Effective Query Expansion for Federated Search

Milad Shokouhi
Microsoft Research
milads@microsoft.com

Leif Azzopardi
University of Glasgow
leif@dcs.gla.ac.uk

Paul Thomas
CSIRO
paul.thomas@csiro.au

ABSTRACT

While query expansion techniques have been shown to improve retrieval performance in a centralized setting, they have not been well studied in a federated setting. In this paper, we consider how query expansion may be adapted to federated environments and propose several new methods: where focused expansions are used in a selective fashion to produce specific queries for each source (or a set of sources). On a number of different testbeds, we show that focused query expansion can significantly outperform the previously proposed global expansion method, and—contrary to earlier work—show that query expansion can improve performance over standard federated retrieval.

These findings motivate further research examining the different methods for query expansion, and other forms of system and user interaction, in order to continue improving the performance of interactive federated search systems.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.4 [**Information Storage and Retrieval**]: Distributed Systems; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries—*hidden web*

General Terms

Design, Experimentation

Keywords

distributed information retrieval, query expansion

1. INTRODUCTION

Federated Information Retrieval (FIR) systems are required to provide effective retrieval performance, in distributed, and generally uncooperative, environments. Past research has largely focused on the necessary building blocks required for effective federated retrieval [3], such as *representation* [4, 20], *selection* [9, 21], and *merging* [22]. This has resulted in the development of FIR systems which deliver performance on par with centralized systems [27].

This success has opened up a significantly more challenging area of federated retrieval research: exploring interaction within the federated search process. Already, system and user interactions such as *pseudo-relevance* feedback [13], query expansion/reformulation [12, 26], implicit feedback [25], etc have been shown to provide significant increases to performance in centralized search systems. Now that the underlying machinery of FIR has matured, the next logical progression is to determine whether such performance enhancing techniques can also be employed to improve the effectiveness of FIR systems. The key challenge is determining how such techniques can be adapted to fit the distributed environment in order to realize potential increases. The focus of this paper is on the application and development of automatic query expansion (QE) techniques for FIR. Given the FIR environment, we propose a number of novel techniques for QE which treat constituent servers differently and dispatch focused, or server-specific, queries. This contrasts the only other proposed method [15] of query expansion for FIR which performs expansion in a globalized, "one query fits all", fashion. By creating query expansions for each local server, a focused query can be generated which better matches the content within the server, potentially avoiding topic drift or vocabulary mismatch. Our experiments on several testbeds, with a variety of training parameters and state-of-the-art FIR components, show that local and focused QE can outperform existing techniques and (contrary to earlier findings [15]) can improve FIR performance over a no-expansion baseline.

The remainder of this paper is as follows: the following section provides the necessary background describing the current state of the art in Federated Information Retrieval. In section 3, we describe our proposed methods for query expansion. We conduct a comprehensive and thorough experimental study in sections 4 and 5 to assess the impact on retrieval accuracy: we consider a number of different scenarios to determine whether the proposed QE techniques can improve retrieval effectiveness over several competitive baselines. Finally, we conclude with a discussion and summary of our main findings, before wrapping up with directions for future work.

2. BACKGROUND

Federated information retrieval. Federated information retrieval (FIR)—also called distributed information retrieval

SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.

(DIR)—provides a single interface to any number of independent *servers* (also referred to as collections, sources, or search engines). An FIR tool, also referred to as a *broker*, will characterize the available servers (representation); accept a user's query; decide which server to use (selection); translate the user's query to some appropriate syntax for each selected server; and collate the results from servers, typically into a single ranked list (merging).

Brokers rely on local representations of each server, which capture for example term occurrence; collection size; or subjects covered. Systems such as STARTS [7] and SDLIP [16] assume that servers are *cooperative* and make accurate representations available. However, in the general case, servers will be *uncooperative* and only provide a conventional query interface. In these cases, a broker must generate its own representation using a technique such as query-based sampling [4] or focused probing [8]. The former, the most commonly-used technique, submits a series of probe queries to a server and downloads a copy of documents from each result set: the union of these documents constitutes a *representation set*, and stands in for the server's contents.

Sampled documents have been used to estimate server characteristics—for example, the capture history estimator used in our experiments [19] uses the overlap between consecutive samples to estimate the size of a server's holdings—and so to inform selection (the CRCS algorithm we use assumes that the subject matter covered by a sample is representative of the whole collection [18]). In algorithms outlined in Section 3 we suggest also using sampled documents to inform query expansion.

Query Expansion. Query expansion techniques are designed to improve search effectiveness by providing a richer description of the information need. One or more additional terms are chosen and used to augment the user's original query: since user queries are typically very short, this can mitigate the vocabulary mismatch between the query and documents, and improve the retrieval effectiveness.

