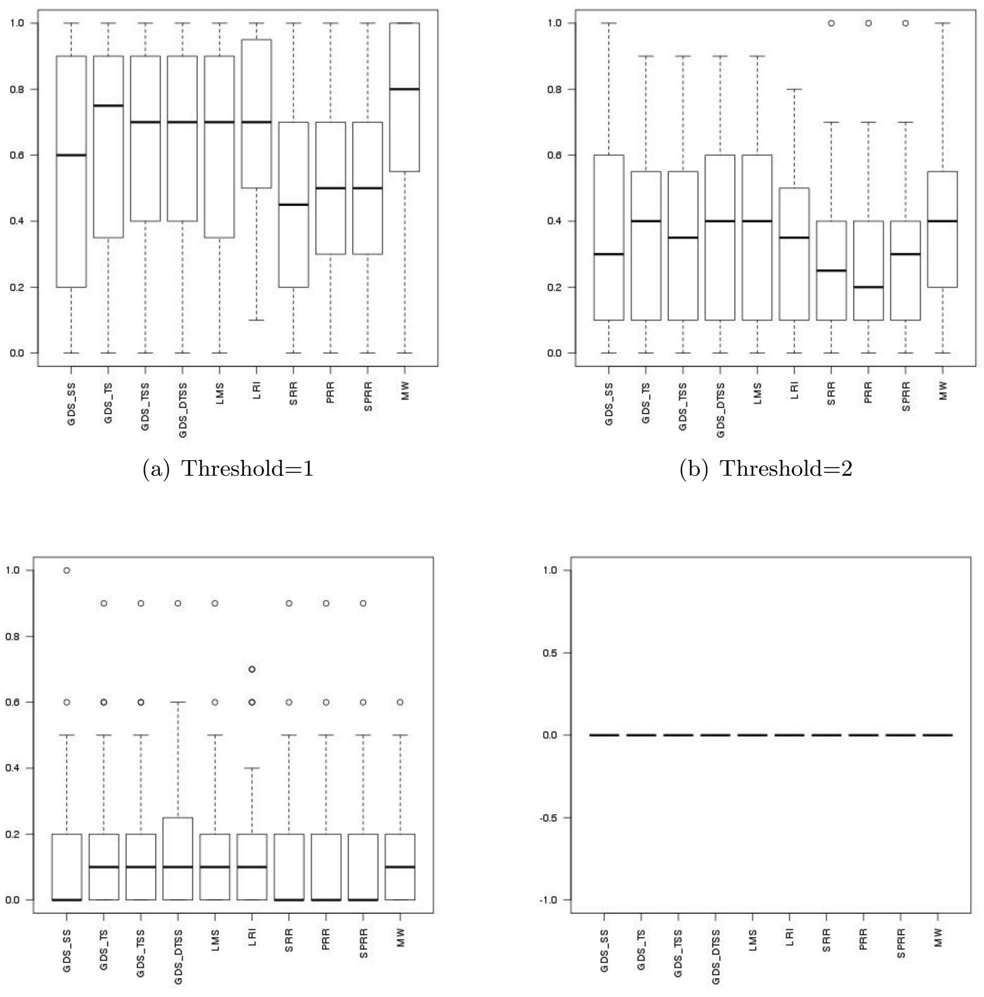


Figure 4.1: precision@10

	GDS_TSS	GDS_TS	GDS_SS	GDS_DTSS	LMS	SRR	PRR	SPRR	MW	LRI
GDS_TSS	—	p=0.2849241, d=0.164888	p=0.2849241, d=0.164888	p=0.6570079, d=0.3138873	p=0.3382643, d=0.1880336	p=2.310635e-05, d=-0.3719031	p=5.558954e-05, d=-0.3399154	p=0.0004892817, d=-0.2528957	p=0.1742379, d=0.1083292	p=0.4817434, d=0.2449329
GDS_TS		—	p=0.00236984, d=-0.1806192	p=0.3820464, d=0.2059694	p=0.9073363, d=0.4548726	p=2.695421e-05, d=-0.3664063	p=6.6984e-05, d=-0.3329079	p=0.0003407972, d=-0.2682688	p=0.1011739, d=0.05668385	p=0.718813, d=0.3407105
GDS_SS			—	p=0.004178282, d=-0.1519879	p=0.002699161, d=-0.1741874	p=0.1019622, d=0.05733364	p=0.2573317, d=0.1521309	p=0.6824074, d=0.3246332	p=0.07800444, d=0.03465159	p=0.001250152, d=-0.2110751
GDS_DTSS				—	p=0.4223028, d=0.2219145	p=1.716752e-05, d=-0.3824303	p=5.005215e-05, d=-0.3438145	p=0.0003489795, d=-0.267274	p=0.1188498, d=0.07107689	p=0.5209916, d=0.2600118
LMS					—	p=2.254158e-05, d=-0.372781	p=5.393764e-05, d=-0.3410388	p=0.0003309153, d=-0.2695003	p=0.09512011, d=0.05130356	p=0.731365, d=0.3464438
SRR						—	p=0.1272298, d=0.077384	p=0.001065238, d=-0.2184404	p=0.0005385612, d=-0.2487635	p=6.327192e-09, d=-0.6193184
PRR							—	p=0.01100594, d=-0.09882246	p=0.000587745, d=-0.2449703	p=4.531094e-08, d=-0.5663733
SPRR								—	p=0.005865978, d=-0.1340298	p=5.046627e-07, d=-0.4967929
MW									—	p=0.1560281, d=0.09709265
LRI										—

collections, the final precision of top 10 documents tends to be more higher than a traditional information system, as shown by Figure 4.1. According to Equation 3.10 , the precision is determined by the number of relevant documents in top 10 documents instead of the similarity of the ranking to a "ideal ranking", as a result, the precision at 10 ranges from 0 to 1 at a 0.1 interval. This turns out to be a reason why the distribution of all the algorithms