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 (VS), Resource Selection (RS) and Results Merging (RM). 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 source. 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 \(|R|\) 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 \(\frac{|R|}{|S|}\):

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

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:

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

where 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.
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.
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 derive 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 FedWebl4 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 FedWebl3 and FedWebl4 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-SDQ) [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 dependency query model with language model (LM+MRF-SDQ).

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:

<table>
<thead>
<tr>
<th>runID</th>
<th>nDCG@20</th>
<th>nDCG@10</th>
<th>nP@1</th>
<th>nP@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>drexelRS1</td>
<td>0.389</td>
<td>0.348</td>
<td>0.222</td>
<td>0.318</td>
</tr>
<tr>
<td>drexelRS2</td>
<td>0.328</td>
<td>0.227</td>
<td>0.125</td>
<td>0.180</td>
</tr>
<tr>
<td>drexelRS3</td>
<td>0.333</td>
<td>0.229</td>
<td>0.125</td>
<td>0.179</td>
</tr>
<tr>
<td>drexelRS4</td>
<td>0.333</td>
<td>0.229</td>
<td>0.125</td>
<td>0.180</td>
</tr>
<tr>
<td>drexelRS5</td>
<td>0.342</td>
<td>0.241</td>
<td>0.135</td>
<td>0.211</td>
</tr>
<tr>
<td>drexelRS6</td>
<td>0.382</td>
<td>0.284</td>
<td>0.201</td>
<td>0.250</td>
</tr>
<tr>
<td>drexelRS7</td>
<td>0.422</td>
<td>0.359</td>
<td>0.293</td>
<td>0.314</td>
</tr>
</tbody>
</table>

\(nP@1\) 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 outperform 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

\(^1\)http://www.lemurproject.org/indri.php

\(^2\)https://github.com/markovi/LiDR
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 verticals 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.

<table>
<thead>
<tr>
<th>runID</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>drexelVS1</td>
<td>0.240</td>
<td>0.506</td>
<td>0.284</td>
</tr>
<tr>
<td>drexelVS2</td>
<td>0.159</td>
<td>0.824</td>
<td>0.233</td>
</tr>
<tr>
<td>drexelVS3</td>
<td>0.134</td>
<td>0.960</td>
<td>0.212</td>
</tr>
<tr>
<td>drexelVS4</td>
<td>0.134</td>
<td>0.960</td>
<td>0.212</td>
</tr>
<tr>
<td>drexelVS5</td>
<td>0.163</td>
<td>0.824</td>
<td>0.244</td>
</tr>
<tr>
<td>drexelVS6</td>
<td>0.171</td>
<td>0.729</td>
<td>0.251</td>
</tr>
<tr>
<td>drexelVS7</td>
<td>0.189</td>
<td>0.732</td>
<td>0.271</td>
</tr>
</tbody>
</table>

Table 3: Vertical Selection Results

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-

Table 2: Performance of all 21 RS runs

<table>
<thead>
<tr>
<th>runID</th>
<th>nDCG@20</th>
<th>nDCG@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>mrfsl-lm-CRCSExp</td>
<td>0.3911</td>
<td>0.3450</td>
</tr>
<tr>
<td>mrfsl-lm-CRCSLinear</td>
<td>0.3618</td>
<td>0.2492</td>
</tr>
<tr>
<td>mrfsl-lm-CISS</td>
<td>0.3487</td>
<td>0.2287</td>
</tr>
<tr>
<td>mrfsl-lm-CISSApprox</td>
<td>0.3496</td>
<td>0.2289</td>
</tr>
<tr>
<td>mrfsl-lm-ReDDE</td>
<td>0.3464</td>
<td>0.2287</td>
</tr>
<tr>
<td>mrfsl-lm-ReDDETop</td>
<td>0.3821</td>
<td>0.2844</td>
</tr>
<tr>
<td>mrfsl-lm-SUSHI</td>
<td><strong>0.4224</strong></td>
<td>0.3591</td>
</tr>
<tr>
<td>plain-lm-CRCSExp</td>
<td>0.3889</td>
<td>0.3477</td>
</tr>
<tr>
<td>plain-lm-CRCSLinear</td>
<td>0.3498</td>
<td>0.2406</td>
</tr>
<tr>
<td>plain-lm-CISS</td>
<td>0.3325</td>
<td>0.2289</td>
</tr>
<tr>
<td>plain-lm-CISSApprox</td>
<td>0.3325</td>
<td>0.2288</td>
</tr>
<tr>
<td>plain-lm-ReDDE</td>
<td>0.3276</td>
<td>0.2268</td>
</tr>
<tr>
<td>plain-lm-ReDDETop</td>
<td>0.3452</td>
<td>0.2424</td>
</tr>
<tr>
<td>plain-lm-SUSHI</td>
<td><strong>0.4047</strong></td>
<td>0.3163</td>
</tr>
<tr>
<td>plain-bm25-CRCSExp</td>
<td>0.3796</td>
<td>0.3238</td>
</tr>
<tr>
<td>plain-bm25-CRCSLinear</td>
<td>0.3423</td>
<td>0.2414</td>
</tr>
<tr>
<td>plain-bm25-CISS</td>
<td>0.3858</td>
<td>0.2927</td>
</tr>
<tr>
<td>plain-bm25-CISSApprox</td>
<td><strong>0.4095</strong></td>
<td>0.3153</td>
</tr>
<tr>
<td>plain-bm25-ReDDE</td>
<td>0.3405</td>
<td>0.2307</td>
</tr>
<tr>
<td>plain-bm25-ReDDETop</td>
<td>0.3479</td>
<td>0.2349</td>
</tr>
<tr>
<td>plain-bm25-SUSHI</td>
<td>0.3336</td>
<td>0.2422</td>
</tr>
</tbody>
</table>

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 verticals 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.

Table 3: Vertical Selection Results

drexelVS1 : LM+PlainQ+CRCSExp
drexelVS2 : LM+PlainQ+ReDDE
drexelVS3 : LM+PlainQ+CiSSApprox
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-
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 task 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 of the all the runs, including submitted, not submitted and post-mortem, are planned on the realistic and valuable methodologies. 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


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

Table 4: Results Merging Task Results