Expansion terms can be selected from external sources such as query logs [24], or dictionaries [6], or can be generated from a feedback process [5, 13, 17]. In the former category, the expansion terms are selected from pre-generated sources based on query reformulations, synonyms, etc. In the latter category however the terms are often generated on the fly based on feedback documents. These can be a hand-picked sample of relevant documents (*explicit feedback*) [17]. Alternatively, the top-ranked documents returned by the retrieval system for a query can be considered as *pseudo-relevant* for *implicit feedback*. The latter form of feedback is obviously far more scalable, but potentially noisier. The implicit form of relevance feedback is not only limited to pseudo-relevant documents but can be also collected from other sources such as clicks [11], or search trails [2].

Term weights in the expanded query can be calculated in a variety of ways. In the work of Ogilvie and Callan [15], which we follow, term weights are estimated using the relevance model (RM) [13]. This is based on language modeling and assumes the query Q and the top-ranked documents are sampled from the same model θ_R—this is the relevance model. θ_R is considered as a black box that determines the likelihood of each term given the query as follows:

$$P_C(t|\theta_Q) \approx \frac{\sum_{d \in F} P(t|\theta_d) P(Q|\theta_d) P(\theta_d)}{\sum_{t \in V} \sum_{d \in F} P(t|\theta_d) P(Q|\theta_d) P(\theta_d)} \quad (1)$$

where C denotes the document collection, V represents the vocabulary used and F represents the set of (pseudo-)relevant documents returned for query Q. We assume that the document prior $P(\theta_d)$ is uniform in our experiments. The most likely k terms according to the relevance model generate the expansion candidates.

While many methods for expansion exist, their application in FIR is largely unexplored.Ogilvie and Callan have proposed a global approach to query expansion for FIR [15]. This approach uses a central index of all the documents sampled from all the servers, which a user's query is matched against. The top-ranked passages/documents are returned, and the terms in these documents are used to expand the user's original query. The expanded query is then submitted to each of the selected servers. However, this approach represents only one possible way in which to perform QE in a distributed environment. We distinguish between the possibilities in the following section.

Ogilvie and Callan's study reveals that the global method does not perform significantly better than the unexpanded FIR baseline suggesting that QE is not particularly useful in FIR. However, this study was only performed on one testbed, with one set of topics, and so it is not possible to generalize these findings. The experimental study conducted here is performed on four testbeds with two sets of topics in order to thoroughly evaluate existing and proposed expansion methods.

3. QUERY EXPANSION IN FIR

The different approaches in which query expansion can be applied to FIR vary over four main factors: (i) data for expansion, (ii) level of aggregation, and (iii) query specialization. The fourth factor is the point of application, for re-selection of servers or for retrieval of documents. In this work, we only consider expansion for document retrieval. Below is a brief outline of the other three factors that could be varied:

1. **Data for Expansion**: Query expansion can be applied before a query is sent to any server, in which case expansion must be based only on sampled documents; alternatively query expansion can be a post-retrieval task, in which case it can be based on documents returned by servers. We refer to the first case as expansion based on *sampled documents*, and the second as expansion based on *source documents*. For efficiency, it is desirable to use sampled documents only.

2. **Level of Aggregation**: the documents (or passages) used to select expansion terms can be combined from all servers—which we refer to as *global expansion* —or can use only the documents from individual servers—*local expansion*. An alternative, *clustered expansion*, provides an intermediate level of aggregation between local and global.

3. **Query Specialization**: The expanded query could be *general*, such that the same query is sent to all servers; alternatively *focused* queries could be sent to

each server, such that different expanded queries are sent to different servers.

Given these different factors, we can say that the previous work on QE in FIR [15] investigated a global method; which sends general queries based on a global sample of documents. It should be noted that there is some dependence between these factors, and that more sophisticated methods may use both local and global information, or decide to send a mixture of general and focused queries. These conceptual factors are used to identify the main characteristics of the methods. In this section, we propose several novel alternatives, which use local and cluster aggregation of data, and focused queries. Thus the aim of this paper is to address the following questions:

- Can query expansion be applied effectively in FIR environments with small server summaries?
- Does *local* query expansion improve retrieval performance over *global* query expansion?
- Does *clustered* query expansion improve retrieval performance over *global* query expansion?
- Are *focused* queries better than *general* queries?

The benefits of using the local information obtained from a source is that a focused query can be generated which is tailored to the content in the source. The idea is that this will avoid vocabulary mismatch problems and topic drift. However, the disadvantage is that there is less data to use in order to select query terms for expansion. An operational question, then, is, what is the trade-off between the level of aggregation and effectiveness?

Below we present each of the proposed methods, along with the previously proposed global method. A summary of the methods is shown in Table 1.

