Classification-Based Resource Selection

Jaime Arguello
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA, USA
jaime@cs.cmu.edu

Jamie Callan
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA, USA
callan@cs.cmu.edu

Fernando Diaz
Yahoo! Labs Montreal
1000 Rue de la Gauchetière
Suite 2400
Montreal, QC H3M4W5
diazf@yahoo-inc.com

ABSTRACT

In some retrieval situations, a system must search across multiple collections. This task, referred to as federated search, occurs for example when searching a distributed index or aggregating content for web search. Resource selection refers to the subtask of deciding, given a query, which collections to search. Most existing resource selection methods rely on evidence found in collection content. We present an approach to resource selection that combines multiple sources of evidence to inform the selection decision. We derive evidence from three different sources: collection documents, the topic of the query, and query click-through data. We combine this evidence by treating resource selection as a multiclass machine learning problem. Although machine learned approaches often require large amounts of manually generated training data, we present a method for using automatically generated training data. We make use of and compare against prior resource selection work and evaluate across three experimental testbeds.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Miscellaneous

General Terms

Algorithms

Keywords

federated search, distributed information retrieval, resource selection, query classification

1. INTRODUCTION

Classic information retrieval systems model search under the assumption of a centralized index. Federated search systems model search across multiple, distributed collections. We focus on one subtask of federated search, *resource selection*, the task of deciding, given a query, which collections to search. The objective of resource selection is to select a few collections whose merged ranking approximates the performance of a ranking generated by searching a centralized index of all collection content.

Traditional approaches to resource selection model the relevance of a collection by analyzing documents from the collection. For example, relevance can be modeled by comparing the text in the query to the text in a collection with metrics used in document retrieval [26, 5, 23]. Relevance can also be modeled as an expectation of the number of relevant documents in a collection as in the ReDDE system [9, 21]. Parameters for these models are usually tuned manually on a small set of training queries.

In this work, we model a collection's relevance based on its impact on a full-dataset retrieval, one that merges content from all collections. We make the following assumption: *given a query, an effective partial-dataset retrieval will resemble a full-dataset retrieval.* The idea is that we want a retrieval that merges content from a few collections to be indistinguishable from one that merges content from all collections. Because users scan results from top to bottom, collections should be prioritized by their contribution of documents to the top ranks of a full-dataset retrieval. We train a classification system that models the inclusion of a collection in the merged results as a function of a set of features. Training data is harvested from full-dataset retrievals conducted offline.

Modeling resource selection as a classification problem allows us to easily incorporate a diverse set of evidence as input features. This evidence can be classified into three categories. *Corpus-based features* derive evidence from collection documents. These include traditional resource selection metrics such as ReDDE. *Query-categorical features* derive evidence from the topic of the query. Finally, *click-through features* derive evidence from queries with clicks on collection documents. In a federated search environment, click-through data can be collected by the portal interface.

A classification-based approach provides several advantages over traditional approaches. First, it is flexible. Depending on the federated search environment, different features can be easily incorporated into the model. Second, it is easy to train. As long as the system has offline access to full-dataset retrievals, more training data can be generated. Third, it is general. Although we adopt a particular machine learning method in this paper, any multiclass learning

CIKM'09, November 2–6, 2009, Hong Kong, China.

method can be used to automatically tune parameters given training data. Finally, it is effective. We demonstrate that in the majority of experimental settings, a classification-based approach significantly outperforms traditional resource selection approaches.

2. RELATED WORK

Most resource selection methods share two things in common: they formulate the problem as resource ranking (i.e., prioritizing collections for selection) and derive evidence from collection content (e.g., from sampled documents). Common approaches view collections (or their sampled documents) as large documents and adapt document retrieval methods to rank collections. CORI adapts INQUERY's inference net document ranking approach [5]. Xu and Croft score collections by the Kullback-Leibler divergence between the query and collection language models [26]. Si *et al.* score collections based on the query generation probability given the collection language model [23]. Large document models have the advantage of being relatively straight-forward adaptations of well studied document ranking techniques. However, they model the relevance of the *entire* collection. For this reason, they may favor a small, topically-focused collection (related to the query) when a larger, more topically-diverse collection contains more relevant documents. Instead of comparing the text in the query with that of the entire collection, Seo and Croft focus on the collection's documents most similar to the query [16]. More specifically, a collection is scored based the geometric average query likelihood from its top m documents.

Methods such as GlOSS [9] and ReDDE [21] rank collections based on their expected number of relevant documents. Similar to a large document model, GlOSS models collection relevance using a query-independent collection language model. ReDDE also scores collections by their expected number of relevant documents, but derives this expectation using a retrieval from a centralized sample index, a mix of documents sampled from each target collection [21]. ReDDE predicts a binary relevance label for every sampled document and then assumes that every relevant sampled document represents some number of relevant documents in the collection from where it originates.

Other resource selection algorithms estimate the distribution of document scores or document probabilities of relevance across collections [22, 19, 24]. Given these estimates, collections can be prioritized by their average document score or by a collection's contribution to an approximated merged ranking. Similar to ReDDE, these methods start by issuing the query to a centralized sample index, producing a retrieval score for every sampled document. A full-collection score distribution is estimated by assuming that every scored sampled document represents some number of documents in its original collection with a similar score.

Sometimes, collections are topically focused. Ipeirotis and Gravano exploit a topical relatedness between collections in order to minimize the negative effect of incomplete content descriptions derived from sampled documents [10]. Collections are first classified into a topic hierarchy using topically-focused queries and their hit counts. A collection's language model (derived from sampled documents) is smoothed with those from topically-related ones. At test time, a "flat" resource selection method (e.g., GlOSS) is applied in a top-down fashion, descending the topic/collection hierarchy.

We cast resource selection as a multiclass classification problem. Therefore, we review some prior work on query classification. Because queries are terse, query-classification approaches often augment the query with features beyond the query string, possibly derived from query-logs [2], query click-through data [25], and documents associated with target categories [17, 13, 18]. Bietzel *et al.* classify queries into semantic categories using an (unlabeled) query-log and a technique known as *selectional preference*. The query "interest rates" belongs to target category *finance* because terms "interest" and "rates" often occur in contexts that co-occur with known finance-related terms. Shen *et al.* [17] and other participants of the KDD 2005 Cup [13] use corpus-based evidence. The query is issued to an index where every document is associated (heuristically) with a target category. Then, similar to ReDDE, the query is classified based on the number of top-ranked documents associated with each category. In later work, Shen *et al.* derive a soft membership of documents to target categories using term similarity [18], after augmenting the category representation with related terms using pseudo-relevance feedback. Li *et al.* take a different approach [12]. Instead of enriching the query representation, classifiers are trained using purely query string features. However, the amount of training data is expanded by propagating class labels to unlabeled queries using a large click-graph.

