

A Joint Probabilistic Classification Model for Resource Selection

Dzung Hong^{*}, Luo Si
Department of Computer Science
Purdue University
250 N. University Street
West Lafayette, IN 47907, USA
{dthong, lsi}@cs.purdue.edu

Paul Bracke, Michael Witt
Purdue University Libraries
Purdue University
504 West State Street
West Lafayette, IN 47907, USA
{pbracke, mwitt}@purdue.edu

Tim Juchcinski
Department of Computer Science
Purdue University
250 N. University Street
West Lafayette, IN 47907, USA
tjuchcin@purdue.edu

ABSTRACT

Resource selection is an important task in Federated Search to select a small number of most relevant information sources. Current resource selection algorithms such as GLOSS, CORI, ReDDE, Geometric Average and the recent classification-based method focus on the evidence of individual information sources to determine the relevance of available sources. Current algorithms do not model the important relationship information among individual sources. For example, an information source tends to be relevant to a user query if it is similar to another source with high probability of being relevant. This paper proposes a joint probabilistic classification model for resource selection. The model estimates the probability of relevance of information sources in a joint manner by considering both the evidence of individual sources and their relationship. An extensive set of experiments have been conducted on several datasets to demonstrate the advantage of the proposed model.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms

Algorithms, Design, Performance

Keywords

Federated Search, Resource Selection, Joint Classification

1. INTRODUCTION

Federated text search provides a unified search interface for multiple search engines of distributed text information sources. There are three major research problems in federated search as resource representation, resource selection

and results merging. This paper focuses on resource selection, which selects a small number of most relevant information sources to search for any particular user query.

Resource selection for federated search has been a popular research topic in the last two decades. Many methods treat each information source as a single big document and rank available sources either by using statistics from the sample documents (CORI [6]), or by building a language model for each source (Xu and Croft [27], Si and Callan [22]). Other methods such as GLOSS [12], Geometric Average [17], ReDDE [20], CRCS [18] and SUSHI [24] look further inside an information source by estimating the relevance of each document and calculate the source's score as an aggregate function of the documents that the source contains. More recent methods such as the classification-based method [1] and the work in [2] treat resource selection as a classification problem and build probabilistic models by combining multiple types of evidence of individual sources.

Existing resource selection methods judge an information source by its own characteristics, but miss an important piece of evidence, which is the relationship between available sources. In practice, we notice that relationship can be meaningful and indicative. An information source that is "similar" to another highly relevant source has a better chance of being relevant. The evidence of source relationship can be very valuable for real world federated search solutions. In particular, the resource representation (e.g., sample documents) of each information source is often limited and prevents resource selection algorithms from identifying relevant sources, while the relationship between sources can help to alleviate the problem by providing more evidence from similar sources. Our study of a real world federated search application with digital libraries also suggests that it is difficult to obtain thorough resource representation from many sources (i.e., digital libraries). For example, some sources may only provide the abstracts of its documents instead of the full texts.

This paper proposes a novel probabilistic discriminative model for resource selection that explicitly models the relationship between information sources. In particular, the new research combines both the evidence of individual sources and the relationship evidence between sources into a single probabilistic model for estimating the joint probability of relevance of a set of sources. Different similarity metrics have been studied to explore the relationship between information sources. An extensive set of experiments have been conducted on two TREC testbeds for federated search

^{*}Vietnam Education Foundation Fellow

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGIR '10, July 19–23, 2010, Geneva, Switzerland.

Copyright 2010 ACM 978-1-60558-896-4/10/07 ...\$10.00.

research and one real world application for searching digital libraries. The experiment results demonstrate the effectiveness and robustness of the proposed resource selection algorithm with the joint classification model.

The rest of this paper is organized as follows: the next section discusses the related research work. Section 3 presents the classification model for resource selection. Section 4 proposes the joint classification model. Section 5 discusses experimental methodology. Section 6 presents experimental results and related discussions. Section 7 concludes and points out some future research work.

2. RELATED WORK

There has been considerable research on all of the three subtasks of federated search as resource representation, resource selection and results merging. Since this paper focuses on the resource selection task, we mainly survey most related prior research work in resource selection and briefly talk about resource representation and results merging.

The first step in federated search is to obtain representative resource descriptions from available information sources. The START protocol [11] provides accurate information in collaborative federated search environments, but it does not work for uncooperative environments. On other side, the query-based sampling technique [4] has been widely used in federated search to obtain sample documents from each source by issuing randomly generated queries. In particular, the query-based sampling approach is used in this work to acquire sample documents from available sources. After that, all sample documents are merged together as a *centralized sample database*.

