Drexel at TREC 2014 Federated Web Search Track

Haozhen Zhao
College of Computing and Informatics
Drexel University
Philadelphia, PA 19104, USA
haozhen.zhao@drexel.edu

Xiaohua Hu
College of Computing and Informatics
Drexel University
Philadelphia, PA 19104, USA
xh29@drexel.edu

ABSTRACT

This paper reports our participation in the Federated Web Search Track in TREC 2014. We submitted 21 runs for all the three tasks: Vertical Selection (7), Resource Selection (7) and Results Merging (7). Our main purpose is to test several established resource selection methods on the new realistic FedWeb test collections. We evaluated 7 well known resource selection methods for the vertical selection and resource selection tasks. The effectiveness of these methods in the RS tasks does not carry to the VS tasks, which implies that more sophisticated algorithms and more diverse sources of evidence are needed for solving the VS task effectively. Our Results Merging experiments reveal the correlation between the performance of RM and the performance of its input RS results.

1. INTRODUCTION

Federated Web Search is the task of searching multiple search engines simultaneously and combining their results in a coherent way for presenting to the end user. The Federated Web Search Track 2014 (FedWeb 2014), with its precedent, FedWeb 2013 [4], features realistic web test collections for the federated web search task. In addition to the Resource Selection (RS) and Results Merging (RM) tasks in FedWeb 2013, FedWeb 2014 introduced a new task, the Vertical Selection (VS) task.

This is our first participation in the Federated Web Search track. In this year's tasks, our main purpose is to evaluate several established resource selection methods on the new Federated Web Search test collections. Though our focus is on the RS task, we also submitted runs for the VS and RM tasks.

2. RESOURCE SELECTION IN FEDERATED SEARCH

In a federated search environment, it is generally desirable to query only a subset of all the available resources. Often, this is considered from efficiency point of view, as a selective search strategy generally means quicker search response and lower latency. Moreover, a recent study shows that search effectiveness would not be reduced even when searches are conducted selectively, in particular given the sources are partitioned or distributed properly[5]. The goal of RS is then, for a given query, to select only the most promising search engines from all those available.

Most existing methods for RS can be categorized into large document approaches, small document approaches, or classification based approaches [6]. In our experiments, we employ several small document approaches for Resource Selection task. Small document approaches rely on a centralized sample index (CSI) of the all the sampled documents from each sources. For a given query, search results on CSI are used to estimate the score of a particular resource. Different small document approaches vary in terms of how they use the search results. The following methods are used in our experiments.

2.1 ReDDE

ReDDE proposed by Si and Callan is arguably the most influential small document approach for resource selection[11]. For a given query, ReDDE estimates the quality of resources based on how relevant documents are distributed in the search results from the CSI. Generally, top k ranked documents are assumed to be relevant. Given sample S and its source resource R, ReDDE assumes each document in the sample represents $\frac{|R|}{|S|}$ documents in the source, where $|R|$, $|S|$ are the sizes of R and S respectively. It should be noted that in the original ReDDE, each document of the sampled index represents a fixed score for the source document. The score for a given resource is calculated by counting the number of documents from it in the top k search results, and then times the scaling factor $\frac{|R|}{|S|}$:

$$\text{ReDDE}(R|q) = \frac{|R|}{|S|} \cdot \sum_{i=1}^{k} \mathcal{I}(d_i \in R). \quad (1)$$

Later, ReDDE.top [1] is proposed by Arguello to replace the fixed score with the actual retrieval score of a document in the search result:

$$\text{ReDDE.top}(R|q) = \frac{|R|}{|S|} \cdot \sum_{i=1}^{k} \mathcal{I}(d_i \in R)\,\text{RSV}(d_i), \quad (2)$$

where $\text{RSV}(d_i)$ is the retrieval status value of d_i, e.g. $P(d_i|q)$ in the case of using language model as the retrieval model.