Global (baseline). Ogilvie and Callan [15] indexed the sampled documents from all servers together on the broker. The top-ranked results returned for each query from this index are then used to expand the query. The selected servers receive the expanded query for document retrieval. Note, that all the selected servers receive exactly the same query. We refer to this method as *global* expansion, and use it as the competitive baseline. The weight of the expansion terms using the global model is estimated using:

$$w(t, Q, C) = P_G(t, Q) \tag{2}$$

where G represents the set of sampled documents from all servers, and $P_G(t, Q)$ denotes the probability of relevance computed by the relevance model as shown in Equation (1). In this approach, the same query is sent to each of the selected servers (C)–i.e. there is no query specialization—and thus the method is global and general. Specialization could be applied to the global method, but we do not propose such an extension here.

Local. In contrast to the Global method, our first expansion strategy performs *server-specific* query expansion. An expanded query is formulated for each server using the documents sampled from that server. The idea here is to expand the queries with representative terms that are specific to each server. The drawback may be that the samples contain fewer documents from which to find any reasonably good pseudo-relevant documents, compared to using the global information. Since this may affect the quality of the query expansion, in our experiments we investigate how the size of the samples affects retrieval performance. The *Local* query expansion method can be formalized as follows. For a server C, with a set of sampled documents S_C, the expansion weights are estimated using:

$$w(t, Q, C) = P_{S_C}(t, Q) \tag{3}$$

Note that the expansion terms and weights vary for each server producing a focused query.

Fuse: local but general QE. Since focused queries may suffer from noise, when the sample of documents is small, we consider employing a local but general approach: where each collection provides a set of possible query expansion terms, then from these suggestions a subset is selected to form the expanded query. This single expanded query is then issued to each of the selected servers. We employ a simple voting scheme to select the best expansion terms, which uses CombMNZ [14] to pick the best k expansion candidates. We refer to this approach as the *Fuse* QE method. The terms are sorted according to their CombMNZ scores (calculated as below), and the top k are selected for query expansion.

$$w(t, Q, C) = \eta_t \times \sum_C P_{S_C}(t, Q) \tag{4}$$

Here η_t, is the number of servers that recommend term t for expansion, and is used to boost the weights of terms that are suggested by more servers. S_C denotes the set of sampled documents from server C.

Cluster. The *Cluster* method is motivated by the two key points discussed for earlier methods; (a) increasing the size of the corpus used for query expansion improves the quality of feedback documents, and (b) using the sample documents from all collections for query expansion, may generate expansion terms that are not suitable for selected servers. In fact, it is possible that the expansion terms do not exist in a selected server. The two contradicting points above motivate a compromise between the local and global approaches: where the information used to perform expansion is based on clusters of sampled documents. Using clusters should increase the amount of data which is used to perform query expansion, while still trying to ensure that the terms selected in expansion are related to the collections.

The Cluster method can be summarized in three steps: (1) cluster the sample of documents into n buckets (here, we employ *kmeans* [10]), where n should be small enough to ensure the quality of feedback terms, and large enough to generate different expansion terms from each cluster, (2) assign servers to clusters according to their number of sampled documents in each cluster. A server C is assigned to a cluster ξ, if cluster ξ has more sampled documents from C than any other cluster; and (3) for each selected server C, expand the query according to the corresponding cluster for C.

$$w(t, Q, c) = P_\xi(t, Q) \tag{5}$$

Here ξ refers to the cluster that server c is assigned to (query

Table 1: Query expansion techniques for FIR considered in this paper. All methods use sampled documents.

	Level of aggregation		
	Local	Clustered	General
Focused queries	Local	Cluster	n/a
General queries	Fuse	—	Global [15]

independent). The expansion terms generated by this model are *cluster-specific*. That is, all the servers in the same cluster share the same expansion candidates.

4. EXPERIMENTAL METHODOLOGY

The aim of this paper is to evaluate the different strategies for query expansion in the context of FIR. In this section, we detail the experiments undertaken in order to perform a comprehensive and thorough evaluation of the competing strategies. These laboratory based experiments are designed to reflect the real world application of such technology and provide the best indication of performance in a simulated environment. The FIR system used in these experiments is described, along with the configuration parameters for query expansion and respective baselines. Then, we describe the set of testbeds used to perform the evaluation.

FIR System Setup. The FIR setup that was employed in our experiments is comprised of the state of the art in representation, selection and merging algorithms. To represent each server, query-based sampling [4] was used to generate the server representation sets (i.e. the set of sampled documents used to represent the collection). Each sampling query is selected with uniform probability for the downloaded documents. The top four documents returned by sampling queries are added to the server representation set. Three representations were used during the course of these experiments, where we considered the impact of sample size on retrieval effectiveness examining representations with 300, and 2900 documents.

CRCS with an exponential decay function [18] was employed for collection selection, whilst for results merging SSL [22] was employed. CRCS uses server size statistics to rank servers. Such information may not be avaible in practice. We used the *capture-history* method [19], using 140 probe queries, to estimate the size of servers. CRCS, SSL, and capture-history are all among state of the art techniques.

