

# Exploration of the Tradeoff between Effectiveness and Efficiency for Results Merging in Federated Search

## ABSTRACT

Federated search is the task of retrieving relevant documents from different information resources. One of the main research problems in federated search is to combine the results from different sources into a single ranked list. Recent work proposed a regression based method to download some documents from each ranked list of the different sources, calculated comparable scores for the documents and estimated mapping functions that transform source-specific scores into comparable scores. Experiments have shown that downloading more documents improves the accuracy of results merging. However downloading more documents increases the computation and communication costs.

This paper proposes a utility based optimization method that enables the system to automatically decide on the desired number of training documents to download according to the user's need for effectiveness and efficiency.

## Categories and Subject Descriptors

H3.3 [Information Search and Retrieval]

## General Terms

Performance, Experimentation, Theory.

## Keywords

Federated Search, Results Merging, Effectiveness and Efficiency Tradeoff.

## 1. INTRODUCTION

The proliferation of online searchable text information sources on local area networks and the Internet creates a problem of finding information that may be distributed among many information sources (federated search) [1,2]. One of the main research topics of federated search is to merge the search results returned from different information sources into a single ranked list. This task is called *results merging*.

Results merging is difficult because each source may use a different ranking algorithm, and may base its ranking on heterogeneous corpus statistics. Merging based upon unnormalized ("raw") document scores or ranks works well when the retrieval algorithms and corpora of different sources are very similar, but can be very inaccurate when they differ. Merging based upon weighted document scores or ranks [1] is more effective, but is heuristic and prone to unexpected failure. An alternative solution is to download the contents of all retrieved documents and then recalculate their scores by a centralized retrieval algorithm, which produces a consistent ranking but can be very time-consuming.

Prior research has proposed an effective regression based approach for the results-merging task [5, 6]. This method tries to obtain the contents of a small number of documents in the ranked lists of each information source either by downloading some retrieved documents from different sources or from the sampled documents by query-based sampling [1] (when they are available).

Then an effective centralized retrieval algorithm can be applied to calculate comparable document scores for these documents. This set of documents has both centralized comparable document scores and source-specific document scores. A source-specific logistic transformation model is estimated with the training documents for each information source to transform source-specific document scores to centralized comparable document scores. Furthermore, all documents can be sorted with respect to their estimated centralized documents scores into a final ranked list. Experiments have shown that when the number of documents downloaded as the training documents is increased, the effectiveness increases as well. However downloading a large number of documents for training data requires a high amount of online communication and computation costs.

This paper proposes an automated approach to decide on the desired number of documents to download for balanced effectiveness and efficiency. This method considers the effectiveness factor by the mean average precision of the merged results. With a set of training queries with relevance judgment, a logistic regression model is built to estimate the mean average precision values when different number of training documents can be downloaded. The efficiency factor is measured by the number of documents to download. Then for a test query, a utility based optimization method decides on the number of documents to download depending on a user-defined tradeoff parameter between effectiveness and efficiency.

## 2. METHOD

The method that we propose is composed of two parts. First, a logistic regression model is constructed to estimate the mean average precision values of merged results when different amount of training documents are downloaded for building regression models. The logistic model to estimate the mean average precision value is as follows:

$$MAP(y) = e^{a*y+b} / (1 + e^{a*y+b}) \quad (1)$$

where  $y$  is the number of downloaded training documents and  $a, b$  are the logistic model parameters. The logistic model is trained by the training queries and their  $\langle MAP(y)$  (mean average precision),  $y$  (number of downloaded training document)  $\rangle$  pairs.

Second, a utility based optimization problem can be formed by exploring the trade off between effectiveness and efficiency. The optimization problem is as follows:

$$U(y) = \beta * \alpha * MAP(y) - (1 - \alpha) * y \\ \arg \max_y U(y) \quad (2)$$

where  $U(y)$  is the utility value by downloading  $y$  documents for results merging.  $\beta$  is a constant that is set to be 250 for adjusting the range of the MAP value.  $\alpha$  is the user defined tradeoff parameter for effectiveness and efficiency. The user informs the system of his/her preference between effectiveness and efficiency;

and the system finds the desired number of training documents to download by computing the  $y$  value that maximizes the utility based optimization problem and so by deciding whether it's worth to download more training documents which increases effectiveness and sacrifices efficiency depending on  $\alpha$ .