Resource selection selects a small set of most relevant sources for each user query [3][8][13]. Most early resource selection algorithms treat each individual source as a single big document which they extract summary statistics from. Those big document methods such as KL [27], CORI [6], and CVV [28] utilize different types of summary statistics of sources and finally rank available sources by matching the statistics with the user’s query. These methods ignore the boundaries of individual documents within individual sources, which limits their performance of identifying sources with a large number of relevant documents.

Some recent resource selection algorithms such as ReDDE [20], DTF [9][10], CRCS [18] and SUSHI [24] step away from treating each source as a single big document. Those algorithms often analyze individual sample documents within resource representation for ranking sources. For example, the ReDDE selection algorithm estimates the distribution of relevant documents by treating top-ranked sample documents as a representative subset of relevant documents in available sources. Related algorithms such as UUM [22], RUM [23] and CRCS [18] have been proposed, which use different methods to weight top-ranked documents and estimate the probability of relevance.

More recent resource selection algorithms such as the classification-based resource selection in federated search [1] or vertical search [2] treat resource selection as a classification problem. A classification model can be learned from a set of training queries and is used to predict the relevance of a source for test queries. It has been shown [1] that the classification approach can outperform state-of-the-art resource selection algorithms like ReDDE.

Existing resource selection methods utilize evidence within

individual sources to judge their relevance but ignore the evidence of the relationship between available sources. However, the relationship evidence is a valuable piece of information, which promises to improve the accuracy of resource selection.

Two other related research work in [25][7] learn from the results of past queries for resource selection. However, these two methods do not model the relationship between information sources and do not use formal models based on classification.

The last step of federated search is results merging, which merges returned documents from selected sources into a single list. The most effective method is to download and recalculate scores for all returned documents within a centralized retrieval model, but this is often inefficient. More efficient methods such as the CORI merging formula, the SSL [21] and the SAFE merging algorithms [19] try to approximate the results of centralized retrieval in different ways.

3. CLASSIFICATION MODEL

3.1 Classification Approach

Many resource selection algorithms are unsupervised and provide one source of evidence. To combine different evidence in a unified framework, one needs a training dataset, usually in the form of binary judgments on sources. Specifically, given a set of sources \mathcal{C} and a set of training queries \mathcal{Q} , the objective is to find a mapping \mathcal{F} of the form

$$\mathcal{F} : \mathcal{Q} \times \mathcal{C} \rightarrow \{+1, -1\}$$

where +1 indicates the relevance between the query and the source, and -1 indicates irrelevance.

Arguello et al.[1] have proposed a method to construct those judgments. Each query $q \in \mathcal{Q}$ will be issued to a full-dataset index for searching. A source $C_i \in \mathcal{C}$ is considered to be relevant with q if more than τ documents from C_i are present in top T of the full-dataset result. Otherwise, it is marked as irrelevant.

While this method can produce a rank list that mimics the rank list produced by a full-dataset retrieval, it is difficult to apply in a real world environment because of the absence of a full-dataset. We propose an alternative method that could be more feasible. A query q is now issued to each remote source C_i and we only count their returned documents that are relevant. Top T documents from each source will be inspected, then a source is marked as relevant if it has more than τ relevant documents presenting in that list. In our work, we set $T = 100$. For dataset with a large average number of relevant documents per query (over 100), we set $\tau = 3$; otherwise τ is equal to 1.

3.2 Sources of Evidence

This section presents different types of evidence of individual sources for building our classification model.

3.2.1 Big Document

Big Document (BIGDOC) approach treats each information source as a big document that contains all of its sample documents. A query is then issued to an index which contains a set of big documents, each representing one source. Sources are then ranked by how their merged sample documents match the query. The disadvantage of this method is that it does not take into account the variation of sources’

sizes. Assuming that the sampling process is uniform, for a very big source, the sampling process only covers a small fraction of its documents. Therefore, it may present fewer relevant documents in the centralized sample database than a much smaller one, although the absolute number of relevant documents in the big source is higher. Without considering the sources' sizes, it would be misleading to conclude that the small source is the better choice. Nevertheless, when combined with other features, BIGDOC could have a good contribution, especially in the case that many sources contain roughly the same number of documents. While CORI (discussed in the next part) also treats each source as one document, BIGDOC approach is more flexible since it can be used with different retrieval algorithms. In our experiments, the algorithm is Indri [14]. For each pair of a query and a source, one BIGDOC feature is built from the sample documents.

3.2.2 CORI

The CORI resource selection algorithm [6] uses Bayesian Inference Network model to rank sources. The belief $P(q|C_i)$ that a source C_i satisfies query q is the combination of multiple $P(r_j|C_i)$, the belief corresponding to each term r_j of query q . CORI applies a variant of *tf.idf* formula to determine each $P(r_j|C_i)$ and combine them together to calculate the final belief score of each source. CORI was proven to have robust performance for resource selection. In our experiments, one CORI feature is used for each pair of a query and a source.