In the context of web search, vertical selection refers to the decision of whether to include content from specialized collections in web search results. In previous work, Diaz proposes a model for predicting whether to include news content based on user click feedback [6]. Arguello *et al.* address the situation where 18 verticals can be integrated into web results [1]. Finally, Diaz and Arguello propose several methods for improving the performance of classification-based vertical selectors by incorporating implicit user feedback [7].

3. PROBLEM DEFINITION

Given a set of n collections and a query q, a resource selector picks k collections from which to retrieve documents. We assume that the rankings from different collections can be merged as though all documents were centrally indexed. This separates the performance of the merging algorithm from resource selection evaluation. When $k = n$, rankings are equivalent to those generated from a centralized collection. We refer to this as a full-dataset retrieval. When $k < n$, we expect performance to be inferior to a full-dataset retrieval. Our objective, therefore, is to perform as well as possible for a given value k.

4. CLASSIFICATION APPROACH

Our classification approach takes the form of n one-vs-all logistic regression models (one per collection).[1] Given a test query, each classifier makes a binary prediction with respect to its collection. Collections are then prioritized based on $P_i(Y = 1|q)$, the confidence of a positive prediction from collection C_i's classifier.

Training collection-specific classifiers requires training data in the form of binary judgements on collections. If $\mathcal{Q}$ denotes the set of training queries and $\mathcal{C}$ the set of target collections,

[1] http://www.csie.ntu.edu.tw/cjlin/liblinear/

we require a function of the form,

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

which maps query-collection pairs to +1, if C_i is relevant to q, and −1, if C_i is not relevant to q.

As mentioned in the introduction, our objective is to learn a model that selects collections based on their contribution to a full-dataset retrieval. This is based on the assumption that, on average, a full-dataset retrieval ($k = n$) outperforms a partial dataset retrieval ($k < n$). Given a full-dataset retrieval of query q, we generate a true label for every collection $C_i \in \mathcal{C}$ as follows. With respect to C_i, query q is a positive instance, +1, if more than τ documents from C_i are present in the top T full-dataset results. Otherwise, q is a negative instance, −1, with respect to C_i.

We set $T = 30$ because we evaluate merged results in terms of $\mathcal{P}@\{5, 10, 30\}$ and set $\tau = 3$ in order to ignore collections that contribute only a few documents to the top 30. We do not claim this is an optimal parameter setting. One alternative would have been to train different models using $T = \{5, 10, 30\}$ when evaluating based on $\mathcal{P}@\{5, 10, 30\}$, respectively.

5. SOURCES OF EVIDENCE

Our approach to resource selection is to exploit sources of evidence known to be effective in previous work. In addition, we propose several other signals likely to be correlated with a collection's impact on a merged retrieval. We focus on features derived from three sources of evidence: collection documents, the query topic, and collection query-click behavior.

Some of the signals require conducting a retrieval from a particular index, for example from a centralized sample index, which combines documents sampled from each target collection. Unless stated otherwise, all retrievals were conducted using Markov random field retrieval [14]. The parameter settings of the algorithm were taken from prior work and have been shown to perform well across various collections and tasks [14].

5.1 Corpus Features

Corpus features derive evidence from collection documents. For example, given a query, we might prioritize collections by their number of documents retrieved. This, however, would require searching every collection, which is impractical given our objective of approximating a full-dataset retrieval by searching only a few collections. For this reason, corpus-features are derived from sampled documents. Documents were sampled from a collection uniformly without replacement. In some federated search environments, collection documents are only accessible through a search interface. Prior work suggests that sampled sets of similar quality (as those obtained using uniform sampling) can be obtained using query-based sampling [4].

Corpus features correspond to three existing resource selection methods: CORI [5], Seo and Croft's geometric average approach [16] (GAVG), and the variant of ReDDE introduced in Arguello *et al.* [1], which we refer to as ReDDE.top. While these three methods derive evidence from the same source (i.e., collection samples), they model different phenomena. CORI and GAVG model the similarity between the query and collection text. However, CORI models the collection as one large query-independent bag of words, while GAVG focuses on the collection documents most similar to the query. ReDDE.top models the collection's average document score in a full-dataset retrieval. We incorporate these three collection scoring functions as features to investigate their relative contribution to resource selection performance.

5.1.1 CORI

CORI adapts INQUERY's inference net document ranking approach to ranking collections [5]. Here, all statistics are derived from sampled documents rather than the full collection. We use n CORI features (one per collection).

5.1.2 Geometric Average

Seo and Croft's approach [16] issues the query to a centralized sample index, one that combines document samples from every collection, and scores collection C_i by the geometric average query likelihood from its top m sampled documents,

$$\text{GAVG}_q(C_i) = \left(\prod_{d \in \text{top } m \text{ from } C_i^{\text{sampled}}} P(q|d) \right)^{\frac{1}{m}},$$

where C_i^{sampled} is the set of documents sampled from C_i and $P(q|d)$ is document d's query likelihood score. If fewer than m sampled documents are retrieved for a given collection, the product above is padded with $P_{\min}(q|d)$, the retrieval's minimum query likelihood. We use n GAVG features (one per collection).

5.1.3 ReDDE.top

Like GAVG, ReDDE.top issues to the query to a centralized sample index and scores collection C_i according to

$$\text{ReDDE.top}_q(C_i) = \mathcal{SF}_i \times \sum_{d \in \mathcal{R}_N^{\text{sampled}}} \mathcal{I}(d \in C_i) \times P(q|d),$$

where $\mathcal{R}_N^{\text{sampled}}$ denotes the top N documents in the centralized sample index retrieval and $\mathcal{SF}_i$ is the *scale factor* of collection C_i. The scale factor quantifies the difference between the size of the original collection, $|C_i|$, and the number of documents sampled from it, $|C_i^{\text{sampled}}|$,

$$\mathcal{SF}_i = \frac{|C_i|}{|C_i^{\text{sampled}}|}. \quad (1)$$

We used two sets of ReDDE.top features, one set using $N = 100$ and a second using $N = 1,000$, for the following reason. The first set accumulates scores from the top 100 sampled documents. A collection with no documents in the top 100 receives a score of zero. This is problematic, however, if the number of collections with a non-zero score is less than k, the number of collections to be selected. To increase the number of collections with a non-zero ReDDE.top feature, we used a second set of ReDDE.top features setting $N = 1,000$. We use $2n$ ReDDE.top features: n features with $N = 100$ and n features with $N = 1,000$.