TREC 2014 Gaithersburg, Maryland, USA November 19–21, 2014

2.2 CRCS

The Central-Rank-based Collection Selection (CRCS) approach [10] proposed by Shokouhi uses the rank of a top k retrieved document to derive its contribution to the calculation of the relevance of a resource to the given query. It uses either a linear or a negative exponential function to convert the document rank to a score, which is then summed in a similar manner as ReDDE to determine the score of the resource. This results CRCSlinear and CRCSExp as two versions of the CRCS algorithm.

2.3 SUSHI and CiSS

Contrary to ReDDE and CRCS which use only rank information of sampled documents, SUSHI [12] and CiSS [9] used the actual relevance scores of the sampled documents to derived the relevance of the sources. SUSHI fits the scores of documents from a particular resource to a smooth curve, and ranks resources via maximizing certain metric, e.g. P@10. SUSHI intentionally selects fewer resources than ReDDE and CRCS methods. To score a resource, CiSS gathers documents belong to that resource in the search result list, and generates a new rank of them based on their relative order. Then the document scores and their new ranks are transformed using exponential function and logarithmic function respectively. A linear function is used to fit documents in the space with log-transformed ranks being the x-axis and exponentially transformed document scores being the y-axis. The resource score is then an integral over this curve.

3. DATASET AND RETRIEVAL SETUP

The FedWeb14 test collection, created by the University of Twente group, is used in this year Federated Web Search track [4, 8]. It consists of snippets and documents sampled from search result pages of 149 search engines. 4000 queries are used in building the sample set. As a part of the Vertical Selection task, search engines are categorized into 24 verticals, such as General, Video, Jobs, Academic, and so on. It is noted that each search engine belongs to only one vertical. Previous federated web search experiments generally run on dataset collection, customized by reusing existing IR test collections. The FedWeb13 and FedWeb14 test collections are crawled directly from different vertical search engines, making them more realistic. To our best knowledge, no work has been done to test established resource selection methods on them.

We created a centralized sample index (CSI) of all the sampled documents. Our index is built with the Indri Toolkit[1], using the Krovetz stemmer and not removing any stop words.

For both the VS and RS tasks, the inputs are generated through the following procedure: retrieval top 1000 documents from CSI for each topic. For the retrieval, we used two kinds of retrieval models and two kinds of query modeling. Of the retrieval models, one is BM25 retrieval model with $k = 1.2$ and $b = 0.75$, the other is language model with Dirichlet smoothing and $\mu = 1350$ which is about the average document length in the CSI. Of the query models, one uses the plain query terms (PlainQ), the other uses the Markov Random Field Model's sequential dependency query model (MRF-SD-Q) [7]. This results the following three set of retrieval results for the topics: plain query terms with language model (LM+PlainQ), plain query terms with Okapi BM25 retrieval model (BM25+PlainQ), MRF sequential dependence query model with language model (LM+MRF-SD-Q).

[1]http://www.lemurproject.org/indri.php

4. EXPERIMENTS AND RESULTS

4.1 Resource Selection

The purpose of the RS task is to predict the quality of individual resources for given topics. It is required that all the resources should be ranked for a given search topic, with more relevant resources being ranked higher. Our RS procedure used the following seven RS methods: ReDDE, ReDDE.top, CRCSLinear, CRCSExp, CiSS, CiSSAprox, SUSHI. All of these small document RS approach have reference implementations in the LiDR library[2] by Ilya Markov [6]. It is noted that many of these algorithms require the size of the resource to approximate the complete ranking with the sampled search results. In our case, size of most involved search engines are not available, therefore we took a bold assumption on the approximation issue by setting the proportion of resource size to sample size to a constant for all resources such that it would not affect the ranking of resources.

With the 3 retrieval setups detailed in Section 3 and 7 RS methods introduced in Section 2, there are 21 RS run settings in total. We first run all our settings on the FedWeb13 collection, and then choose the top 7 run settings for our FedWeb14 submissions.