3.2.3 Geometric Average

In this method, a query is first issued to a centralized sample database, which was mentioned in section 2. Then, each source C_i is scored according to the geometric average query likelihood of its top K sample documents [17],

$$GAVG_q(C_i) = \left(\prod_{j=1}^K P(q|d_{ij}) \right)^{\frac{1}{K}}$$

where d_{ij} is the j -th sample document in the rank list of source C_i . If C_i presents less than K documents in the rank list, the product above is padded with the minimum query likelihood score.

3.2.4 Modified ReDDE & ReDDE.top

Recall that ReDDE score [20] is calculated according to :

$$ReDDE_q(C_i) = \frac{\mathcal{N}_i^{est}}{\mathcal{N}_i^{samp}} \times \sum_{d \in \mathcal{R}_N^{samp}} \mathcal{I}(d \in C_i) \times P_q(rel|d)$$

where \mathcal{R}_N^{samp} is the top N documents returned from searching the centralized sample database. \mathcal{N}_i^{est} is the *estimated size* of source C_i , \mathcal{N}_i^{samp} is the *sample size* of C_i , and $\mathcal{I}(\cdot)$ is the indicator function. The number of top returned documents, N , is equal to $\alpha \times \mathcal{N}_{all}^{est}$, where \mathcal{N}_{all}^{est} is the estimated total number of documents of all sources and α is a constant, which is usually in the range 0.002-0.005.

ReDDE uses a step function to estimate $P_q(rel|d)$, the probability that the document d is relevant to query q . For all top N documents, that probability is equal to a constant. In our experiment, we use a modified version of ReDDE, which replaces $P_q(rel|d)$ by $P(q|d)$, the retrieval score of document d with respect to query q . The Indri retrieval

algorithm [14] is used for searching the centralized sample database. The modified ReDDE feature has been shown empirically better than the original ReDDE feature. There is one modified ReDDE feature for each pair of a query and a source.

ReDDE.top [1] is another variant of ReDDE. Unlike ReDDE, ReDDE.top set a specific number to N . In our experiment, we add another two ReDDE.top features with $N = 100$ and $N = 1000$ respectively.

4. JOINT CLASSIFICATION MODEL

4.1 Probabilistic Discriminative Model

We propose a novel joint probabilistic model for the resource selection task. First of all, a logistic model is built to combine all the features of individual sources. We refer to this model as the independent model (Ind).

Let $\vec{v} = \{v_1, \dots, v_n\}$ be the relevance vector. $v_i = 1$ indicates that the i -th source is relevant, otherwise $v_i = 0$. The relevance probability of a source c_i given its feature vector $\vec{f}(c_i)$ is calculated as:

$$P(v_i = 1|c_i) = \frac{\exp(\vec{f}(c_i) \cdot \vec{\theta})}{1 + \exp(\vec{f}(c_i) \cdot \vec{\theta})}$$

where $\vec{\theta}$ denotes the combination weight vector. For simplicity, the vector $\vec{f}(c_i)$ contains the bias feature (which is 1 for every pair of a query and a source) and the weight vector $\vec{\theta}$ contains the bias element θ_0 . The conditional probability of \vec{v} given n sources is:

$$P(\vec{v}|\vec{c}) = \frac{1}{\mathcal{Z}} \exp\left(\sum_i^n \log(P(v_i = 1|c_i)^{v_i} P(v_i = 0|c_i)^{1-v_i})\right)$$

where \mathcal{Z} is the normalizing constant.

Our joint classification model (Jnt) expands the above formula with a new term to model the relationship between sources. The conditional probability of \vec{v} given n sources is now:

$$P(\vec{v}|\vec{c}) = \frac{1}{\mathcal{Z}'} \exp\left(\sum_i^n \log(P(v_i = 1|c_i)^{v_i} P(v_i = 0|c_i)^{1-v_i}) + \frac{\alpha}{|\vec{v}|} \sum_{i,j(i < j)} sim(c_i, c_j) v_i v_j\right)$$

which can be rewritten as:

$$P(\vec{v}|\vec{c}) = \frac{1}{\mathcal{Z}'} \exp\left(\sum_i^n (1 - v_i)(\vec{f}(c_i) \cdot \vec{\theta}) - \log(1 + \exp(\vec{f}(c_i) \cdot \vec{\theta})) + \frac{\alpha}{|\vec{v}|} \sum_{i,j(i < j)} sim(c_i, c_j) v_i v_j\right)$$

where $sim(c_i, c_j)$ denotes the similarity between two sources c_i and c_j , and \mathcal{Z}' is another normalizing constant.

The parameter α controls the influence of similarity. If $|\alpha|$ is high, the model tends to promote only similar (or dissimilar) sources. When $\alpha = 0$, we get back to the independent model.