5.2 Query Category Features

If collections are topically-focused, a potentially useful source of evidence is the topic of the query. We selected 166 topics from the Open Directory Project (ODP) hierarchy and crawled Web documents associated with these ODP

nodes.[2] These document sets were used to train logistic-regression classifiers (one per category) using unigram features.[3] Because queries are terse, instead of applying our trained classifiers directly on the query string, we apply them to documents in the centralized sample index and classify the query using a retrieval from this index. We set the value of category feature y_i according to,

$$\text{CAT}_q(y_i) = \frac{1}{\mathcal{Z}} \sum_{d \in \mathcal{R}_N^{\text{sampled}}} P(q|d) \left(\frac{P(y_i|d)}{\sum_{y_j \in \mathcal{Y}} P(y_j|d)} \right), \quad (2)$$

where $P(y_i|q)$ is category y_i's confidence value on document d and $\mathcal{Z} = \sum_{d \in \mathcal{R}_N^{\text{sampled}}} P(q|d)$. For these features, we set $N = 100$. We use 166 query category features (one per category).

5.3 Click-through Features

Once in operation, a resource selection system has access to user feedback in the form of clicks on collection documents. A click on a document can be viewed as a surrogate for document relevance. We view a click on a document as a surrogate for collection relevance, in favor of the collection from which the document originates. Click-through features exploit a possible correlation between collection relevance and the similarity between the test query and queries with clicks on collection documents.

Our approach is to model queries which result in a click on a collection document. For a collection, C_i, let Q_i denote all queries associated with a click event on a document in C_i (allowing duplicate queries). We index each Q_i as an individual document in a corpus of n documents. Given a query, we use the retrieval score of each collection as a feature. We use n click-through features (one per collection).

6. METHODS AND MATERIALS

6.1 Data

The TREC GOV2 test collection is a large crawl of the ".gov" portion of the Web, containing about 25M documents.[4] The GOV2 corpus was used to construct 3 experimental federated search testbeds, varying the number of target collections: 1,000, 250, and 30. We refer to these testbeds as gov2.1000, gov2.250, and gov2.30, respectively. We constructed the gov2.1000 testbed following the procedure described in Fallen and Newby [8]. While the GOV2 corpus consists of about 17,000 unique hosts (e.g., www.epa.gov), the largest 1,000 hosts contain about 90% of the GOV2 collection (i.e., about 22M documents). The gov2.1000 testbed was constructed by treated each of the largest 1,000 hosts as a separate collection.

The gov2.250 and gov2.30 testbeds were constructed by clustering hosts in the gov2.1000 testbed into 250 and 30 clusters, respectively, as follows. First, to represent hosts, we randomly sampled 1,000 documents from each. We define a host's vocabulary by all term-stems (using the Porter stemmer [15]) appearing at least 10 times in its document sample. Host-specific language models were constructed using maximum likelihood without smoothing. The distance between hosts H_i and H_j was computed using the Jeffrey divergence between their respective language models [11], also known as the symmetric Kullback-Leibler divergence,

$$D_J(\theta_i||\theta_j) = \sum_w (P(w|\theta_i) - P(w|\theta_j)) \log_2 \left(\frac{P(w|\theta_i)}{P(w|\theta_j)} \right).$$

We used average-link agglomerative clustering, iteratively merging clusters according to their hosts' average pair-wise similarity. We seeded the clustering by first combining hosts belonging to the same government entity (e.g., nih, usgs, usda, epa, uspto, nasa).

Figure 1 shows each testbed's collection size distribution. The gov2.1000 and gov2.250 testbeds have a few large collections and many small collections, while gov2.30 has many large collections and a few small ones. In the gov2.1000 testbed, 720 (72%) collections have fewer than 10,000 documents and 438 (44%) have fewer than 5,000 documents. In the gov2.250 testbed, 131 (66%) collections have fewer than 10,000 documents. In the gov2.30 testbed, 24 (80%) collections have more than 1M documents.

As described in Section 4, we train a classification system to predict the inclusion of a collection in the merged results based on its impact on a full-dataset retrieval. To this end, we require a set of queries used to produce full-dataset retrievals. There are multiple possibilities for selecting a set of "training" queries (e.g., using a query-log or generating artificial queries from collection text). However, one requirement is that there be enough positive instances for training for every collection. In other words, for each collection, there should be a sufficient number of queries with hits in the collection.

In this work, training queries were sampled from the AOL query-log. Recall that our three experimental testbeds consist of clusters of hosts from the ".gov" domain ($1,000$ singleton host clusters in the case of the gov2.1000 testbed). Click events in the AOL query-log are uniquely identified by user ID, query, date/time and host URL (i.e., for the host associated with the document clicked). Therefore, it is possible to identify all AOL click events associated with any one of our $1,000$ hosts. For each host, we estimate a query multinomial using the query's relative frequency in click events associated with documents from the host. A set of 75,000 queries was sampled (without replacement) using a two-step iterative processes. First, a host is sampled uniformly from the set of $1,000$ hosts. Then, a query is sampled from the host's query multinomial. Hosts were sampled uniformly to favor coverage across hosts and, thereby, coverage across collections in our three testbeds. Queries were sampled according to their relative frequency in click events to favor popular queries likely to have hits in the collection.

Click-through features require simulating click events on collection documents. Click-through data collected over time was simulating also using the AOL query-log. We collected a total of $305,236$ click events associated with our 1,000 ".gov" hosts. There were no click events for about 25% of hosts. Of the 75% of hosts with click events, only about half had more than 50 click events. Click events associated with a query in our test set (described later) were omitted from the set of queries used for training and from those used to simulate click-through data.