Avrahami et al. [1] showed that in real-world applications of federated search, selecting three to five servers usually produces the best trade-off between search effectiveness and efficiency. Therefore, in our experiments we report results for server selection cutoff value of three. Note that we found similar trends when a cutoff value of five was used; they are excluded here for brevity.

All servers use the KL divergence retrieval model, while query expansion was performed using the relevance model. We use Indri's[1] implementation of Kullback-Leibler divergence as our document ranking method for all the experiments in this paper. The relevance score of a document d for query Q is computed according to the divergence between the language models of query θ_Q, and document θ_d:

$$S(Q,d) = -KL(\theta_Q||\theta_d) = -\sum_{t \in V} P(t|\theta_Q) \log \frac{P(t|\theta_Q)}{P(t|\theta_d)} \quad (6)$$

where t denotes a term in the corpus vocabulary V. Dirichlet smoothing is used to avoid the zero probability. To generate the expansion candidates and their weights we use the relevance model [13], with the default parameters.

Expansion Parameters. On the training data, a sweep of the parameter space over the following parameters was employed, where the best settings were used on the testing test. Given the relevance model [13], the number of feedback documents was set according to $\alpha = \{1, 10, 50\}$, the number of expansion terms $\beta = \{1, 10, 50\}$, and the combination weight $\gamma = \{0.5, 0.7, 0.9\}$ (i.e the weight specifying the importance of original query words). For the cluster method we empirically set the number of clusters to $n = 4$ and leave further investigation of the best value for future work.

Baselines. In our experiments, the following baselines will be used in order to compare and contrast the differences between methods: each gold standard baseline (or oracle) assumes complete information under a centralized setting. The first oracle baseline (BONE) is without query expansion and the second oracle baseline (BOQE) is with query expansion. Our third baseline is obtained by performing federated retrieval without query expansion (BSNE). These baselines provide the reference points, while we also include Ogilvie and Callan's global model as a fourth baseline. This baseline is the only QE method currently proposed for FIR and thus represents the state of the art.

Testbeds. The standard testbeds developed by Callan et al. [4, 21, 22, 23] were used in this study. These testbeds provide four different distributed environments to thoroughly evaluate the performance of FIR systems.

- **trec123 (Trec123-100col)**: the documents are organized by source and publication date, and the testbed contains 100 servers, where the size of the servers is relatively uniform.
- **relevant (Trec123-AP-WSJ-60col)**: the same as trec123, except the AP and WSJ documents are all placed into two separate servers, and the remaining 60 databases are kept as is. In this case, most of the relevant documents are within the AP and WSJ servers.
- **nonrelevant (Trec123-FR-DOE-81col)**: the same as the trec123-by-source, except the two servers that contain the least relevant material are combined into separate servers (FR and DOE). This means the testbed contains two large servers that mainly contain non relevant information.

[1] `www.lemurproject.org`

Table 2: The effectiveness of query expansion methods on different testbeds for TREC topics 101–150. Expansion parameters are tuned by training on TREC topics 51–100. CRCS is used to select three servers per query.

	Method	P@5	P@10	MRR
relevant	BSNE	0.3080	0.2780	0.4062
	Local	0.2960	0.2780	0.3830
	Fuse	0.3040	0.2800	0.4001
	Cluster	0.3040	0.2780	0.3973
	Global	0.2960	0.2800	0.3954
nonrel.	BSNE	0.3000	0.2440	0.4830
	Local	0.3040	0.2480	0.4836
	Fuse	*0.2880*†	0.2400	0.4763
	Cluster	0.3120	**0.2700**†	0.4887
	Global	0.2840	0.2600†	0.4882
trec123	BSNE	0.3200	0.2780	0.4995
	Local	0.3080	0.2800	0.4950
	Fuse	0.3040	0.2720	0.4986
	Cluster	0.3240	0.2820	0.5106
	Global	0.3240	0.2780	0.5074
represent.	BSNE	0.3600	0.3160	0.5548
	Local	0.3520	0.3180	0.5665
	Fuse	0.3520	0.3180	0.5843
	Cluster	0.3600	0.3100	0.5631
	Global	0.3520	0.3320	0.5545
	BONE	0.4200	0.4160	0.5484
	BOQE	0.4560	0.4500	0.5852

- **representative (Trec123-2ldb-60col)**: The servers in the trec123 are sorted by their names. Then, starting from the first server, every fifth server is collapsed into a large collection. The process is repeated by starting from the second server. In total, the testbed includes 62 servers, with 2 servers being significantly larger than the others.

Evaluation. For these testbeds, we used TREC topics 51–100 and 101–150 to evaluate the performance of each approach. A training-testing methodology was employed, where each topic set was used for training the expansion parameters, while the other was used for testing, and vice versa. The performances reported are those obtained from the test sets. During training the parameters are tuned for maximizing the P@5 value when three servers are selected (CO=3). Optimizing for P@10 and other server selection cutoff values showed very similar results.