Table 1 shows our submitted results:

runID	nDCG@20	nDCG@10	nP@1	nP@5
drexelRS1	0.389	0.348	0.222	**0.318**
drexelRS2	0.328	0.227	0.125	0.180
drexelRS3	0.333	0.229	0.125	0.179
drexelRS4	0.333	0.229	0.125	0.180
drexelRS5	0.342	0.241	0.135	0.211
drexelRS6	0.382	0.284	0.201	0.250
drexelRS7	**0.422**	**0.359**	**0.293**	0.314

Table 1: Resource Selection Results

drexelRS1 : LM+PlainQ+CRCSExp
drexelRS2 : LM+PlainQ+ReDDE
drexelRS3 : LM+PlainQ+CiSSAprox
drexelRS4 : LM+PlainQ+CiSS
drexelRS5 : BM25+PlainQ+CRCSLinear
drexelRS6 : LM+MRF-SD-Q+ReDDETop
drexelRS7 : LM+MRF-SD-Q+SUSHI

nP@1 and nP@5 are the normalized graded precision measures introduce in [4].

Based on our submitted results, SUSHI with language model and sequential dependency queries performs the best among all the submitted settings in terms of nDCG@20, nDCG@10 and nP@1. CRCSExp with language model and plain queries performs best in terms of nP@5.

A query by query comparison between the best performed runs, drexelRS7 and drexelRS1, shows that even though drexelRS7 outperforms drexelRS1 in nDCG@20, both of the two outperforms the other in half of the topics (Figure 1).

With the released RS qrels data, we analyzed all our 21 runs and report nDCG@20 and nDCG@10 for all the 21

[2]https://github.com/markovi/LiDR

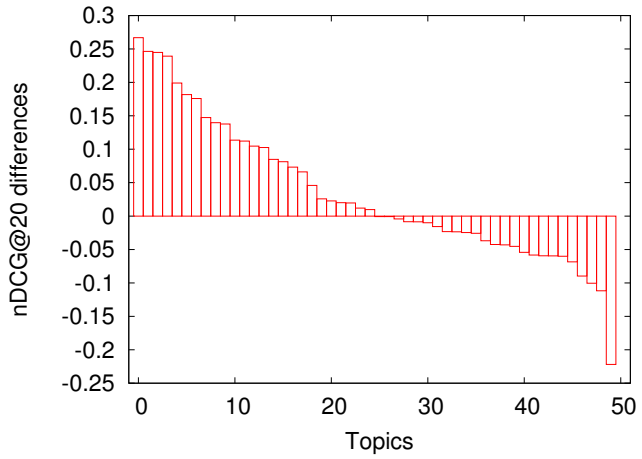

Figure 1: nDCG@20 differences between drexelRS7 and drexelRS1 among topics; positive bars indicate drexelRS7 works better for that topic and negative bars worse.

runs in Table 2. For the three retrieval settings, SUSHI performs best in two of them, and CiSSApprox performs best in the rest BM25 retrieval model setting. The performance of CRCS related methods is more robust across different setups than others, which is consistent with earlier findings in [13].

runID	nDCG@20	nDCG@10
mrfsd-lm-CRCSExp	0.3911	0.3450
mrfsd-lm-CRCSLinear	0.3618	0.2492
mrfsd-lm-CiSS	0.3487	0.2287
mrfsd-lm-CiSSApprox	0.3496	0.2289
mrfsd-lm-ReDDE	0.3464	0.2287
mrfsd-lm-ReDDETop	0.3821	0.2844
mrfsd-lm-SUSHI	**0.4224**	0.3591
plain-lm-CRCSExp	0.3889	0.3477
plain-lm-CRCSLinear	0.3498	0.2406
plain-lm-CiSS	0.3325	0.2289
plain-lm-CiSSApprox	0.3325	0.2288
plain-lm-ReDDE	0.3276	0.2268
plain-lm-ReDDETop	0.3452	0.2424
plain-lm-SUSHI	**0.4047**	0.3163
plain-bm25-CRCSExp	0.3796	0.3238
plain-bm25-CRCSLinear	0.3423	0.2414
plain-bm25-CiSS	0.3858	0.2927
plain-bm25-CiSSApprox	**0.4095**	0.3153
plain-bm25-ReDDE	0.3405	0.2307
plain-bm25-ReDDETop	0.3479	0.2349
plain-bm25-SUSHI	0.3336	0.2422

Table 2: Performance of all 21 RS runs

More in-depth study can be done to investigate the contributions of different factors, i.e. query model, retrieval model, and RS algorithm, to the differences in IR metrics.