[2] http://www.dmoz.org
[3] http://www.csie.ntu.edu.tw/cjlin/liblinear/
[4] http://ir.dcs.gla.ac.uk/test_collections/gov2-summary.htm

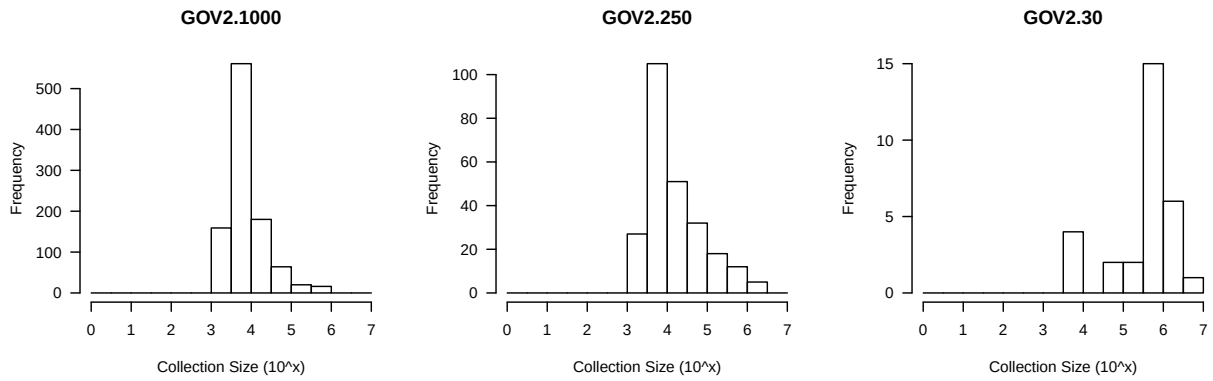

Figure 1: Collection size distribution of our three experimental testbeds.

6.2 Single-Evidence Baselines

The classification approach was evaluated against six single-evidence baselines, including one for every type of feature used in the classification approach. Corpus-based single-evidence baselines CORI, GAVG, and ReDDE.top score collections as described in Sections 5.1.1-5.1.3. ReDDE.top was used as a single-evidence baseline as follows. First collections are prioritized by ReDDE.top score using $N = 100$. A second priority list is constructed using $N = 1,000$. If the first priority list has fewer than k collections, the remaining collections are selected from the second priority list.

We also evaluated a baseline approach that scores collections by query likelihood given its click-through queries, denoted by CLICK, as described in Section 5.3.

6.2.1 ReDDE

In addition to ReDDE.top, we evaluate against the original version of ReDDE [21], which estimates the number of relevant documents in a collection. ReDDE, like GAVG and ReDDE.top, conducts a retrieval from a centralized sample index and then scores collection C_i according to,

$$\text{ReDDE}_q(C_i) \quad = \quad \mathcal{SF}_i \times \sum_{d \in C_i^{\text{sampled}}} P(rel|d),$$

where $P(rel|d)$ is the probability that document d is relevant and $\mathcal{SF}_i$ is the scale factor of collection C_i, defined by Equation 1.

ReDDE models $P(rel|d)$ as a step function, based on document d's projected rank in an unobserved full-dataset retrieval, $\hat{\mathcal{R}}^{\text{full}}(d)$, according to,

$$P(rel|d) = \begin{cases} 1 & \text{if } \hat{\mathcal{R}}^{\text{full}}(d) < \left(\tau \times |C_{all}|\right) \\ 0 & \text{otherwise,} \end{cases}$$

where $|C_{all}| = \sum_{C_j \in \mathcal{C}} |C_j|$ and τ is a parameter. Document d's projected rank in the unobserved full-dataset ranking, $\hat{\mathcal{R}}^{\text{full}}(d)$, is the sum of scale factors for collections represented by documents ranked above d in the centralized sample index retrieval, $\mathcal{R}^{\text{sampled}}$,

$$\hat{\mathcal{R}}^{\text{full}}(d) = \sum_{i=1}^{n} \left(\sum_{j=1}^{\mathcal{R}^{\text{sampled}}(d)-1} \mathcal{I}(d_j \in C_i) \times \mathcal{SF}_i \right),$$

where $\mathcal{R}^{\text{sampled}}(d)$ is document d's rank in the centralized sample index retrieval. Our baseline ReDDE algorithm corresponds to the version referred to as *modified* ReDDE in [21]. Modified ReDDE ranks collections using ReDDE with $\tau = 0.0005$. A second priority list is constructed for all collections with a ReDDE mass of less than 0.10 using $\tau = 0.003$. If the first priority list has less than k collections, the remaining collections are selected from the second priority list.

6.2.2 Category-based Similarity

Our CATS baseline scores collections based on the similarity between the topical profile of the query and the topical profile of the collection. The query's topical profile is given by normalizing Equation 2 across categories, such that $\sum_{y_j \in \mathcal{Y}} \text{CATS}_q(y_j) = 1$. The collection's topical profile is defined by,

$$P(y_j|C_i) = \frac{1}{|C_i^{\text{sampled}}|} \sum_{d \in C_i^{\text{sampled}}} \frac{P(y_j|d)}{\sum_{y_k \in \mathcal{Y}} P(y_k|d)}.$$

The similarity between the query and collection topical profiles is given by the Bhattacharya distance between these two distributions [3],

$$\mathcal{B}(q, C_i) = \sum_{y_k \in \mathcal{Y}} \sqrt{P(y_k|q) \times P(y_k|C_i)}.$$

6.3 Evaluation

We are interested in the quality of document rankings produced by selecting only a few collections and combining their documents into a single ranked list. For this reason, we evaluate in terms of precision at different cut-off points, $\mathcal{P}@\{5, 10, 30\}$, when selecting between 1-5 collections. To focus evaluation on resource selection rather than results merging, we assume access to a function that provides the score that a centralized retrieval would have provided for every document retrieved. Given a set of collections selected, we combine their documents into a single ranked list according to each document's centralized retrieval score.

We evaluate on TREC queries 701-850, used in the ad-hoc retrieval task of the Terabyte Track from 2004, 2005, and 2006. Recall that we are missing about 10% of the GOV2 collection in our testbeds, corresponding to those documents in GOV2 not originating from the 1,000 largest hosts. However, all queries had at least one relevant document in our

subset of GOV2, except query 703, which has no relevant documents in the full GOV2 collection.

All approaches were evaluated on all three testbeds under two conditions: sampling 1,000 documents and sampling 300 documents from each collection. We denote these 6 experimental conditions as gov2.1000.1000, gov2.1000.300, gov2.250.1000, gov2.250.300, gov2.30.1000, and gov2.30.300. Our motivation is to investigate the effect of sampled set size on resource selection performance across the classification approach and single-evidence baselines.