are quite similar to each other. And to amplify the difference, we tried to increase the threshold of the relevance. The term *threshold* here means documents under which score would be classified as irrelevant documents. Figure 4.2 are the distribution of precision at 10 when the threshold were setted to 1 to 4, respectively.



(a) Threshold=1

(b) Threshold=2

(c) Threshold=3

(d) Threshold=4

Figure 4.2: Precision at different thresholds

NDCG is a graded measurement that we used in our experiment which calculate a score by accumulate the difference with respect to the ideal ranking. In that case, the method cares about the relative order of documents. The NDCG@10 result is shown is Figure 4.3

The LRI (Local Re-index) method fetch the original web pages that were retrieved and treat them as a document collection. The method then use an traditional indexing

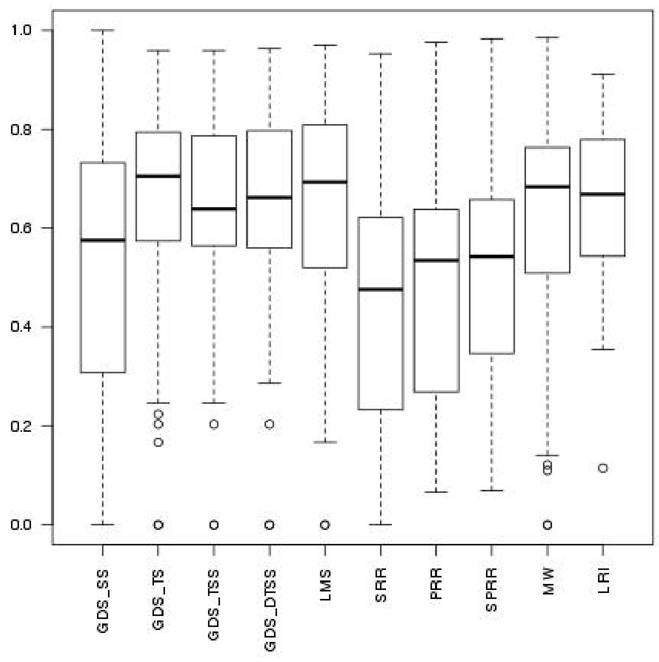


Figure 4.3: NDCG@10 of all algorithms

Table 4.1: Paired T-test result

GDS_SS	GDS_TS	GDS_TSS	GDS_DTSS	LMS	SRR	PRR	SPRR	MW
2.8693	0.3151	0.6113	0.5477	0.3599	4.4313	4.0654	3.4918	-0.1644

approach on the document collection as what we usually do in a centralised indexing information system. The documents were then ranked based on the scores that were calculated by the similarity between the query and the document text. So the performance of LRI methods is totally depended on the performance of the indexing method as well as the accuracy of the scoring method. If a distributed information retrieved system is no worse than an index centralised information retrieval system, it is considered as an acceptable method. So the LSI is regarded as baseline method for all other methods as a top performance regardless of its efficiency in the aspect of time and space. It is also quite obvious that the LRI methods outperforms others in most of the cases as we can see in both precison@10 and NDCG@10.

