University of Padua at TREC 2014: Federated Web Search Track

Emanuele Di Buccio and Massimo Melucci

Department of Information Engineering, University of Padua, Italy
{dibuccio,melo}@dei.unipd.it

Abstract. This paper reports on the participation of the University of Padua to the TREC 2014 Federated Web Search track. The objective was the experimental investigation of the TWF·IRF weighting framework for resource and vertical selection in Federated Web Search settings.

1 Introduction

This paper reports on the participation of the Information Management System (IMS) Research Group of the University of Padua to the TREC 2014 Federated Web Search track (FedWeb14).[1] The participation to the FedWeb14 track aimed at the investigation of the effectiveness of the TWF·IRF weighting framework when adopted in Federated Web search setting; in particular, the participation to the second edition of the track allowed us to complement the experimental investigation carried out last year when our focus was on the effectiveness of the weighting scheme for resource selection [4, 5]. In the resource selection task, given a set of search engines and a query, the objective is to select the most promising search engines to which the query will be forwarded.

In FedWeb14, a new test collection was adopted and a new task was added: *vertical selection*. This task considers an additional resource level, i.e. the *vertical* level, where a vertical is a subset of the entire set of search engines. The vertical level can be considered as a categorization of the set of search engines; examples of verticals are Academic, Jobs, and Video search engines. In FedWeb14, the set of verticals was a partition over the set of search engines — each resource was associated to one and only one vertical. The objective of the task was to select the most promising verticals to retrieve from. Once the most promising verticals have been selected, resource selection could be performed for each vertical, thus selecting the most promising search engines associated to the vertical.

The remainder of this paper is organized as follows. Section 2 briefly reviews the TWF·IRF weighting framework; Section 3 reports the research questions, the experimental methodology and the obtained results. Section 4 reports some final remarks.

[1] The identifier adopted in TREC for our research group is UPD.

2 A Recursive Weighting Scheme

According to the literature on Distributed IR [1, 2], the approach adopted in this work is to describe the informative resources at the diverse levels (document, search engines, verticals) in terms of document descriptors, e.g. terms. Therefore, a search engine is described by a set of document descriptors, specifically the distinct descriptors appearing in the documents stored in it; each vertical is described by the set of document descriptors used to describe the search engines associated to the vertical. The innovative contribution of our approach consists in the way of computing the weights of the descriptors.

The weight of a descriptor in a resource consists of two components: TWF and IRF. The Inverse Resource Frequency (IRF) is a generalization of the Inverse Document Frequency (IDF) for the higher resource levels. Generalizations of the IDF were proposed in [1] to rank collections (Inverse Collection Frequency, ICF) and in [2] to rank peers (Inverse Peer Frequency, IPF). The IRF extends this idea for an arbitrary resource level:

$$irf_t^{(z)} = \log N^{(z)}/n_t^{(z)} \tag{1}$$

where t denotes the term, $N^{(z)}$ is the number of resources at level z contained by the resource at level $z+1$ and $n_t^{(z)}$ is the number of those resources that are indexed by t. The instantiation of IRF at level 1 results in the IDF; ICF and IPF are instances of the IRF weight at level 2. In the FedWeb14 settings there are four resource levels: (1) document, (2) search engines, (3) verticals, and (4) set of verticals.

Unlike IRF, Term Weighted Frequency (TWF) is peculiar to this scheme. The weight of a descriptor t in a resource i at level z is

$$w_{i,t}^{(z)} = twf_{i,t}^{(z)} \cdot irf_t^{(z)}, \tag{2}$$

where

$$twf_{i,t}^{(z)} = \sum_{r_j \in R_i^z} twf_{j,t}^{(z-1)} \cdot irf_t^{(z-1)} \tag{3}$$

and R_i^z denotes the sets of resources in the ith resource at level z. For a given query q, resources at level z can be ranked according to $\sum_{t \in q} w_{i,t}^{(z)}$.