7. EXPERIMENTAL RESULTS

We evaluate resource selection based on the quality of the merged retrieval when selecting between 1-5 collections. Results are presented based on $\mathcal{P}@\{5, 10, 30\}$. Table 1 shows results across our three testbeds when sampling 1,000 documents from each collection. Table 2 shows results when sampling 300 documents from each collection. In addition, to evaluate the overall performance of federated search, we present results from centralized retrieval (denoted as "full"), from a single index of all n collections combined.

The classification-based approach either significantly outperforms or is statistically indistinguishable from the *best* single-evidence baseline in all cases. In the gov2.1000.1000 condition, the GAVG and ReDDE.top baselines perform at the same level as the classification approach. We investigate how this experimental condition favors these methods in the next section.

From the performance of our single-evidence baselines, we notice two trends. First, all baselines that derive evidence from sampled documents (i.e., CORI, GAVG, ReDDE.top, and ReDDE) perform better when sampling $1,000$ vs. 300 documents from each collection. This shows that these methods are sensitive to the sampled set size. They perform better with more evidence, which is consistent with previous evaluations [20]. Second, their relative performance varies across experimental conditions. In the gov2.1000.1000 condition, GAVG and ReDDE.top clearly outperform CATS and CLICK in all cases. This is not true in the gov2.30.300 condition. When $k = 1$, in the gov2.30.300 condition, CATS and CLICK both outperform GAVG and ReDDE.top. These approaches derive evidence from different sources. GAVG and ReDDE.top derive evidence exclusively from sampled documents. CATS derives evidence from the topical similarity between the query and the collection. CLICK derives evidence from click-through data. Different types of evidence was particularly useful under different conditions.

Two results support the hypothesis that full-dataset retrievals can be used to harvest data for training a machine learned resource selection method. First, a full-dataset retrieval outperforms all methods, including the classification approach, in all cases. Second, the classification approach, trained on data harvested from full-dataset retrievals, performs at same level or better than the best single-evidence baseline in all cases.

Finally, the fact that a full-dataset retrieval outperforms all methods in all cases indicates that there is room for improvement. We may more closely approximate the performance of a full-dataset retrieval by integrating new sources of evidence into the classification approach. The performance gap between full-dataset and federated retrieval is larger than that observed in some prior work. This may be a product of our three testbeds. Standard testbeds frequently used in prior work contain about 100 collections, with no collection containing more than 1M documents.

7.1 Representation Quality

ReDDE.top and GAVG, which derive evidence from sampled documents, perform well in the gov2.1000.1000 experimental condition. In this condition, 1,000 documents were sampled from every collection. As previously mentioned, 72% of collections in the gov2.1000 testbed have fewer than 10,000 documents and 44% have fewer than 5,000 documents. This means that a sample set of 1,000 documents constitutes at least 20% of the full collection for about half the collections in gov2.1000. In other words, in this condition, ReDDE.top and GAVG have access to fairly complete representations for about half the collections. Furthermore, we would expect these methods to do well if these smaller collections frequently contain relevant documents. To examine this, we binned collections by their number of documents and determined, for each bin, the number of times a collection from the bin contained at least 10 documents relevant to a test query. These histograms are shown in Figure 2. In the gov2.1000 testbed, the smallest collections, with 1,000-10,000 documents, most often contain at least 10 documents relevant to a test query. In contrast, in the gov2.250 and gov2.30 testbeds, the collections that most often contain at least 10 documents relevant to a test query have more than 100,000 documents. Therefore, we can conclude that in the gov2.1000.1000 condition, corpus-based single-evidence baselines such as ReDDE.top and GAVG benefit from having fairly complete representations (i.e. large sampled sets relative to the collection size) for collections containing many relevant documents. In the other conditions, we see a more clear benefit from integrating multiple sources of evidence.

7.2 Feature Ablation Studies

The classification approach integrates different types of evidence as input features. In this section, we conduct a set of feature ablation studies to test the contribution of evidence integration to the classification approach's performance. We focus on experimental condition gov2.1000.1000 and gov2.30.300. Our motivation is to verify that the classification approach is capable of focusing on the most reliable features under different experimental conditions. Based on the analysis from Section 7.1, in the gov2.1000.1000 condition, we expect the classification approach to focus on evidence derived from sampled documents (i.e., CORI, GAVG, and ReDDE.top features). In the gov2.30.300 condition, we expect it to focus on other types of evidence. We individually omitted each feature type (CORI, GAVG, ReDDE.top, CATS, and CLICK) and measure its contribution to performance based on the classifier's percent decrease in precision. Significance, again, is tested using a paired t-test on queries. Results are presented in Table 3. These results confirm our hypothesis. In the gov2.1000.1000 condition, in the majority of cases, omitting ReDDE.top features leads to a significant drop in performance. This is because in the gov2.1000.1000 condition, ReDDE.top has access to fairly complete representations for those collections with relevant content. On the other hand, in the gov2.30.300 condition, CLICK features are more predictive, particularly in terms of $\mathcal{P}@30$. This shows that the classification approach is capable on focusing on the most reliable features depending on the condition. Also, although CORI, GAVG, and ReDDE.top

features derive evidence from the same source (i.e., sampled documents), they model different phenomena. Our results show that they do not contribute equally to performance. This further motivates a feature integration approach, even when the evidence is derived from the same source.

8. CONCLUSION

We evaluated a classification approach to resource selection against a number of single-evidence baselines, including three existing resource selection methods that have produced good results in previous evaluations. Evaluation was done across six experimental conditions, varying the number of target collections and the number of documents sampled from each. The classification approach performed either at the same level or significantly better than all single-evidence baselines in all cases.

Most existing approaches to resource selection derive evidence from collection content. Often, the content in the collection is represented using sampled documents. Our evaluation shows that these methods perform better when they have access to fairly complete representations. Their performance deteriorates, however, when most collections are large and sample sets are small. Our classification-based approach combines these approaches as input features along with features that capitalize on the query-collection topic similarity and click-through information. The end result is a method that is more robust. We show that when collection representation quality is poor, the classifier learns to focus on more reliable sources of evidence from training data.

We also show that full-dataset retrievals, which merge content from every collection, can be used to produce data to train a machine learned approach. More training examples can be produced as long as there is (offline) access to full-dataset retrievals. This training procedure may be particularly valuable in a dynamic environment where collection content is continually updated. A new model can be easily trained using a new set of full-dataset retrievals.