4.2 Vertical Selection

In web search, verticals can be defined by topic, e.g. weather, sports, etc., or by media type, e.g. image, video, etc., or by genre of content, e.g. news, blogs, encyclopedia, etc. The user's query may have a strong indication of vertical intent, e.g. "arrow icon", which is clearly oriented to the image vertical, or is intrinsically ambiguous, e.g. "Barack Obama", which may be associated with verticals such as encyclopedia, news, general web and so on. In these scenarios, presenting search results from multiple relevant verticals is desirable and would improve users' satisfaction of the search service.

The task of vertical selection is to predict and rank the verticals for a given query. A vertical is relevant to a query can be interpreted in two senses. First, the vertical is overall aligned to the user's search intent. Second, the vertical has many relevant documents for the user's query. Zhou et. al. recently empirically showed that the two correlate well with each other. Therefore, the ground truth relevant vertical sets can be determined based on the vertical collection relevance[14]. The source of evidences for vertical selection may include query string, vertical-representative corpora, and query log associated with the vertical and so on[2].

In this year's work, we approach the VS task in the same way as the RS task. Each vertical is treated as a single resource; all the returned results belong to the resources of a particular vertical are treated as being from the same source. Then the general resource selection procedures are applied on these verticals. Because in the vertical selection task, only a subset of verticals should be returned, we therefore applied a threshold in selecting only the top verticals. With the normalized scores of verticals for each query, we set a cutoff threshold only selecting verticals that by selecting which the discounted gain is beyond the threshold. In the submitted runs, this threshold value is set to 0.01. Table 3 shows the performance of our submitted runs.

runID	Precision	Recall	F1
drexelVS1	**0.240**	0.506	**0.284**
drexelVS2	0.159	0.824	0.233
drexelVS3	0.134	0.960	0.212
drexelVS4	0.134	0.960	0.212
drexelVS5	0.163	0.824	0.244
drexelVS6	0.171	0.729	0.251
drexelVS7	0.189	0.732	0.271

Table 3: Vertical Selection Results

drexelVS1 : LM+PlainQ+CRCSExp
drexelVS2 : LM+PlainQ+ReDDE
drexelVS3 : LM+PlainQ+CiSSAprox
drexelVS4 : LM+PlainQ+CiSS
drexelVS5 : BM25+PlainQ+CRCSLinear
drexelVS6 : LM+MRF-SD-Q+ReDDETop
drexelVS7 : LM+MRF-SD-Q+SUSHI

CRCSExp with language model and plain query achieved the highest precision and F1 scores. Overall, our approach are among the medianly performed submissions, perhaps due to to relatively low precision. With the release of the qrels for VS, we investigated whether increasing the cut-off threshold for VS will increase F1 score. Figure 2 shows our results that sweep threshold value from 0.01 to 0.5. Some algorithms such as CiSS and CRCSLinear, witness an increase of F1 at some point, and many other algorithms do not. Our experiments indicated that naively treating verti-

cal selection task as a traditional resource selection task is not very effective.