3 Experiments

3.1 Experimental Methodology and Research Questions

The experimental methodology consists of two tasks:

- **Resource selection**: Consider a set of search engines S, a set of queries $Q_{\mathcal{T}}$ and a set of sample documents obtained by query-based sampling performed on each of the search engines: The goal of this task is to return a ranked list of search engines for each query in $Q_{\mathcal{T}}$, where the search engines should be ranked according to their capability to satisfy the user's information need expressed by the query.

– **Vertical selection**: Consider a set of search engines S, a partition of the set of search engines in a set of verticals V, a set of queries $Q_{\mathcal{T}}$ and a set of sample documents obtained by query-based sampling performed on each of the search engines: The goal of this task is to return the most promising verticals for each query in $Q_{\mathcal{T}}$.

Our approach to performing the *vertical selection* task was to rank the verticals by TWF·IRF and select the top k verticals in the ranked list, for a given query. The motivation for this choice was to gain some insights into the vertical ranking capability of the adopted weighting framework. The cut-off of $k = 5$ was arbitrarily chosen. The selection of the cut-off can affect the effectiveness since the number of relevant verticals for a query can be less than k or greater than k; in the former case we could provide non-relevant verticals, in the latter case we could provide only a subset of all the relevant verticals.

The participation to FedWeb14 allowed us to investigate some specific research questions on TWF·IRF, specifically on the problems of *resource description* and *resource selection*.

With regard to the problem of *resource description*, the main research question was:

– **RQ0**: In the Federated Web Search settings there is no complete information on the collections indexed by the distinct search engines. *Is the TWF·IRF effective even if resource description is based on document obtained by query-based sampling?*

The comparison with the other FedWeb participants in TREC2013 and TREC2014 will provide us with some insights into RQ0.

Since TWF·IRF describes resources at the diverse levels by document descriptors, other research questions concerned with the pre-processing operations, such as stemming and stop-words removal:

– **RQ1**: *How does stemming affect the TWF·IRF effectiveness?*
– **RQ2**: *How does stop-words removal affect the TWF·IRF effectiveness?*

RQ1 was addressed by investigating diverse stemming algorithms, particularly Porter Stemmer, Krovetz Stemmer and comparing the effectiveness with the configuration without stemming. RQ2 was addressed by comparing the effectiveness of TWF·IRF with and without stop-words removal.

With regard to the problem of *resource selection*, the research questions involve the effect of the IRF component.

– **RQ3**: *Is the IRF component necessary for improving resource selection? Is the TWF component sufficient?*

In the experiments reported in this paper we did not compute IRF according to the Spark Jones formulation of IDF — that corresponds to Eq. 1; we instead used the formulation derived from the RSJ weight [7] when no relevance

information is available, specifically $\log(x)$ where

$$x = \frac{N^{(z)} - n_t^{(z)} + 0.5}{n_t^{(z)} + 0.5} \tag{4}$$

Since $\log(x) < 0$ when $n_t^{(z)} > N^{(z)}/2$, we considered the following variant $irf_t^{(z)} = \log(1 + x)$ in order to avoid negative values for the IRF. With regard to the IRF instantiations, the research question was:

– **RQ4**: *What is the effect of the IRF instantiation on the resource selection effectiveness?*

3.2 Test Collection and Effectiveness Measures

The experiments were carried out on the FedWeb14 test collection. This test collection consists of sampled search results from 149 web search engines crawled between April and May 2014. These 149 engines were a subset of the 157 search engines in the FedWeb 2013 test collection. Four thousand queries were adopted to gather samples from the diverse search engines; these samples were the basis for building descriptions for the informative resources at the various levels (search engines and verticals). The participants were provided with 75 test topics, but only 50 of them were actually used for the evaluation.

The effectiveness measures adopted in the resource selection task were nDCG@20 (official metric), nDCG@10, nP@1, nP@5. nDCG [6] was computed using the trec_eval variant where the discounting factor is $\log_2(i+1)$; for the nP@k metric the reader can refer to [3].

The effectiveness measures adopted in the vertical selection task were precision (P), recall (R) and F1-measure; the relevance for each vertical was obtained using the GMR+II approach described in [8].