In the learning step, we learn the feature weight vector $\vec{\theta}$ from the independent model by using logistic regression. This vector is then used in the joint model. Learning α , however, is generally intractable. One can see that the space of vector \vec{v} is 2^n , and so inferencing and estimation become

impossible when n is large. We resolve this issue by first ranking the sources using the independent model, and then apply the joint classification model only to the top $K = 10$ sources. This is equivalent to reranking the top K sources.

From the set of training queries, we use maximum log-likelihood estimation to learn the parameter α . Because there is no closed-form solution for the maximum of this log-likelihood function, gradient search method is used instead.

In the prediction step, for a test query, the score of each source c_i is assigned by its probability of being relevant:

$$\mathcal{R}(c_i) = P(v_i = 1|\vec{c}) = \sum_{\vec{v}\setminus v_i} P(v_1, v_2, \dots, v_i = 1, \dots, v_n|\vec{c})$$

where $\vec{v}\setminus v_i$ denotes the set of variables in \vec{v} with variable v_i omitted. In practice, the summation is taken over $K - 1$ variables and so is feasible when K is small. After that, the top K sources will be reranked according to the new score.

4.2 Similarity Metrics

4.2.1 Similarity Metric based-on Evaluation

Given a set of training queries, the similarity between two sources can be measured by looking at the set of queries for which they are both relevant. The bigger that set is, the more related they are. Specifically, we apply a cross-product formula to measure this metric:

$$SME(c_i, c_j) = \sum_{q \in \mathcal{Q}} rel(c_i, q)rel(c_j, q)$$

where \mathcal{Q} is the set of training queries, $rel(c_i, q)$ is equal to 1 if source c_i is relevant to query q based on the classification approach described above, otherwise it is 0. This method is called Similarity Metric based-on Evaluation (SME).

4.2.2 Similarity Metric based-on Query-specific Evaluation

One issue with the SME is that it is independent of the query. A source may be highly related with another source with respect to a query but unrelated with that source with respect to another query. Therefore, it is better to incorporate the similarity between queries into this formula. By extending the above SME, we derive another metric called Similarity Metric based on Query-specific Evaluation (SMQE).

$$SMQE_q(c_i, c_j) = \sum_{q' \in \mathcal{Q}} sim(q, q')rel(c_i, q')rel(c_j, q')$$

where $sim(q, q')$ denotes the correlation (or similarity) between the test query q and a training query q' . There are many studies that explore the topicality or classification of queries, however, in this paper, we choose one simple approach. A query in consideration is issued to the centralized sample database, and the number of documents from each source that appear in top M documents of the result is recorded. In our work, M is equal to 100. The correlation between two queries is derived by a cosine-like formula:

$$sim(q, q') = \frac{\sum_i numdoc(q, c_i)numdoc(q', c_i)}{\sqrt{\sum_i numdoc(q, c_i)^2} \sqrt{\sum_i numdoc(q', c_i)^2}}$$

where $numdoc(q, c_i)$ is the number of documents of source c_i that appear in the top M documents returned from query q .

Both the SME and SMQE metrics can be modified in many ways. First of all, the term $rel(i, q)$ can be represented either by a binary number or the absolute number of relevant documents. Or we can set different thresholds to the searching on the centralized sample database. Another choice is to normalize the relevance vector. However, in our experiments, those changes do not have much effect on the results. In fact, SMQE provides the best result, proving that it better reflects the relationship between sources.

4.2.3 Similarity-Metric based-on Kullback-Leibler divergence

This method tries to reveal the similarity between sources by looking at their own vocabularies. Specifically, a language model [16] is built for each sample source. Then we calculate the Kullback-Leibler divergence between those two language models. Recall that the Kullback-Leibler divergence is actually the distance between two probabilistic models, which is the inverse of their similarity. However, because our model can adapt this change by inferring a negative similarity coefficient α , we keep the KL-value as it is. This metric is referred to as SMKL.

5. EXPERIMENTAL METHODOLOGY

We evaluate our proposed algorithms on 3 datasets. The first two datasets are well-known TREC testbeds, the last one comes from a real world application.