We use a pure round-robin method in the experiment and two other prioritized round-robin methods to give different to different collections to rank documents in an order of collection importance. The figures shown that the overall performance of all the round-robin method performs the worst among all other algorithms.

4.2 Round-Robin

Round-robin was the most fundamental mechanical that has been widely used in the early meta-search system. As it has been stated, round-robin method is based on the hypothesis that documents are distributed evenly in different collections, which means that if all

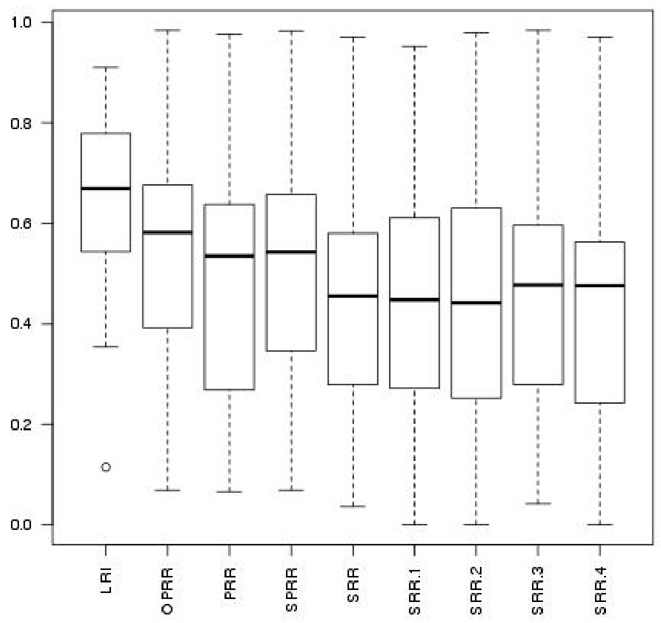


Figure 4.4: NDCG@10 of Simple Round Robin on 5 runs

the collections have an equally distribution of documents according to their relevance to the query, the method can gain an excellent result. However, due to the fact that the collection may have different weights on a particular topic in reality, and the situation when documents are evenly distributed in collections seldom happens. As a result, round-robin method can not outperform others in most cases, and since the rank of documents in each cycle is random, the performance of the methods various in different runs. We run the round-robin algorithm five times and compare them with the LRI method as well as two other prioritized round-robin methods. Also, to test the optimized performance of round-robin, we designed an optimized round-robin method($OPRR$), which use the final relevance score judged by assessors as a criteria of collection weight and run a collection weight based prioritized round-robin. The results are displayed in Figure 4.2. As been stated, round-robin performs quite randomly in each run. The underlying reason is that documents were ranking randomly in round-robin cycle. But, however, a phenomenon that we can observe from the Figure 4.2 is that although these round-robin runs perform differently, the variation is not significant. The reason is because that we documents that lying in the top 10 list are pretty much the same in each run, and in that case, the difference is the relative order of these documents.

Two other prioritized round-robin methods are also evaluated in the experiments. The original intention of both two algorithms is to rank collections in an order of relevance at the first place and then rank document in order of the collection score in each round-robin cycle. The PPR use the document length, namely the number of document retrieved, as an indication of collection relevance. The second prioritized round-robin method get the collection weight using the average similarity of the retrieved documents. This method is