3.3 Parsing and Indexing

The indexing module of our system relies on the Apache Lucene library and on an XML parser written in Java for extracting the document fields from the sample searches and the sample documents in the test collection. The sample documents in the FedWeb14 Test Collection were indexed by creating a distinct index for each of the 149 search engines. These indexes were *document-level* indexes. Each (Lucene) document in a document-level index was constituted of three fields: title, description, and the content of the document associated to the sample search result. For each field, the document-level index stored the positions and the frequency of the descriptors in each document and in the collection.

Starting from these indexes, a search engine-level index was built. The set of descriptors in this index is the union of all the distinct descriptors in the distinct document-level indexes associated to the search engines. As the document-level index, a list of posting is associated to each descriptor in the search engine-level

index. Each posting stores information on the identifier of the search engine, the number of documents in the search engine where the descriptor appears, and the TWF of the descriptor — see Section 3.4 for the computation of the TWF at search engine-level. In the specific Lucene-based implementation adopted, the TWF weight was stored in the payload that can be associated to each term; the weight value was approximated and stored as a float.[2] The search engine-level index was adopted for the resource selection task.

For the vertical selection task we built a search engine-level index for each of the 24 verticals. Starting from these indexes, a vertical level index was built. In a vertical level index the set of descriptors is the union of all the distinct descriptors in the diverse search engine-level indexes associated to the verticals. A list of posting is associated to each descriptor; each posting stores information on the identifier of the vertical, the number of search engines where the descriptor appears, and the TWF of the descriptor at vertical level — see Section 3.5 for the computation of the TWF at vertical level.

3.4 Resource Selection

The runs submitted to the FedWeb14 track exploited both the TWF and the IRF components of the weighting framework described in Section 2. This score was adopted to rank search engines in the resource selection task. The score of a search engine for a query q was computed as

$$\sum_{t \in q} twf_{i,t}^{(2)} \cdot irf_t^{(2)} \tag{5}$$

where $twf_{i,t}^{(2)} = \sum_{d_j \in D_i} twf_{j,t}^{(1)} \cdot irf_t^{(1)}$ and D_i denotes the sets of documents in the ith search engine, $twf_{j,t}^{(1)} = tf(t,j)$ is the term frequency of term t in the document d_j. The IRF at the document level was implemented as:

$$irf_t^{(1)} = \log\Big(\frac{N^{(1)} - n_t^{(1)} + 0.5}{n_t^{(1)} + 0.5}\Big) \tag{6}$$

where $N^{(1)}$ is the number of documents indexed by the search engine and $n_t^{(1)}$ is the number of those documents where the descriptor t appears.

Differently from last year, we exploited both the TWF and the IRF components because the experiments carried out with the FedWeb13 test collection suggested that IRF at search engine level can be beneficial in terms of resource ranking effectiveness [5]. The IRF at search engine level was computed as follows:

$$irf_t^{(2)} = \log\Big(\frac{N^{(2)} - n_t^{(2)} + 0.5}{n_t^{(2)} + 0.5}\Big) \tag{7}$$

where $N^{(2)}$ is the number of search engines and $n_t^{(2)}$ is the number of those search engines the index of which contains the descriptor t.

[2] Single-precision 32-bit IEEE 754 floating point

The ranked list of search engines was obtained by appending three ranked lists:

L_1: the list of search engines ranked by their TWF·IRF weight with regard to the query, and using the AND boolean constraint among the occurrence of the distinct terms in the query[3];
L_2: the list of search engines that did not belong to L_1 and ranked by their TWF·IRF weight with regard to the query by using the OR boolean constraint among the occurrence of the distinct terms in the query;
L_3: the list of search engines that did not belong to L_1 and L_2, ranked by their identifier — the identifier associated to the search engine in the test collection.

The final ranked list of search engines was obtained by appending L_2 to L_1, and then L_3 to the fusion of the first two lists.