- **TREC123-100col-bysource (TREC123):** 100 collections (information sources) were created from TREC CDs 1,2 and 3 [3]. They are organized by publication source and publication date. The size of each source varies from 7,000 to 39,700 documents (see Table 1 for more statistics). This testbed comes with 100 queries (TREC topics 51-150) with judgments.
- **TREC4-100col-bysource (TREC4):** 100 collections were created according to the publication source of documents in TREC4 [26]. One actual publication source is distributed across a number of information sources depending on its total number of documents. Each information source has roughly 5,675 documents. This testbed comes with 50 queries (TREC topics 201-250) with judgments. More statistics about both TREC123 and TREC4 are presented in Table 1.
- **Digital Library (DIGLIB):** This real world dataset contains 80 sources (i.e., digital libraries) that are accessible from Purdue University Libraries¹. This testbed presents a heterogeneous sources of information. Each document from those sources composes of many fields. Three fields that convey rich information are the abstract, subject heading (or document's category) and full text. Not all sources provide all those three fields. A document from one source may not be provided with its full text, while a document from another source may not have the subject heading. Table 2 shows that only 65% of 80 sources provide abstracts, 65% provide subject headings, and only 30% provide full texts. In our current work, we temporally merge all those available information into one document. Future research may

¹We make the dataset available as feature file at <http://www.cs.purdue.edu/homes/dthong/>

Table 1: Summary Statistics of TREC123 and TREC4

Testbed	Size (GB)	Number of Documents (x1000)			Size(MB)		
		Min	Avg	Max	Min	Avg	Max
TREC123	3.2	0.7	10.8	39.7	28	32	42
TREC4	2.0	5.6	5.6	5.6	4	20	138

Table 2: Statistical Information about DIGLIB: Number of Sources Corresponding to their Available Information Fields

	Abstract	Subject	Full Text
Number of Sources	52(65%)	52(65%)	24(30%)

consider to treat each source differently according the type of information that is available.

We also build a set of 100 queries, some of them are extracted from the library log, which are real queries. For each pair of a query and a source, we manually assign a binary value to indicate their relevance.

A note on resource specific retrieval algorithm: DIGLIB is a real world application of digital libraries, each of its sources implements a different retrieval algorithm, which is not known. We can only access those sources through a unified interface. For TREC123 and TREC4, we assign one retrieval algorithm to each source in a round-robin manner. The set of assigned algorithms is Inquiry, Language Model and Vector Space (tf.idf). These algorithms influence the query-based sampling process, as well as the classification process. A less effective retrieval algorithm like Vector Space model may reduce a source’s chance of being marked as relevant.

For each testbed, we repeat every experiment 5 times. In each trial, we randomly select 50% of the queries as training set, and test on the other 50%. All the results shown in the next section are averaged over 5 trials.

Each source is sampled with 300 documents. We also compare the main results with 100 sample documents. The experiments are measured on several levels:

- **Source Level (Resource Selection), Accuracy:** This level measures the precision of the resource selection algorithms. Top sources (i.e., top 10) are judged by their precisions at different levels. The judgments come from the classification method, as described in section 3.1. We report the results at $P@\{1, 3, 5, 10\}$ accordingly.
- **Source Level, Recall Metric (R-metric):** This metric is widely used in comparing resource selection algorithms [3]. Let E denote the ranking produced by a resource selection algorithm, B denote the based line ranking, in this case, the Relevance-Based Ranking. At level k , the R-metric is defined as:

$$R_k = \frac{\sum_{i=1}^k E_i}{\sum_{i=1}^k B_i}$$

Let E_i denote the number of relevant documents of the i -th source according to the ranking E , B_i denote the same thing with respect to ranking B . We also report the results at $P@\{1, 3, 5, 10\}$ as in source level accuracy.

- **Document Level, High Precision:** To make it independent from result merging algorithm, we use full-dataset retrieval as our merging method. The chosen retrieval algorithm is Inquiry in Lemur Toolkit [5]. For each query, top 5 sources from the joint classification rank list are selected for this step. Documents not from selected sources are filtered out from the full-dataset rank list. The remaining list is checked by their precision at $P@\{5, 10, 15, 20, 30\}$ accordingly. We also report a full-dataset precision which includes all sources.

On TREC123 and TREC4, all tests at different levels are presented. On DIGLIB dataset, we only report the results at source level because the document judgments are difficult to make as many sources do not provide their full text information. In most of the experiments, SMQE is used as our default similarity metric. However, in section 6.4, we also discuss the experimental results with different other metrics.

6. EXPERIMENTAL RESULTS

In all of our experiments, we use paired t-test on queries to check significance. A * denotes a significance on $p < 0.1$ level; † corresponds to $p < 0.05$ level and ‡ corresponds to $p < 0.01$ level.

6.1 TREC123 & TREC4

First of all, we compare the joint classification model with the independent model on the two TREC testbeds. Table 3 represents the source level results in accuracy on TREC123 and TREC4. The second column of each dataset is the joint classification model. Numbers in parentheses show the relative improvement of the joint classification model (denoted as “Jnt”) over the independent model (denoted as “Ind”).

Table 4 shows the R-metric comparison between the independent model and joint classification model. Table 5 shows the high precision at document level. We also report the full centralized retrieval, which includes all sources. This is denoted as the “Full” column in the table.