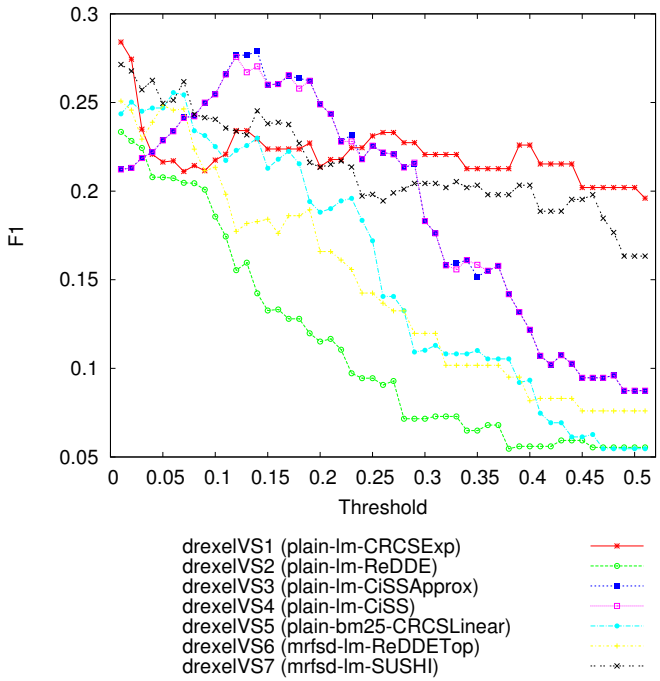

Figure 2: Change of F1 as threshold is changed from 0.01 to 0.51

4.3 Results Merging

The Results Merging (RM) task is to merge search result snippets from resources selected at the RS stage into a single rank ordered list. The track organizer provides topic search snippets from the 149 search engines for 75 topics. Therefore for each topic, there are 149 sets of snippets organized based on the resources, and for each resource there are 75 sets of snippets organized based on the topics. During the merging stage, only the top 20 resources can be selected as the sources of snippets to be merged. A baseline RS result is provided by the organizer and required to be the input of at least one submitted RM run.

There exist mainly two kinds of approaches of doing result merging: score based and rank based approaches. Previous researches show that rank-based approaches such as Reciprocal Rank Fusion (RRF) [3] generally outperforms score based approaches. In our case, there is no score information provided for the snippets, therefore rank-based approach becomes the natural choice.

Our solution to the result merging task is to leverage the reciprocal rank (RR) of a document as the basic retrieval status value (RSV) for a given snippet. For a given query q, the RR of a document d from the results of a resource R_i is given by:

$$\mathrm{RR}(d|q, R_i) = \frac{1}{k + r(d)} \tag{3}$$

where $r(d)$ is d's rank in the result list, and k is generally set to 60.

This score is further weighted based on the score or reciprocal rank of the selected resource. Document score weighted by selected resource score is:

$$\mathrm{Score}(d|q, R_i) = \mathrm{RS}(R_i|q) \times \mathrm{RR}(d|q, R_i) \tag{4}$$

where $\mathrm{RS}(R_i|q)$ is the score of resource R_i from the RS stage. Document score weighted by selected resource reciprocal rank is:

$$\mathrm{Score}_{\mathrm{rank}}(d|q, R_i) = \frac{c}{\mathrm{RS}_{\mathrm{rank}}(R_i|q)} \times \mathrm{RR}(d|q, R_i) \tag{5}$$

where $\mathrm{RS}_{\mathrm{rank}}(R_i|q)$ is the rank of resource R_i from the RS stage, and c is a constant.

The above score is used to output the final merged document ranking list for a given query. It should be noted, we did not consider duplication in the submitted runs.

Other than the runs based on the baseline resource list from the organizer, we submitted 5 runs based on our resource selection results. The final results are shown in Table 4; the runID prefix indicates its corresponding resource selection run, and the tailing W or R indicates whether it is based on resource score (W) or resource reciprocal rank (R).

From our results, we can see that the baseline resource list outperforms our RS results. With the qrels of the RS task, we find out the nDCG@20 and nDCG@10 for the baseline RS run is 0.428 and 0.372, respectively. For our best RS run drexelRS7, the nDCG@20 and nDCG@10 are 0.422 and 0.359, which is rather close to the baseline RS run. The nDCG@20 and nDCG@10 of their corresponding RM runs, FW14basemW and drexelRS7mW, are also very close. Therefore, there is a high possibility that performance of RM correlated with the performance of RS in our current methodology. More thorough analysis need to be done to confirm this conjecture.