In this work, in order to separate results merging performance from resource selection evaluation, full-dataset retrievals were produced by issuing queries to a centralized index of all collection content. In some federated search environments, it may not be possible to combine collections in a single index. Future research may consider generating training data using a merging algorithm that does not assume access to a single index of all collection content.

9. ACKNOWLEDGMENTS

This work was supported in part by the NSF grants IIS-0841275 and IIS-0534345 and a generous gift from Yahoo!. Any opinions, findings, conclusions, and recommendations expressed in this paper are the authors' and do not necessarily reflect those of the sponsors.

10. REFERENCES

[1] J. Arguello, F. Diaz, J. Callan, and J.-F. Crespo. Sources of evidence for vertical selection. In *SIGIR 2009*, pages 315–322. ACM, 2009.

[2] S. M. Beitzel, E. C. Jensen, O. Frieder, D. D. Lewis, A. Chowdhury, and A. Kolcz. Improving automatic query classification via semi-supervised learning. In *ICDM 2005*, pages 42–49. IEEE, 2005.

[3] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by probability distributions. *Bull. Calcutta Math. Soc.*, 35:99 – 109, 1943.

[4] J. Callan and M. Connell. Query-based sampling of text databases. In *TOIS*. ACM, 2001.

[5] J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *SIGIR 1995*, pages 21–28. ACM, 1995.

[6] F. Diaz. Integration of news content into web results. In *WSDM 2009*, pages 182–191. ACM, 2009.

[7] F. Diaz and J. Arguello. Adaptation of offline vertical selection predictions in the presence of user feedback. In *SIGIR 2009*, pages 323–330. ACM, 2009.

[8] C. T. Fallen and G. B. Newby. Partitioning the gov2 corpus by internet domain name: A result-set merging experiment. In *TREC 2006*, 2006.

[9] L. Gravano, H. Garcia-molina, and A. Tomasic. Gloss: Text-source discovery over the internet. *TOIS*, 24:229–264, 1999.

[10] P. G. Ipeirotis and L. Gravano. Distributed search over the hidden web: hierarchical database sampling and selection. In *VLDB 2002*, pages 394–405. VLDB Endowment, 2002.

[11] H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

[12] X. Li, Y.-Y. Wang, and A. Acero. Learning query intent from regularized click graphs. In *SIGIR 2008*, pages 339–346. ACM, 2008.

[13] Y. Li, Z. Zheng, and H. K. Dai. Kdd cup-2005 report: facing a great challenge. *SIGKDD Explor. Newsl.*, 7(2):91–99, 2005.

[14] D. Metzler. A markov random field model for term dependencies. In *SIGIR 2005*, pages 472–479. ACM Press, 2005.

[15] M. F. Porter. An algorithm for suffix stripping. pages 313–316, 1997.

[16] J. Seo and B. W. Croft. Blog site search using resource selection. In *CIKM 2008*, pages 1053–1062. ACM, 2008.

[17] D. Shen, R. Pan, J.-T. Sun, J. J. Pan, K. Wu, J. Yin, and Q. Yang. Q2c@ust: our winning solution to query classification in kddcup 2005. *SIGKDD Explor. Newsl.*, 7(2):100–110, 2005.

[18] D. Shen, J.-T. Sun, Q. Yang, and Z. Chen. Building bridges for web query classification. In *SIGIR 2006*, pages 131–138. ACM, 2006.

[19] M. Shokouhi. Central rank based collection selection in uncooperative distributed information retrieval. In *ECIR 2007*, pages 160–172. ACM, 2007.

[20] M. Shokouhi, F. Scholer, and J. Zobel. Sample sizes for query probing in uncooperative distributed information retrieval. In *APWeb 2006*, pages 63–75. Springer, 2006.

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

[22] L. Si and J. Callan. Unified utility maximization framework for resource selection. In *CIKM 2004*, pages 32–41. ACM, 2004.

[23] L. Si, R. Jin, J. Callan, and P. Ogilvie. A language modeling framework for resource selection and results merging. In *CIKM 2002*, pages 391–397. ACM, 2002.

[24] P. Thomas and M. Shokouhi. Sushi: Scoring scaled samples for server selection. In *SIGIR 2009*. ACM, 2009.

[25] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang. Query clustering using content words and user feedback. In *SIGIR 2001*, pages 442–443. ACM, 2001.

[26] J. Xu and W. B. Croft. Cluster-based language models for distributed retrieval. In *SIGIR 1999*, pages 254–261. ACM, 1999.

gov2.1000.1000

P@5

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.569	0.224	**0.405**	0.360	0.166	0.192	0.183	0.392 (-3.31%)
2	0.569	0.315	0.446	**0.447**	0.275	0.256	0.239	0.436 (-2.40%)
3	0.569	0.372	0.479	**0.489**	0.336	0.302	0.277	0.482 (-1.37%)
4	0.569	0.405	0.483	**0.506**	0.380	0.321	0.322	**0.506 (0.00%)**
5	0.569	0.417	0.495	**0.529**	0.395	0.336	0.337	0.510 (-3.55%)

P@10

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.534	0.188	0.331	0.321	0.150	0.152	0.147	**0.355 (7.30%)**
2	0.534	0.264	0.390	0.394	0.248	0.215	0.194	**0.399 (1.19%)**
3	0.534	0.323	0.423	0.436	0.302	0.261	0.228	**0.446 (2.47%)**
4	0.534	0.359	0.438	0.457	0.344	0.285	0.270	**0.458 (0.15%)**
5	0.534	0.380	0.442	**0.484**	0.364	0.302	0.281	0.468 (-3.33%)

P@30

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.452	0.113	0.201	0.206	0.102	0.095	0.091	**0.224 (8.68%)**
2	0.452	0.167	0.266	0.268	0.168	0.139	0.124	**0.281 (4.59%)**
3	0.452	0.217	0.305	0.312	0.206	0.170	0.152	**0.319 (2.51%)**
4	0.452	0.247	0.319	0.337	0.248	0.194	0.185	**0.339 (0.53%)**
5	0.452	0.266	0.325	**0.362**	0.275	0.205	0.195	0.352 (-2.60%)

gov2.250.1000

P@5

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.569	0.137	0.294	0.326	0.238	0.174	0.220	**0.419 (28.40%)**‡
2	0.569	0.228	0.328	0.408	0.360	0.242	0.303	**0.494 (21.05%)**‡
3	0.569	0.291	0.360	0.432	0.417	0.272	0.340	**0.497 (14.91%)**‡
4	0.569	0.323	0.374	0.475	0.483	0.313	0.364	**0.505 (4.44%)**
5	0.569	0.357	0.389	0.489	0.503	0.345	0.388	**0.515 (2.40%)**