We submitted seven runs for the resource selection task; the difference among the diverse runs is determined by the following variables:

- adoption of the IRF variant used at the document-level also for the search engine-level — use of $\log(1 + x)$ instead of $\log x$
- stemming algorithm adopted
- adoption of the stop-list

The label associated to the diverse runs is structured on the basis of the choice made for each variable. The first seven letters of each label (UPDFW14) are shared by all the runs since they refer to the participating group (UPD) and the track (FW14).

The eighth and the ninth letter denote the adopted IRF variant: `ti` refers to that reported in Equation 7, while `r1` refers to the $\log(1 + x)$ variant.

The tenth letter refers to the stemming algorithm adopted: (k) Krovetz stemmer, (p) Porter stemmer, and (n) no stemmer.

The eleventh letter denotes whether or not a stop-list was adopted: (s) Lemur stop-list, and (n) no stop-list.

The twelfth letter refers to the Boolean constraint adopted on the query term occurrence: (m) denotes the "cascade approach" described above that exploits AND, OR and then append the list of remaining search engines ordered by search engine identifier.

Results. Table 1 reports the obtained results for the resource selection runs; we omitted the first seven letters from the run labels – reported in the first column – since they are the same for all the runs. The most effective configurations involve the adoption of the stemmer and the stop-list. The Porter stemmer seems to provide slightly better results than Krovetz stemmer in terms of nDCG@20 — the values of the metric are basically the same but looking at the standard

[3] The Lucene query was a BooleanQuery constituted of PayloadTermQuery connected by MUST clause.

deviation the configuration with Porter stemmer results in less variability among the topics. The most effective runs in terms of nDCG@20 are those for which the stop-list was adopted.

With regard to the adoption of the IRF component for resource selection, the results obtained using the FedWeb13 test collection [5] suggested that it is beneficial in terms of nDCG@20. The same results was observed on the FedWeb14 test collection. Indeed, the nDCG@20 was 0.2985 in the most effective configuration – porter stemming and stop-words removal – when only the TWF component was adopted, and it was 0.3112 when both TWF and IRF were used.

The way IRF is instantiated affects the capability of the TWF·IRF at search-engine level as suggested by the comparison between the `ti-k-s-m` and the `r1-k-s-m` runs in terms of nDCG@20; the $\log(x)$ version outperformed the $\log(1 + x)$ version of the IRF.

Table 1. Comparison among the UPD resource selections runs.

run	nDCG@20	nDCG@10	nP@1	nP@5
ti-p-s-m	0.311 (+-0.143)	0.226 (+-0.156)	0.123 (+-0.219)	0.187 (+-0.163)
ti-k-s-m	0.310 (+-0.150)	0.223 (+-0.153)	0.126 (+-0.218)	0.188 (+-0.161)
ti-n-s-m	0.306 (+-0.152)	0.221 (+-0.155)	0.153 (+-0.255)	0.197 (+-0.184)
r1-k-s-m	0.292 (+-0.151)	0.209 (+-0.164)	0.148 (+-0.236)	0.180 (+-0.164)
ti-n-n-m	0.281 (+-0.155)	0.212 (+-0.146)	0.134 (+-0.242)	0.201 (+-0.179)
ti-p-n-m	0.280 (+-0.144)	0.212 (+-0.148)	0.115 (+-0.217)	0.191 (+-0.159)
ti-k-n-m	0.278 (+-0.152)	0.209 (+-0.146)	0.118 (+-0.216)	0.191 (+-0.157)

3.5 Vertical Selection

As mentioned in Section 3.1 the vertical selection task was investigated as a ranking task using a cut-off of $k = 5$ in the ranked list. The runs submitted to the vertical selection task exploited only the TWF component of the adopted weighted scheme. Ranking was performed using the same approach adopted for resource selection, i.e. appending the three lists L_1, L_2, and L_3; in this case, the informative resources in the result lists are not search engines but verticals.

For the vertical selection task we submitted six runs; these runs were based on different configurations of two variables: the way in which the IRF at search engine level was computed — (v0) $\log(x)$ or (v1) $\log(1 + x)$, and the stemming algorithm adopted.