Between the two weighting schemes, based on selected resource score or reciprocal rank, the latter generally performances better than the former.

5. CONCLUSION AND FUTURE WORK

We described here the 21 runs we submitted to the Federated Web Search track in TREC 2014. We evaluated 7 well known resource selection methods for the vertical selection and resource selection tasks. The effectiveness of these methods in the RS tasks does not carry to the VS tasks, which implies that more sophisticated algorithms and more diverse sources of evidence are needed for solving the VS task effectively. Our Results Merging experiments reveal the correlation between the performance of RM and the performance of its input RS results.

More in-depth and comprehensive analysis and comparison of the all the runs, including submitted, not submitted and post-mortem, are planned on the realistic and valuable FedWeb13 and FedWeb14 test collections.

6. REFERENCES

[1] J. Arguello, J. Callan, and F. Diaz. Classification-based resource selection. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1277–1286, New York, NY, USA, 2009. ACM.

[2] J. Arguello, F. Diaz, J. Callan, and J.-F. Crespo. Sources of evidence for vertical selection. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in*

runID	nDCG@20	nDCG@100	nDCG@20_wdup	nDCG@20_local	nDCG@100_local	nDCG-IA@20
FW14basemR	0.322	0.318	0.361	0.446	0.626	0.107
FW14basemW	0.260	0.298	0.312	0.367	0.592	0.086
drexelRS1mR	0.219	0.298	0.222	0.264	0.491	0.059
drexelRS4mW	0.144	0.244	0.148	0.177	0.420	0.036
drexelRS6mR	0.198	0.270	0.194	0.232	0.443	0.050
drexelRS6mW	0.196	0.270	0.193	0.231	0.444	0.049
drexelRS7mW	0.250	0.305	0.249	0.318	0.535	0.070

Table 4: Results Merging Task Results

Information Retrieval, SIGIR '09, pages 315–322, New York, NY, USA, 2009. ACM.

[3] G. V. Cormack, C. L. A. Clarke, and S. Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 758–759, New York, NY, USA, 2009. ACM.

[4] T. Demeester, D. Trieschnigg, D. Nguyen, and D. Hiemstra. Overview of the TREC 2013 federated web search track. In *TREC 2013*, 2013.

[5] A. Kulkarni and J. Callan. Topic-based index partitions for efficient and effective selective search. In *SIGIR 2010 Workshop on Large-Scale Distributed Information Retrieval*, volume 1, 2010.

[6] I. Markov. *Uncertainty in Distributed Information Retrieval*. PhD thesis, University of Lugano, 2014.

[7] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 472–479, New York, NY, USA, 2005. ACM.

[8] D. Nguyen, T. Demeester, D. Trieschnigg, and D. Hiemstra. Federated search in the wild: The combined power of over a hundred search engines. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1874–1878, New York, NY, USA, 2012. ACM.

[9] G. Paltoglou, M. Salampasis, and M. Satratzemi. Modeling information sources as integrals for effective and efficient source selection. *Information Processing & Management*, 47(1):18–36, Jan. 2011.

[10] M. Shokouhi. Central-rank-based collection selection in uncooperative distributed information retrieval. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 160–172, Berlin, Heidelberg, 2007. Springer-Verlag.

[11] L. Si and J. Callan. Relevant document distribution estimation method for resource selection. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 298–305, New York, NY, USA, 2003. ACM.

[12] P. Thomas and M. Shokouhi. SUSHI: scoring scaled samples for server selection. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 419–426, New York, NY, USA, 2009. ACM.

[13] K. Zhou, R. Cummins, M. Lalmas, and J. M. Jose. Evaluating reward and risk for vertical selection. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2631–2634, New York, NY, USA, 2012. ACM.

[14] K. Zhou, T. Demeester, D. Nguyen, D. Hiemstra, and D. Trieschnigg. Aligning vertical collection relevance with user intent. In *CIKM 2014*, 2014.