P@10

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.534	0.105	0.248	0.283	0.209	0.142	0.188	**0.371 (31.35%)**‡
2	0.534	0.186	0.293	0.363	0.311	0.201	0.262	**0.452 (24.40%)**‡
3	0.534	0.248	0.330	0.394	0.372	0.229	0.291	**0.460 (16.70%)**‡
4	0.534	0.282	0.338	0.432	0.430	0.266	0.308	**0.477 (10.58%)**‡
5	0.534	0.293	0.350	0.438	0.457	0.297	0.334	**0.487 (6.46%)**†

P@30

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.452	0.068	0.158	0.197	0.143	0.090	0.130	**0.265 (34.13%)**‡
2	0.452	0.124	0.196	0.272	0.230	0.132	0.182	**0.343 (26.19%)**‡
3	0.452	0.168	0.233	0.309	0.283	0.151	0.213	**0.359 (16.05%)**‡
4	0.452	0.204	0.245	0.337	0.331	0.187	0.227	**0.372 (10.22%)**‡
5	0.452	0.226	0.262	0.344	0.353	0.208	0.246	**0.382 (8.38%)**‡

gov2.30.1000

P@5

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.569	0.281	0.302	0.322	0.295	0.323	0.298	**0.370 (14.52%)**
2	0.569	0.380	0.403	0.419	0.428	0.384	0.374	**0.447 (4.39%)**
3	0.569	0.434	0.446	0.456	0.447	0.427	0.421	**0.487 (6.76%)**
4	0.569	0.462	0.468	0.487	0.472	0.451	0.454	**0.499 (2.48%)**
5	0.569	0.474	0.472	0.503	0.491	0.482	0.460	**0.507 (0.80%)**

P@10

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.534	0.246	0.264	0.269	0.246	0.280	0.255	**0.318 (13.67%)**
2	0.534	0.332	0.348	0.361	0.368	0.340	0.335	**0.393 (6.56%)**
3	0.534	0.391	0.387	0.403	0.392	0.384	0.374	**0.438 (8.49%)**†
4	0.534	0.426	0.415	0.442	0.415	0.407	0.413	**0.461 (4.41%)**
5	0.534	0.445	0.429	0.462	0.444	0.433	0.423	**0.471 (2.03%)**

P@30

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.452	0.181	0.188	0.185	0.167	0.195	0.176	**0.220 (12.87%)**
2	0.452	0.253	0.262	0.261	0.269	0.267	0.241	**0.304 (13.32%)**†
3	0.452	0.309	0.294	0.304	0.304	0.302	0.280	**0.346 (11.71%)**†
4	0.452	0.339	0.326	0.337	0.328	0.313	0.309	**0.361 (6.47%)**
5	0.452	0.353	0.341	0.358	0.345	0.334	0.320	**0.377 (5.25%)**

Table 1: Results for experimental conditions gov2.1000.1000, gov2.250.1000, and gov2.30.1000. Percent improvement is with respect to the best single-evidence baseline. Statistical significance is with respect to all single-evidence baselines. Significance, using a paired t-test on queries, is denoted with a † at the $p < 0.05$ level and a ‡ at the $p < 0.005$ level.

gov2.1000.300

P@5

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.569	0.209	0.323	0.303	0.125	0.200	0.183	**0.383 (18.26%)**
2	0.569	0.306	0.358	0.407	0.221	0.263	0.239	**0.427 (4.95%)**
3	0.569	0.340	0.399	0.450	0.283	0.301	0.277	**0.463 (2.99%)**
4	0.569	0.370	0.427	0.466	0.313	0.313	0.322	**0.482 (3.46%)**
5	0.569	0.381	0.440	0.478	0.350	0.333	0.337	**0.490 (2.53%)**

P@10

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.534	0.174	0.277	0.270	0.110	0.166	0.147	**0.332 (19.90%)**†
2	0.534	0.260	0.317	0.358	0.191	0.224	0.194	**0.392 (9.57%)**
3	0.534	0.293	0.361	0.399	0.253	0.269	0.228	**0.432 (8.07%)**
4	0.534	0.321	0.381	0.419	0.283	0.276	0.270	**0.444 (5.93%)**
5	0.534	0.339	0.399	0.433	0.321	0.297	0.281	**0.456 (5.27%)**

P@10

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.452	0.097	0.174	0.171	0.074	0.101	0.091	**0.218 (25.65%)**†
2	0.452	0.162	0.216	0.245	0.126	0.141	0.124	**0.270 (10.52%)**
3	0.452	0.191	0.249	0.284	0.172	0.176	0.152	**0.304 (7.10%)**
4	0.452	0.211	0.267	0.304	0.197	0.185	0.185	**0.324 (6.47%)**
5	0.452	0.230	0.284	0.321	0.224	0.208	0.195	**0.341 (6.05%)**

gov2.250.300

P@5

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.569	0.125	0.212	0.274	0.228	0.111	0.220	**0.391 (42.65%)**‡
2	0.569	0.184	0.267	0.353	0.333	0.176	0.303	**0.472 (33.84%)**‡
3	0.569	0.246	0.286	0.407	0.391	0.232	0.340	**0.494 (21.45%)**‡
4	0.569	0.267	0.306	0.434	0.428	0.268	0.364	**0.498 (14.86%)**‡
5	0.569	0.290	0.319	0.462	0.447	0.279	0.388	**0.518 (12.21%)**‡

P@10

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.534	0.096	0.178	0.228	0.186	0.089	0.188	**0.342 (49.71%)**‡
2	0.534	0.161	0.234	0.304	0.270	0.156	0.262	**0.427 (40.40%)**‡
3	0.534	0.218	0.249	0.366	0.350	0.195	0.291	**0.450 (22.94%)**‡
4	0.534	0.236	0.273	0.399	0.387	0.218	0.308	**0.456 (14.31%)**‡
5	0.534	0.258	0.283	0.417	0.409	0.233	0.334	**0.476 (13.99%)**‡