It can be seen that joint classification model always leads to better results than independent model, as it shows in all three tables. Both models have the same source level accuracy and R-metric values at top 10 because of the fact that we rerank the top 10 sources. The results are more statistically significant on TREC123 than on TREC4. This can be explained as in TREC123, we have trained on 50 queries; whereas in TREC4, we use only 25 queries out of 50 for training.

6.2 Digital Library

The result at source level of Digital Library is reported in Table 6. In this real world dataset, the joint classification model significantly outperforms the independent model. This accounts to the fact that many sources only provide

Table 3: Source Level Results in Accuracy on TREC123 & TREC4 with 300 Sample Documents

Src Rank	TREC123		TREC4	
	Ind	Jnt	Ind	Jnt
@1	0.512	0.524(2.3%)	0.480	0.536(11.7%)
@3	0.456	0.499(9.4%) †	0.451	0.475(5.3%)
@5	0.451	0.484(7.3%)*	0.430	0.446(3.7%)
@10	0.439	0.439(0%)	0.414	0.414(0%)

Table 4: Source Level Results in R-metric on TREC123 & TREC4 with 300 Sample Documents

Src Rank	TREC123		TREC4	
	Ind	Jnt	Ind	Jnt
@1	0.262	0.319(21.8%) ‡	0.287	0.309(7.7%)
@3	0.309	0.364(17.8%) ‡	0.324	0.340(4.9%)
@5	0.354	0.400(13.0%) ‡	0.343	0.355(3.5%)
@10	0.426	0.426(0%)	0.414	0.414(0%)

partial information about themselves. This also shows that the joint classification model can alleviate the problem of missing information.

6.3 Tests with Different Sample Sizes

We conduct experiments on three datasets with only 100 documents sampled from each source. This test is to show the robustness of the model, as well as the effect of sampling size on the results. The results of source level (both in accuracy and R-metric) and document level are reported for TREC123 and TREC4 (Table 7, Table 8 and Table 9 respectively), while only source level is reported for DIGLIB (Table 10).

The sample size clearly affects TREC123. Its performance of the independent model drops significantly. However, this also leaves room for joint classification model to show its effectiveness: the accuracy on source level is statistically more significant. On document level, the improvement is a bit weaker. This can be explained as the initial choice of top 10 sources from the independent model is less precise, so is the joint classification model, which uses the initial ranking list directly.

Most results on TREC4 from Table 7 to Table 9 indicate the advantage of the joint classification model against independent model with a small number of sample documents, although the difference is smaller than TREC123 due to the limited amount of training information.

The results on DIGLIB (Table 10) are also consistent. The performances of both resource selection algorithms drop with 100 sample documents. However, the results of the joint classification method are still significantly better than those of the independent method.

6.4 Test with Different Similarity Metrics

We conduct tests on three testbeds with different similarity metrics discussed in Section 4.2. Figure 1 shows the

Table 5: Document Level Results in High Precision on TREC123 & TREC4 with 300 Sample Documents

Docs Rank	TREC123		
	Full	Ind	Jnt
@5	0.446	0.392	0.410(4.6%)
@10	0.444	0.355	0.360(1.4%)
@15	0.435	0.332	0.347(4.5%)*
@20	0.430	0.309	0.326(5.5%) †
@30	0.414	0.280	0.300(7.1%) ‡

Docs Rank	TREC4		
	Full	Ind	Jnt
@5	0.549	0.282	0.290(2.8%)
@10	0.459	0.238	0.254(6.7%)
@15	0.422	0.209	0.224(7.2%)
@20	0.384	0.186	0.200(7.5%)
@30	0.354	0.167	0.170(1.8%)

Table 6: Source Level Results in Accuracy on DIGLIB with 300 Sample Documents

Src Rank	DIGLIB	
	Ind	Jnt
@1	0.552	0.640(15.9%) †
@3	0.460	0.536(16.5%) ‡
@5	0.419	0.487(16.2%) ‡
@10	0.356	0.356(0%)

Table 7: Source Level Results in Accuracy on TREC123 & TREC4 with 100 Sample Documents

Src Rank	TREC123		TREC4	
	Ind	Jnt	Ind	Jnt
@1	0.320	0.380(18.8%)*	0.496	0.480(-3.2%)
@3	0.299	0.373(24.7%) ‡	0.405	0.411(1.5%)
@5	0.318	0.357(12.3%) ‡	0.379	0.403(6.3%)
@10	0.319	0.319(0%)	0.367	0.367(0%)

Table 8: Source Level Results in R-metric on TREC123 & TREC4 with 100 Sample Documents