We use Student's t-test to measure the significance of differences between the methods, particularly compared to the no expansion baseline (BSNE). The notations † and ‡ respectively indicate significant differences at $p < 0.05$ and $p < 0.01$. Italicized text represents performance significantly worse than the BSNE baseline. We focus mainly on P@5, as this is the metric we use to tune the expansion parameters.

5. RESULTS

Retrieval Performance. Table 2 shows the performance of different expansion methods compared to the baselines for TREC topics 101–150. The first notable observation is that the previously proposed global method is outperformed by the baseline without expansion (BSNE) on P@5 for three of four testbeds (relevant, representative, nonrelevant). This is consistent with the findings reported by Ogilvie and Callan [15]. Similar observations can be made for the Local and Fuse models. Local, Fuse and Global, do slightly better on P@10 overall. The cluster method does slightly better than BSNE on the trec123, slightly worse on the relevant testbed and equally good on the representative testbed. On the non-relevant testbed, Fuse performs significantly worse than the baseline and other expansion methods, and the advantage of Local over the BSNE baseline is negligible. The global method has a lower P@5 than BSNE, but produces significantly better results on P@10. The cluster method again produces the best result by outperforming other QE methods across all testbeds (given the trained parameters from TREC Topics 51–100). The differences between the Cluster method and all the other methods are statistically significant for P@10 ($p < 0.05$).

Table 4 provides an overall comparison between methods across all testbeds for TREC topics 101–150 (trained on Topics 51–100). The Local and Fuse models perform slightly worse than BSNE. The P@5 loss is statistically significant for the fuse model, which we expect is due to differences in quality between servers: servers which have few on-topic documents or which are otherwise poor candidates still contribute votes to select expansion terms. This might be mitigated by using a selection algorithm to rank servers ahead of expansion, and by disregarding those which seem poor or off-topic. The global method produces lower P@5 values compared to BSNE, while significantly outperforms it in terms of P@10. The cluster model shows the best performance among all expansion models, and outperforms all the alternatives significantly ($p < 0.05$) for P@5.

We also repeated the experiments, by tuning the expansion parameters on TREC topics 101–150, and testing them on TREC topics 51–100. The results an be found in Table 3, under the "300 documents" column (also see Table 4 for an overall summary). Except for one case in the representative testbed where Local performed significantly poorer than BSNE, no other statistically significant difference was detected among methods. While all query expansion methods produce slightly better results than BSNE overall, none of the differences were statistically significant. In summary, the cluster and global expansion method tend to outperform the other methods.

Though, it should be noted that none of the tested methods produced operationally significant improvements over the no expansion baseline. However, we argue that our thorough analysis on four testbeds provides sufficient evidence that query expansion can be applied effectively in federated search, even with small summaries.

Impact of Sample Size. In earlier work, larger document samples have shown to improve FIR performance, and we therefore considered the effect of larger samples on our expansion methods. Work by Ogilive and Callan [15] used samples of varying size, from 300 to 2900 documents per server, and we repeat this analysis here.

Figure 1 depicts the changes to retrieval performance using different sample sizes: Table 3 summarizes results when we increased our sample size to 2900 documents, using the

Table 3: The impact of sample size on the effectiveness of query expansion methods for TREC topics 51-100. The expansion parameters are tuned using the TREC topics 101–150 on 300 document summary sets (best case for 300 document summaries). CRCS is used to select three servers per query.

	Method	P@5	P@10	MRR	P@5	P@10	MRR
		Sample size = 300 documents			**Sample size = 2900 documents**		
relevant	BSNE	0.3040	0.2720	0.4687	0.3040	0.2720	0.4687
	Local	0.3120	0.2780	0.4499	0.3160	0.2780	0.4756
	Fuse	0.3160	0.2700	0.4498	0.3240	0.2780	0.4702
	Cluster	0.3040	0.2700	0.4556	0.3000	0.2780	0.4575
	Global	0.2960	0.2740	0.4233	0.3120	0.2800	0.4447
nonrel.	BSNE	0.4760	0.4180	0.6238	0.4760	0.4180	0.6238
	Local	0.4480	0.3940	0.5810	0.4680	0.4180†	0.6153
	Fuse	0.4440	0.3840	0.6186	0.4840‡	0.4240‡	0.6440
	Cluster	0.4760	0.4180	0.6346	0.4720	0.4160	0.6344
	Global	0.4880	0.4280	0.6508	0.4680	0.4220	0.6341
trec123	BSNE	0.4280	0.3740	0.5727	0.4280	0.3740	0.5727
	Local	0.3960	0.3500	0.4996	0.4000	0.3520	0.5466
	Fuse	0.4200	0.3540	0.5842	0.4040	0.3640	0.5406
	Cluster	0.4160	0.3860	0.5449	0.4440†	0.3760	0.5814
	Global	0.4120	0.3800	0.5119	0.4640†	0.3840	0.5834†
represent.	BSNE	0.3840	0.3600	0.5671	0.3840	0.3600	0.5671
	Local	*0.3560*	0.3300	0.4965	0.3840	0.3500	0.5108
	Fuse	0.3880	0.3680	0.5292	0.4160†	0.3660	0.5493
	Cluster	0.3840	0.3580	0.5699	0.4000	0.3580	0.5699
	Global	0.4080	0.3580	0.5331	0.3840	0.3660	0.5299
	BONE	0.4800	0.4400	0.6119	0.4800	0.4400	0.6119
	BOQE	0.5240	0.4960	0.6394	0.5240	0.4960	0.6394