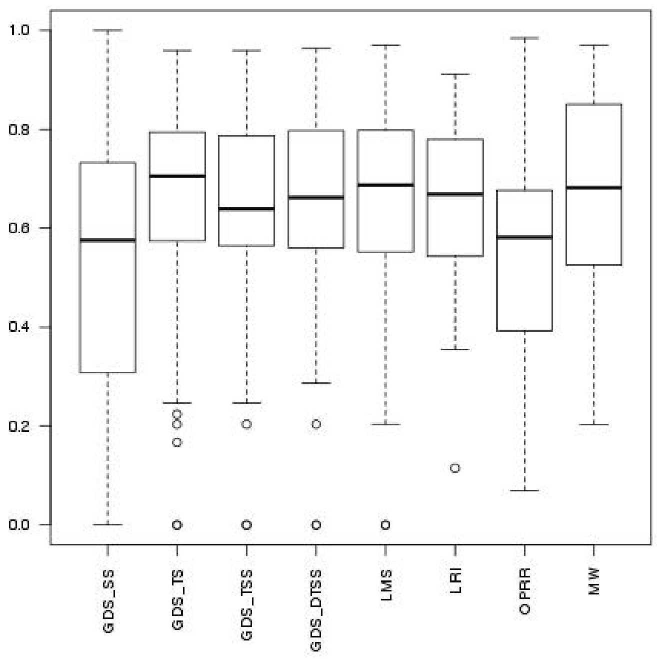


Figure 4.5: NDCG@10 of OPRR with respect to other algorithms

similar to a previous algorithm named *TopSRR* [14], which use title and summary fields of top k documents to calculate the collection score. The original TopSSR method initialized two vectors TV_j and SV_j by merging all the title and summary fields together respectively and then use Equation to get the collection score.

$$S_j = c * Similarity(Q, TV_j) + (1 - c) * (Q, SV_j) \tag{4.1}$$

In our method, we applied an average function on the similarity of all the top k documents that is retrieved. We then use the aggregated document similarity as the collection score in the round-robin method. As we can observe from Figure **??**, both two algorithms have promote the median of NDCG@10 to some degree. But they are neither optimized round-robin method as we can see from Figure 4.2.

Even if we optimized the round-robin by providing an ideal collection score, the method is still not comparable to most of other merging algorithms, as shown in Figure 4.5 The reason is round-robin methods itself is not a suitable method, which rank documents from different collections evenly in a result. Saying we have 10 collections, one of which contains many relevant documents that would be ranked at top 10 in an ideal ranking and none of the remaining collections contain relevant documents. Using a round-robin method, even we rank the first document of the most relevant collection at the top of the final rank, all the other remaining nine documents in the list are irrelevant. As a result, the accuracy of the documents ranking compared to the ideal ranking is quite low.

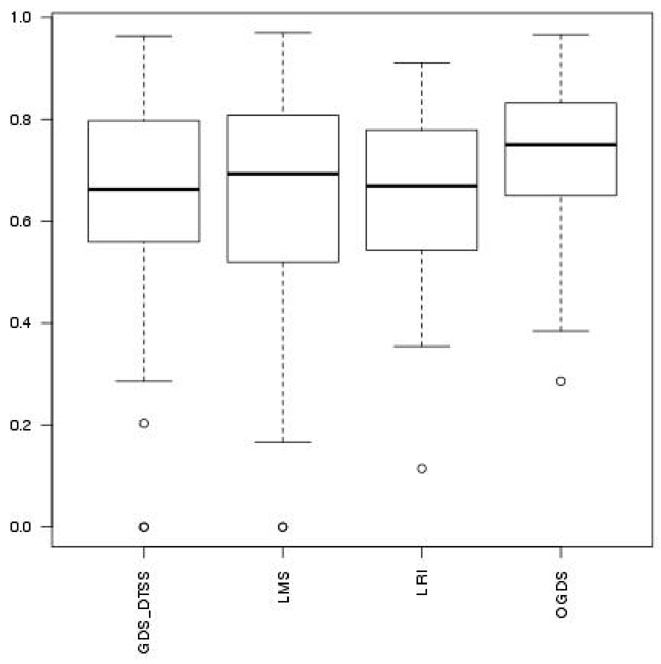


Figure 4.6: NDCG@10 of OGDS with respect to other algorithms

4.3 Evidence of Collection Weight

Collection weight is always taken into consideration when doing a meta-search engine. The collection weight reveals the significance of a collection with respect to a topic. In an uncooperative environment, there exists limited information on the collection.

4.3.1 Document Length

In our experiments, the limited information can be obtained from the result page. Number of documents retrieved from the search engine is usually accessible in the result page. In LMS, the document length, has been considered as the factor that represent the collection weight. We combined the collection weight get by the LMS with other score based algorithms, here we used the GDS_DTSS method. As we can see from Figure 4.3, the LMS has increased the median of all the NDCG@10 across all the queries. But the total distribution has not been changed so much. Also, in the last section, we used LMS to get a collection weight in the prioritized algorithm PRR, and get the conclusion that LMS method cannot guarantee the optimized round-robin. Again, we used the method in previous section to get an optimized collection weight and combine it with the GDS_DTSS, and compare it with the LMS algorithm.