### 3. EXPERIMENTS

In this section, we present the experimental results of the results merging task by comparing the proposed technique of choosing the number of documents to download for maximizing utility and a technique that always chooses a fixed number of documents to download. The experiments were conducted on two testbeds with eight multilingual information sources, namely the UNINE testbed (eight ranked lists from the UniNE system [4]) and the Hum testbed (eight ranked lists from Hummingbird system [3]). We used 20 training queries (i.e. queries 141-160) with relevance judgments to train our logistic regression model in Equation 1 and 40 test queries (i.e. queries 161-200) to test our system. The experiments only utilized downloaded documents as training data for building regression models. The results merging task is to merge eight ranked lists in different languages into a single multilingual ranked list. More detailed information about the testbeds can be found in [6].

**Table 1.** Utility values for the UniNE Testbed.

M	$\alpha$ Values				
	0.1	0.25	0.5	0.75	0.9
T_3	5.969	19.422	41.845	64.267	77.721
T_5	4.7618	19.404	<b>43.809</b>	68.213	82.856
T_10	0.6507	16.627	43.254	<b>69.881</b>	<b>85.857</b>
T_15	-3.6982	13.254	41.509	69.763	<b>86.716</b>
T_30	-17.288	1.7794	33.559	65.338	84.406
T_100	-80.215	-50.536	-1.0725	48.391	78.069
T_Opt	<b>6.6885</b>	<b>19.721</b>	<b>43.626</b>	<b>69.763</b>	<b>85.531</b>

**Table 2.** Utility values for the Hum Testbed.

M	$\alpha$ Values				
	0.1	0.25	0.5	0.75	0.9
T_3	3.0278	12.069	27.139	42.208	51.25
T_5	1.3577	10.894	26.789	42.683	52.22
T_10	-2.9638	7.5906	25.181	42.772	<b>53.326</b>
T_15	-7.4175	3.9562	22.912	41.869	<b>53.243</b>
T_30	-20.833	-7.0819	15.836	38.754	52.505
T_100	-83.789	-59.473	-18.946	21.581	45.897
T_Opt	<b>4.1248</b>	<b>13.312</b>	<b>28.624</b>	<b>42.956</b>	<b>52.72</b>

T\_X represents the merging method which always downloads top X documents for the training process. We have conducted the experiments for T\_X where X is in {3, 5, 10, 15, 30, 100} with different values of  $\alpha$  is in {0.1, 0.25, 0.5, 0.75, 0.9}. T\_Opt represents the merging method which downloads a desired number of top ranked documents for the training process.  $\alpha = 0.9$  corresponds to the case in which effectiveness is of higher importance, and  $\alpha = 0.1$  corresponds to the case where the main

focus is efficiency. The utility values for all the T\_X methods are directly calculated using human relevance, when the system strictly downloads a fixed number of documents for training. The utility values of the T\_Opt model are calculated using human relevance judgment, while system chooses a desirable number of training documents to download with respect to different  $\alpha$  values. From the results shown in Table 1 and Table2, it can be seen that no single T\_X method can obtain consistent good utility values for different  $\alpha$  values. However, the T\_Opt method can always achieve good utility values for different  $\alpha$  values by automatically choosing different desired number of documents to download. This explicitly demonstrates the power of the T\_Opt method for exploring the trade off between effectiveness and efficiency.

### 4. CONCLUSION

Results merging is an important task for federated search. Regression-based results merging method has been shown as an effective merging method. However, no research has been conducted to explore the trade off between the effectiveness of the regression based merging method and the efficiency of the merging method measured by the numbers of documents for downloading. This paper proposes an automated approach to decide a desired number of training documents to download for the results merging task by considering both effectiveness and efficiency. Experiment results have demonstrated the advantage of the proposed method to achieve good utility values in different configurations.

There are several possibilities to extend the research. The current method uses the same model for estimating mean average precision with respect to different number of documents to download for different queries. A more sophisticated query-specific estimation model may provide more accurate estimation for mean average precision with respect to different characteristics of different queries.

### 5. REFERENCES

- [1] Callan, J. 2000. "Distributed information retrieval" In W.B. Croft, editor, *Advances in information retrieval*. Kluwer Academic Publisher.
- [2] Yu C, Liu K. L., Meng W, Wu Z, Rish N. 2002. "A Methodology to Retrieve Text Documents from Multiple Databases". IEEE Transaction on Knowledge and Data Engineering.
- [3] HummingBird System <http://www.hummingbird.com/products/searchserver/>
- [4] Savoy, J. 2003. Report on CLEF-2003 Experiments. In C. Peters(Ed.), Results of the CLEF2003 cross-language evaluation forum
- [5] Si, L. and Callan, J. 2003. "A Semi-Supervised Learning Method to Merge Search Engine Results" In *ACM Transactions on Information Systems*, 24(4).
- [6] Si, L. and Callan, J. 2005. "CLEF2005: Multilingual Retrieval by Combining Multiple Multilingual Ranked Lists". In C. Peters(Ed.), Results of the CLEF2005 cross-language evaluation forum.