same training/testing split as earlier. To make the results comparable across different experiments, we use the original 300 samples for server selection and result merging. Therefore, any changes in search effectiveness is solely due to the quality of query expansion candidates. It can be seen from the table that increasing the sample size does not always improve. Here, † and ‡ represent statistically significant differences between experiments with small and large samples. Contrary to earlier work [15], we find this is particularly the case for the global method which performs poorly on two of the testbeds despite more data to estimate query expansions from. However, the performance of the local and cluster methods generally improved with larger samples. This is intuitive, because larger samples provide richer sources of text for pseudo-relevance feedback and are more likely to be representative of servers' holdings. For the local methods this is more important because this is the only information used for expansion.

Impact of Selection. A broker has a choice of algorithms for selection. We have so far been using CRCS for selection as it has been shown to be one of the best-performing algorithms [18]; a further set of experiments considered the impact of using CORI [3], another popular algorithm for selection. By doing so, different servers will be selected, and it will be interesting to see whether this will improve or degrade the query expansion methods.

Figure 2 illustrates P@5 for each selection algorithm. What is immediately striking is that using the CORI selection algorithm results in a significant loss in performance: this was the case across almost any combination of param-

Table 4: The average performance of query expansion methods across different testbeds for TREC topics 101–150 and TREC topics 51-100.

Method	P@5	P@10	MRR
TREC Topics 101-150			
BSNE	0.3220	0.2790	0.4858
Local	*0.2810*	0.2810	0.4820
Fuse	*0.3100*†	0.2775	0.4898
Cluster	**0.3250**†	0.2845	0.4899
Global	0.3140	0.2865†	0.4863
TREC Topics 51-100			
BSNE	0.3980	0.3560	0.5580
Local	0.3970	0.3505	0.5418
Fuse	0.4050	0.3530	0.5471
Cluster	0.4030	0.3550	0.5637
Global	0.4100	0.3605	0.5471

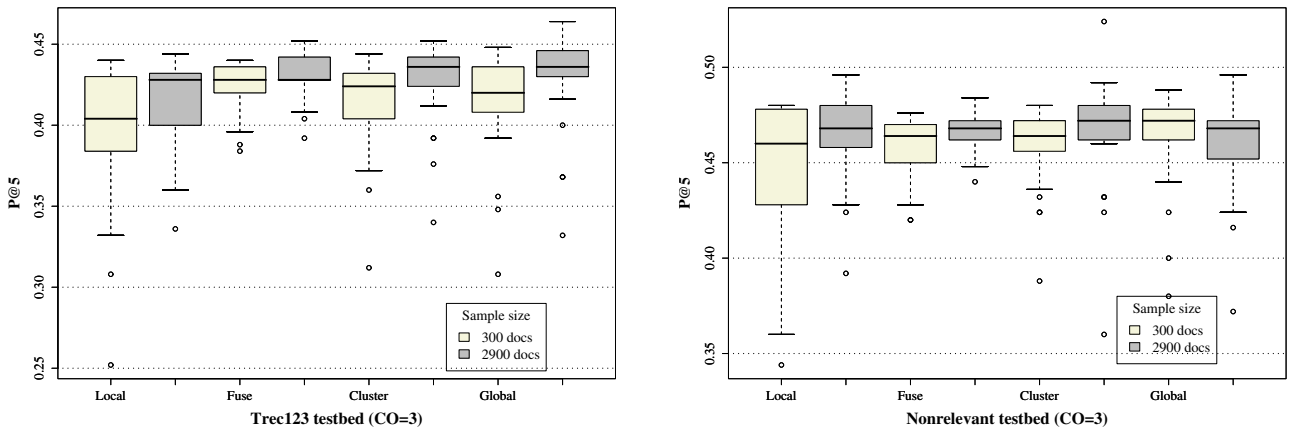

Figure 1: The impact of increasing sample size on P@5 values for query expansion methods on the trec123 (left) and non-relevant (right) testbeds. The boxplots illustrate the range of P@5 values that each expansion model produces during the training process on TREC topics 51–100. The solid black line denotes the median, and whiskers show the extent of the data range. Other testbeds showed similar trends.

eters, testbeds, and expansion methods.