Figure 4.6 illustrates the importance of collection score, if we can add an ideal collection score to the algorithm, it can be improved in a large scale. The OGDS (Optimized GDS) method, has improved the performance of the original GDS and even better than the LRI method. We examine how much difference from the weighted algorithms to the original by applying a Paired T-test(Table 4.2).

Table 4.2: t-value with respect to GDS

LMS:GDS_DTSS	OGDS:GDS_DTSS	LMS:OGDS
0.8102	4.4879	-3.4213

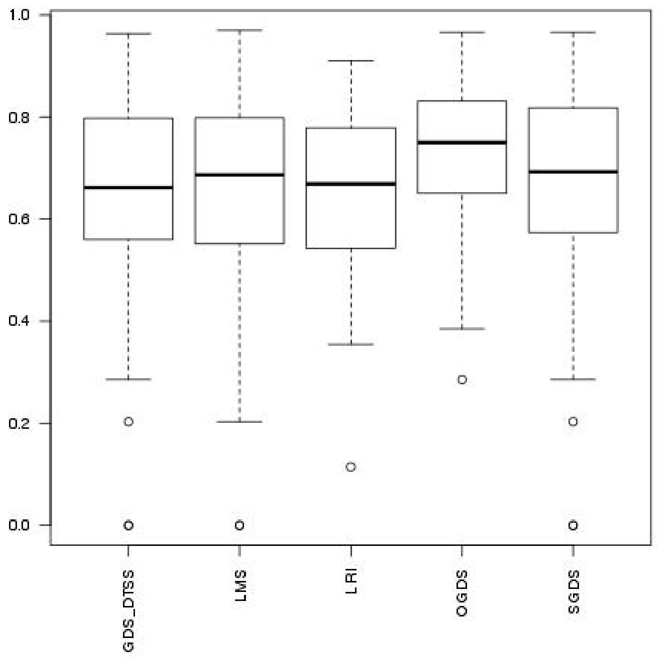


Figure 4.7: NDCG@10 of SGDS with respect to other algorithms

It shows that the LMS is much less than optimized. The sizes of document collections themselves are various from each other in certain degree, so the number of documents retrieved by these collections are also influenced by the total number of documents as well as the precision of those search engines. For example, some search engines with a bad retrieval algorithm may return many inaccurate and irrelevant documents while some other search engines with high precision return a limited number of documents that is very relevant to the topic. So document length would not be a sufficient evidence of collection significance.

4.3.2 Average document similarity

Another method of getting the collection is by average the similarity of the text fields of all documents in the first result page with the topic. These text fields, in some way, is a sample of the document collection, which can reveal some information on the particular collection.

Table 4.3: T-test result for SGDS

LMS:GDS_DTSS	OGDS:GDS_DTSS	OGDS:LMS	SGDS:LMS	SGDS:OGDS
0.8102	4.4879	3.4213	0.9462	-3.4031

Compared to the document length based collection scoring method, LMS, the performance of average document similarity method is more close to the optimized collection weight.

4.4 Value of document fields

The GDS algorithms estimate the document collection score by a general scoring function using the text field provided by the result page. We used two fields, title and summary in our experiment, as text fields. We used four methods to estimate the document score,

- **GDS_TS** Scoring function uses the title field only.
- **GDS_SS** Scoring function uses the summary field only
- **GDS_TSS** Scoring function uses the title field as the first priority, if score is not usable, use the summary field
- **GDS_DTTS** Scoring function uses a combination of title field as well as summary field

The performance described by Figure 4.3 shows that GDS_SS performs the worst among all other general scoring functions, while *GDS_TS* performs the best. The phenomenon demonstrates the fact that the title may be more valuable than summary in representing the main topic of a document. Of all the 44 queries that we evaluated, GDS_SS method can perform better than an average performance in 16 queries while *GDS_TS* can outperform others in most of the queries, making 33 of all.

4.5 Parameter setting of multi-weight method

To this point, we have taken four factors into our algorithms to get the final comparable document scores to rank the documents. The final method is named multi-weight because we took all these factors into consideration when calculating the document score. As we described in the experiment section, we set 4 weights to these four factors respectively and summarize the weighted factors. We test the performance of this method by a 4-fold cross validation.