Src Rank	TREC123		TREC4	
	Ind	Jnt	Ind	Jnt
@1	0.183	0.233(27.3%) †	0.278	0.317(14%)
@3	0.214	0.262(22.4%) ‡	0.264	0.293(11%)
@5	0.244	0.279(14.3%) †	0.293	0.311(6.1%)
@10	0.311	0.311(0%)	0.341	0.341(0%)

Table 9: Document Level Results in High Precision on TREC123 & TREC4 with 100 Sample Documents

Docs Rank	TREC123		TREC4	
	Ind	Jnt	Ind	Jnt
@5	0.328	0.329(0.3%)	0.283	0.301(6.4%)
@10	0.302	0.316(4.6%)	0.243	0.254(4.5%)
@15	0.288	0.306(6.2%)*	0.223	0.227(1.8%)
@20	0.277	0.296(6.9%)†	0.195	0.204(4.6%)
@30	0.253	0.268(5.9%)†	0.165	0.166(0.6%)

Table 10: Source Level Results in Accuracy of DIGLIB with 100 Sample Documents

Src Rank	DIGLIB	
	Ind	Jnt
@1	0.436	0.620(42.2%)†
@3	0.383	0.531(38.6%)†
@5	0.375	0.474(26.4%)†
@10	0.318	0.318(0%)

results of TREC123 and TREC4 at document level. From this figure, we notice that the SMQE method outperforms all other metrics, due to the fact that it considers the similarity between queries. The SME produces a quite close-to-best result, but the SMKL tends not to be a good choice for the joint classification model. On TREC4, SMKL is comparable with independent model, but it is worse than SMQE and SME.

Figure 2 shows the results of DIGLIB at source level. In this case, both SMQE and SME are comparable, except for the precision at top 1. Again SMKL is not a good choice.

7. CONCLUSION & FUTURE WORK

This paper proposes a novel joint probabilistic classification model for the resource selection task in federated text search. Existing resource selection algorithms only utilize evidence of individual information sources to select relevant sources, but they do not model the valuable relationship information between the sources. The proposed algorithm estimates the probability of relevance of information sources in a joint manner by combining both the evidence of individual sources and the relationship between the sources. The importance of different types of evidence is determined in a discriminative manner for maximizing the accuracy of resource selection with some training queries. Different types of similarity metrics have been explored to model source similarity based on the performance of available sources on training queries and the Kullback-Leibler divergence on the contents of the sources. A set of experiments were conducted with two TREC datasets and one real world application with digital libraries. The empirical results in different configurations have demonstrated the effectiveness of the proposed joint classification model.

There are several directions to extend the research work in the paper. First, one advantage of the proposed joint probabilistic model is to integrate different types of evidence of

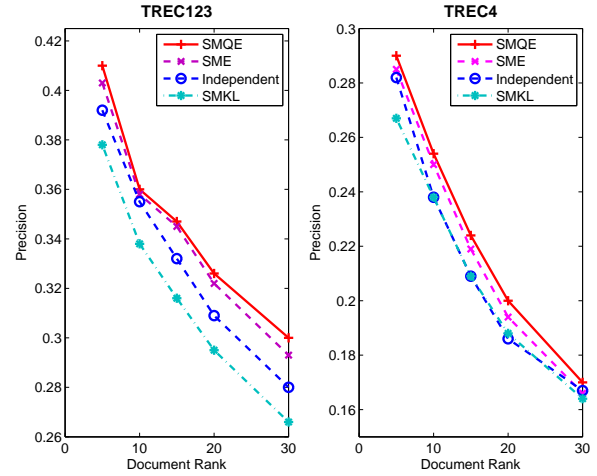


Figure 1: Document Level High Precision on TREC123 & TREC4 with Different Similarity Metrics

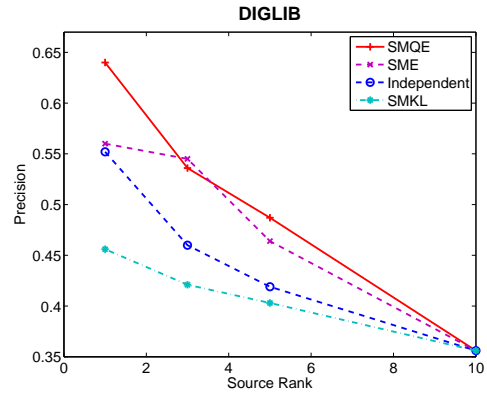


Figure 2: Source Level Accuracy on DIGLIB with Different Similarity Metrics

individual sources and their relationship. We plan to explore more features for improving the performance of resource selection. For example, we can combine multiple types of similarity evidence in a single framework (with different α weights), which may better model sources' relationship for more accurate resource selection. Second, the joint model in this paper utilizes a reranking approach in resource selection with a small set of information sources (e.g., top 10) to avoid large computational complexity. It is possible to break this limit by utilizing some other approximate inference algorithms (e.g., the pseudo likelihood approach [15]) or making further assumptions on the sources' relationship. For example, one strategy is to first divide available sources into groups of closely related sources. Inference can be conducted by building a small model in each group and assuming independence of sources between different groups.