However, using CORI the local query expansion methods did perform as well as, if not better than the cluster and global methods. For example, see Figure 2 for the results on the nonrelevant testbed.

6. CONCLUSIONS AND FUTURE WORK

Although query expansion techniques have been well-studied in the case of centralized IR, they have been largely ignored in federated IR research. As shown in Table 1, we have considered several means by which a FIR system could make use of query expansion: choosing expansion terms based on each collection separately (local expansion) and sending individual expanded queries to each collection (focused querying) using sampled documents.

We performed a comprehensive study on query expansion in FIR, testing the four algorithms (three novel) across four testbeds and employing a train-test methodology with two TREC topic sets. The results suggest that QE techniques can significantly improve FIR performance over the baseline, even with relatively small document samples from which to draw additional terms, although some level of aggregation is useful. This is an important finding as previous work [15] suggested that no improvement could be obtained over a baseline without query expansion (using small server summaries). However, this work provides evidence to suggest that query expansion can lead to improvements in performance in the FIR environment.

Generally, the cluster method performed consistently well across the different testbeds, which significantly outperformed the baseline without expansion, and the global method on TREC topics 101–150. On TREC topics 51–100 there was no significant difference between the global and cluster methods. The use of some local evidence (through clustering) shows that different aggregations can generate better expansions. While the local method was sensitive to the amount of data available to expansion, the focused queries performed better than the fuse method, which combined the local knowledge to form general queries. Despite the poor results obtained by these methods, the local method could be more useful when relevance feedback is provided, as this would remove the problems with using a small sample of documents, but provide specific and focused queries for each server. This remains a direction for future work. On the other hand, the Fuse method suffers because all local servers were treated equally in the process of forming a general query and this lead to noisy, and poorly performing expansions. By restricting the set of servers that form the general query to the best servers, improvements could also be achieved.

The taxonomy of factors for applying QE in FIR suggests further scope for experimentation: for example, with methods which use returned (not sampled) documents as sources of terms. Further investigation of clustered expansion, possibly in conjunction with ideas from the fuse or local methods, may also be worthwhile. More generally, other query modification and interaction techniques from conventional retrieval could also be investigated in the context of FIR environments, where the interaction can be used to drive and develop more effective methods for interactive FIR.

Acknowledgments

We would like to thank Dr. Mark Baillie for his input and assistance in developing this work.

7. REFERENCES

[1] T. Avrahami, L. Yau, L. Si, and J. Callan. The FedLemur project: Federated search in the real world. *JASIST*, 57(3):347–358, 2006.

[2] M. Bilenko and R. White. Mining the search trails of surfing crowds: identifying relevant websites from user activity. In *Proceedings of the Int.l Conf. on World Wide Web*, pages 51–60, Beijing, China, 2008.

[3] J. Callan. *Advances in Information Retrieval*, chapter Distributed Information Retrieval, pages 127–150. Kluwer

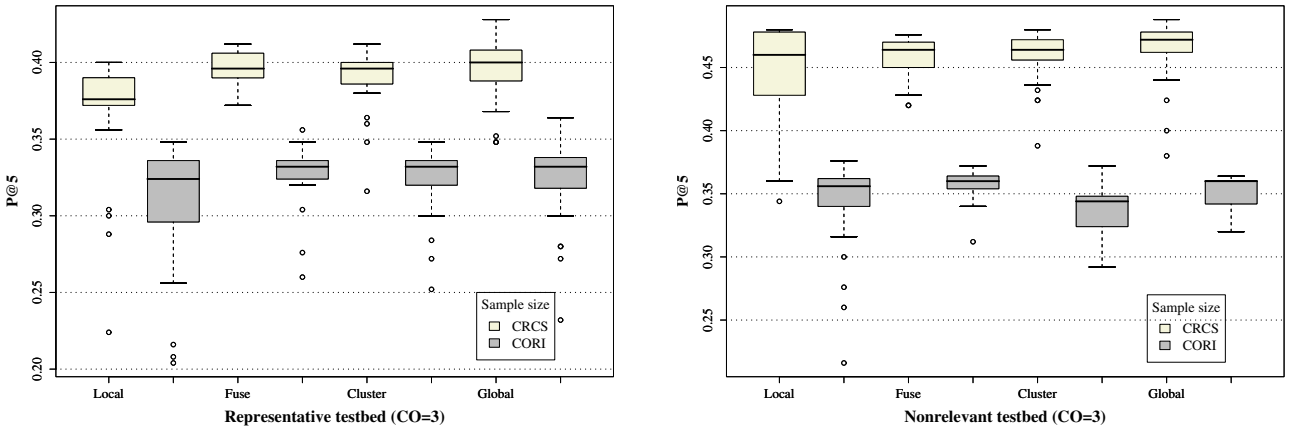

Figure 2: The impact of collection selection on P@5 values for different query expansion methods on the representative (left) and nonrelevant (right) testbeds. The broker selects three servers per query (CO = 3). The boxplots illustrate the range of P@5 values that each expansion model produces during the training process on TREC topics 51–100. The solid black line denotes the median, and whiskers show the extent of the data range. Other testbeds showed similar trends.