We do not report the results based on the official runs since they were affected by a bug in the ranking procedure. Table 2 reports the results where Porter stemming and stop-words removal were adopted in the two configurations v0 and v1. The results suggest that the way IRF is computed at search engine level – that affects the TWF weight at vertical level – affects the effectiveness in terms of vertical selection: differently from what was observed for resource selection, v1 performed better than v0.

Table 2. Comparison among the UPD vertical selections runs.

run	P	R	F1
UPDFW14v0psm	0.1480	0.4337	0.2053
UPDFW14v1psm	0.1560	0.4737	0.2203

4 Final Remarks

This paper reported on the participation of the IMS Research Group of the University of Padua at the TREC2014 Federated Web Search Track. The participation allowed us to gain some insights into diverse research questions concerning with TWF·IRF, our recursive weighting scheme, when it is used in Federated Web Search setting:

- (RQ1) stemming has no significant effect on the TWF·IRF effectiveness in terms of search engine ranking;
- (RQ2) stop-words removal *improves* the TWF·IRF effectiveness in terms of search engine ranking;
- (RQ3) the IRF component *can improve* the TWF·IRF effectiveness in terms of search engine ranking;
- (RQ4) the way IRF is computed has different effects depending on whether it is used for search engine ranking or vertical representation.

With regard to the research RQ0, i.e. on the effectiveness of TWF·IRF when resource description is based on document obtained by query-based sampling, the comparison with the other participants in TREC2013 and TREC2014 does not provide a clear picture. This year the approach was not so effective as in TREC2013 test collection; since the number of queries used for sampling doubled – 4000 in 2014 versus 2000 in 2013 – the obtained results seems to suggest that TWF·IRF is more effective than other methods when less precise descriptions are available. In order to gain additional insights on the possible causes for the lack of effectiveness, future investigations will be focused on

- the effect of sampling strategy on resource selection effectiveness, e.g. by using distributed IR test collections where also the complete description is available, or the samples obtained by considering the diverse query sets (for sampling) in the FedWeb test collections;
- the use of diverse weighting scheme at document level, e.g. BM25 instead of the TF·IDF;
- the use of external evidence to obtain a more effective information need representation.

With regard to the vertical selection task, a possible direction for future investigations is the adoption of TWF·IRF as a feature for document descriptors in classification algorithms.

References

1. J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '95, pages 21–28, New York, NY, USA, 1995. ACM.
2. F. M. Cuenca-Acuna and T. Nguyen. Text-Based Content Search and Retrieval in Ad-hoc P2P Communities. In E. Gregori, L. Cherkasova, G. Cugola, F. Panzieri, and G. Picco, editors, *Web Engineering and Peer-to-Peer Computing*, volume 2376 of *Lecture Notes in Computer Science*, pages 220–234. Springer Berlin Heidelberg, 2002.
3. T. Demeester, D. Trieschnigg, D. Nguyen, and D. Hiemstra. Overview of the TREC 2013 Federated Web Search Track. In *Proceedings of the Twenty-Second Text REtrieval Conference (TREC 2013)*. National Institute of Standards and Technology (NIST), 2014.
4. E. Di Buccio, I. Masiero, and M. Melucci. University of Padua at TREC 2013: Federated Web Search Track. In *TREC*. National Institute of Standards and Technology (NIST), 2013.
5. E. Di Buccio, I. Masiero, and M. Melucci. Evaluation of a Recursive Weighting Scheme for Federated Web Search. In R. Basili, F. Crestani, and M. Pennacchiotti, editors, *IIR 2014*, volume 1127 of *CEUR Workshop Proceedings*, pages 1–10. CEUR-WS.org, 2014.
6. K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446, Oct. 2002.
7. S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.
8. K. Zhou, T. Demeester, D. Nguyen, D. Hiemstra, and D. Trieschnigg. Aligning vertical collection relevance with user intent. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1915–1918, New York, NY, USA, 2014. ACM.