8. ACKNOWLEDGMENTS

This research was partially supported by the Vietnam Education Foundation (VEF) and the NSF grant IIS-0749462.

The opinions, findings, and conclusions stated herein are those of the authors and do not necessarily reflect those of the sponsors.

9. REFERENCES

- [1] J. Arguello, J. Callan, and F. Diaz. Classification-based resource selection. *Proceeding of the 18th ACM Conference on Information and Knowledge Management*, 2009.
- [2] J. Arguello, F. Díaz, J. Callan, and J. Crespo. Sources of evidence for vertical selection. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2009.
- [3] J. Callan. Distributed information retrieval. *Advances in Information Retrieval*, pages 127–150, 2000.
- [4] J. Callan and M. Connell. Query-based sampling of text databases. *ACM Transactions on Information Systems*, 19(2):97–130, 2001.
- [5] J. Callan, W. B. Croft, and S. M. Harding. The inquiry retrieval system. In *Proceedings of the Third International Conference on Database and Expert Systems Applications*, 1992.
- [6] J. Callan, Z. Lu, and W. B. Croft. Searching distributed collection with inference networks. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1995.
- [7] S. Cetintas, L. Si, and H. Yuan. Learning from past queries for resource selection. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management*. ACM, 2009.
- [8] N. Craswell, P. Bailey, and D. Hawking. Server selection on the world wide web. In *Proceedings of the 5th ACM Conference on Digital Libraries*. ACM, 2000.
- [9] N. Fuhr. A decision-theoretic approach to database selection in networked ir. *ACM Transactions on Information Systems (TOIS)*, 17(3):229–249, 1999.
- [10] N. Fuhr. Resource discovery in distributed digital libraries. In *In Digital Libraries '99: Advanced Methods and Technologies, Digital Collections*, 1999.
- [11] L. Gravano, K. Chang, C.-C., H. García-Molina, and A. Paepcke. Starts: Stanford proposal for internet meta-searching. In *Proceedings of the ACM-SIGMOD International Conference on Management of Data (SIGMOD)*. ACM, 1997.
- [12] L. Gravano, H. García-Molina, and A. Tomasic. Gloss: Text-source discovery over the internet. *ACM Transactions on Database Systems*, 24(2):229–264, 1999.
- [13] W. Meng, C. Yu, and K. Liu. Building efficient and effective metasearch engines. *ACM Computing Surveys (CSUR)*, 34(1):48–89, 2002.
- [14] D. Metzler and W. Croft. Combining the language model and inference network approaches to retrieval. *Information Processing and Management*, 40(5):735–750, 2004.
- [15] S. Parise and M. Welling. Learning in markov random fields: An empirical study. In *Joint Statistical Meeting (JSM2005)*, volume 4, 2005.
- [16] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1998.
- [17] J. Seo and W. B. Croft. Blog site search using resource selection. In *CIKM '08: Proceeding of the 17th ACM Conference on Information and Knowledge Management*, pages 1053–1062, New York, NY, USA, 2008. ACM.
- [18] M. Shokouhi. Central-rank-based collection selection in uncooperative distributed information retrieval. In *Proceedings of the 29th European Conference on Information Retrieval*, 2007.
- [19] M. Shokouhi and J. Zobel. Robust result merging using sample-based score estimates. *ACM Transactions on Information Systems*, 27(3):1–29, 2009.
- [20] L. Si and J. Callan. Relevant document distribution estimation method for resource selection. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2003.
- [21] L. Si and J. Callan. A semi-supervised learning method to merge search engine results. *ACM Transactions on Information Systems*, 21(4):457–491, 2003.
- [22] L. Si and J. Callan. Unified utility maximization framework for resource selection. In *Proceedings of 13th ACM International Conference on Information and Knowledge Management (CIKM)*, 2004.
- [23] L. Si and J. Callan. Modeling search engine effectiveness for federated search. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2005.
- [24] P. Thomas and M. Shokouhi. Sushi: scoring scaled samples for server selection. In *SIGIR '09: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2009.
- [25] E. Voorhees, N. K. Gupta, and B. Johnson-Laird. Learning collection fusion strategies. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1995.
- [26] J. Xu and J. Callan. Effective retrieval with distributed collections. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1998.
- [27] J. Xu and W. B. Croft. Cluster-based language models for distributed retrieval. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1999.
- [28] B. Yuwono and D. L. Lee. Server ranking for distributed text retrieval systems on the internet. In *Proceedings of the 5th Annual International Conference on Database Systems for Advanced Applications*, 1997.