Academic Publishers, 2000.

[4] J. Callan and M. Connell. Query-based sampling of text databases. *ACM Transactions on Information Systems*, 19(2):97–130, 2001.

[5] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the ACM SIGIR Conference*, pages 243–250, Singapore, 2008.

[6] K. Collins-Thompson and J. Callan. Query expansion using random walk models. In *Proceedings of the ACM CIKM Conference*, pages 704–711, Bremen, Germany, 2005.

[7] L. Gravano, K. Chang, H. García-Molina, C. Lagoze, and A. Paepcke. STARTS: Stanford protocol proposal for internet retrieval and search. In *Proceedings of the ACM SIGMOD Conference*, pages 207–218, Tucson, Arizona, 1997.

[8] L. Gravano and P. G. Ipeirotis. QProber: A system for automatic classification of hidden-web databases. *ACM Transactions on Information Systems*, 21(1):1–41, 2003.

[9] D. Hawking and P. Thomas. Server selection methods in hybrid portal search. In *Proceedings of the ACM SIGIR Conference*, pages 75–82, Salvador, Brazil, 2005.

[10] A. Jain and R. Dubes. *Algorithms for clustering data*. Prentice-Hall, Upper Saddle River, NJ, 1988.

[11] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the ACM SIGIR Conference*, pages 154–161, Salvador, Brazil, 2005.

[12] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *Proc. of the Int. Conf. on World Wide Web*, pages 387–396, Scotland, 2006.

[13] V. Lavrenko and B. Croft. Relevance based language models. In *Proceedings of the ACM SIGIR Conference*, pages 120–127, New Orleans, LA, 2001.

[14] M. Montague and J. Aslam. Relevance score normalization for metasearch. In *Proceedings of the ACM CIKM Conference*, pages 427–433, Atlanta, GA, 2001.

[15] P. Ogilvie and J. Callan. The effectiveness of query expansion for distributed information retrieval. In *Proceedings of the ACM CIKM Conference*, pages 183–190, Atlanta, GA, 2001.

[16] A. Paepcke, R. Brandriff, G. Janee, R. Larson, B. Ludaescher, S. Melnik, and S. Raghavan. Search middleware and the simple digital library interoperability protocol. *D-Lib magazine*, 6(3), 2000.

[17] J. Rocchio. The SMART retrieval system: Experiments in automatic document processing. In *Relevance feedback in information retrieval*, pages 313–323, 1971.

[18] M. Shokouhi. Central-rank-based collection selection in uncooperative distributed information retrieval. In *Proceedings of the ECIR Conference*, pages 160–172, Rome, Italy, 2007.

[19] M. Shokouhi, J. Zobel, F. Scholer, and S. Tahaghoghi. Capturing collection size for distributed non-cooperative retrieval. In *Proceedings of the ACM SIGIR Conference*, pages 316 – 323, Seattle, WA, 2006.

[20] M. Shokouhi, J. Zobel, S. Tahaghoghi, and F. Scholer. Using query logs to establish vocabularies in distributed information retrieval. *IPM*, 43(1), 2007.

[21] L. Si and J. Callan. Relevant document distribution estimation method for resource selection. In *Proceedings of the ACM SIGIR Conference*, pages 298–305, Toronto, Canada, 2003.

[22] L. Si and J. Callan. A semisupervised learning method to merge search engine results. *ACM TOIS*, 21(4):457–491, 2003.

[23] L. Si and J. Callan. Unified utility maximization framework for result selection. In *Proceedings of the ACM CIKM Conference*, pages 32–41, Washington, DC, 2004.

[24] J. Wen, J. Nie, and H. Zhang. Query clustering using user logs. *ACM TOIS*, 20(1):59–81, 2002.

[25] R. W. White, I. Ruthven, J. M. Jose, and C. J. V. Rijsbergen. Evaluating implicit feedback models using searcher simulations. *ACM TOIS*, 23(3):325–361, 2005.

[26] J. Xu and B. Croft. Query expansion using local and global document analysis. In *Proceedings of the ACM SIGIR Conference*, pages 4–11, Zurich, Switzerland, 1996.

[27] J. Xu and W. B. Croft. Cluster-based language models for distributed retrieval. In *Proceedings of the ACM SIGIR Conference*, pages 254–261, Berkeley, CA, 1999.