P@30

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.452	0.057	0.115	0.148	0.133	0.055	0.130	**0.241 (62.60%)**‡
2	0.452	0.109	0.161	0.209	0.190	0.107	0.182	**0.315 (51.13%)**‡
3	0.452	0.149	0.182	0.253	0.251	0.138	0.213	**0.333 (31.42%)**‡
4	0.452	0.173	0.200	0.292	0.277	0.154	0.227	**0.350 (19.77%)**‡
5	0.452	0.187	0.210	0.314	0.302	0.166	0.246	**0.362 (15.34%)**‡

gov2.30.300

P@5

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.569	0.224	0.266	0.251	0.231	0.282	0.298	**0.374 (25.68%)**‡
2	0.569	0.317	0.322	0.350	0.342	0.353	0.374	**0.450 (20.07%)**‡
3	0.569	0.409	0.376	0.391	0.400	0.412	0.421	**0.493 (16.88%)**‡
4	0.569	0.446	0.424	0.403	0.413	0.444	0.454	**0.487 (7.40%)**
5	0.569	0.467	0.436	0.443	0.442	0.464	0.460	**0.509 (8.91%)**

P@10

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.534	0.194	0.222	0.206	0.178	0.238	0.255	**0.321 (25.79%)**‡
2	0.534	0.287	0.271	0.313	0.292	0.299	0.335	**0.402 (20.04%)**‡
3	0.534	0.361	0.327	0.353	0.338	0.352	0.374	**0.442 (18.13%)**‡
4	0.534	0.403	0.363	0.370	0.376	0.394	0.413	**0.457 (10.55%)**†
5	0.534	0.432	0.385	0.413	0.401	0.428	0.423	**0.479 (10.71%)**†

P@30

k	full	cori	gavg	redde.top	redde	cats	click	classification
1	0.452	0.128	0.153	0.143	0.118	0.153	0.176	**0.223 (26.75%)**‡
2	0.452	0.206	0.198	0.227	0.200	0.218	0.241	**0.312 (29.38%)**‡
3	0.452	0.263	0.243	0.265	0.246	0.271	0.280	**0.347 (24.06%)**‡
4	0.452	0.308	0.282	0.291	0.282	0.300	0.309	**0.367 (18.51%)**‡
5	0.452	0.336	0.295	0.334	0.317	0.322	0.320	**0.390 (15.97%)**‡

Table 2: Results for experimental conditions gov2.1000.300, gov2.250.300, and gov2.30.300. Percent improvement is with respect to the best single-evidence baseline. Statistical significance is with respect to all single-evidence baselines. Significance, using a paired t-test on queries, is denoted with a † at the $p < 0.05$ level and a ‡ at the $p < 0.005$ level.

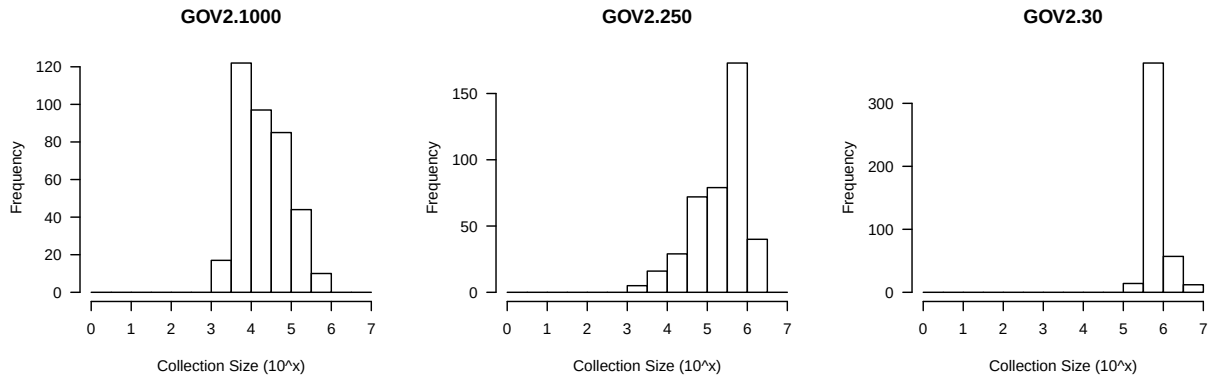

Figure 2: Number of instances in which a collection of a given size (bin) contributes at least 10 relevant documents to a test query.

gov2.1000.1000

P@10

k	all.features	no.cori	no.gavg	no.redde.top	no.cats	no.click
1	0.355	0.355 (0.00%)	0.357 (0.57%)	**0.331 (-6.81%)**	0.355 (0.00%)	0.354 (0.19%)
2	0.399	0.399 (0.00%)	0.393 (-1.52%)	**0.383 (-4.04%)**	0.385 (-3.37%)	0.401 (-0.51%)
3	0.446	0.446 (-0.15%)	0.436 (-2.26%)	**0.401 (-10.23%)** ‡	0.436 (-2.41%)	0.438 (-1.95%)
4	0.458	0.456 (-0.29%)	0.442 (-3.52%) †	**0.425 (-7.18%)** †	0.450 (-1.76%)	0.449 (-1.91%)
5	0.468	0.467 (-0.14%)	0.454 (-3.01%) †	**0.431 (-7.89%)** †	0.466 (-0.43%)	0.456 (-2.58%)

P@30

k	all.features	no.cori	no.gavg	no.redde.top	no.cats	no.click
1	0.224	0.224 (0.00%)	0.227 (1.40%)	**0.213 (-5.19%)**	0.229 (2.20%)	0.225 (-0.20%)
2	0.281	0.281 (0.16%)	0.274 (-2.39%)	**0.266 (-5.02%)**	0.271 (-3.51%)	0.277 (-1.44%)
3	0.319	0.317 (-0.77%)	0.311 (-2.59%)	**0.292 (-8.61%)** †	0.312 (-2.24%)	0.313 (-2.10%)
4	0.339	0.338 (-0.20%)	0.330 (-2.70%)	**0.319 (-5.80%)** †	0.331 (-2.38%)	0.336 (-0.79%)
5	0.352	0.350 (-0.51%)	0.344 (-2.35%)	**0.331 (-5.97%)** †	0.347 (-1.52%)	0.344 (-2.35%) †

gov2.30.300

P@10

k	all.features	no.cori	no.gavg	no.redde.top	no.cats	no.click
1	0.321	0.321 (0.21%)	0.319 (-0.63%)	0.305 (-4.81%)	0.324 (1.05%)	**0.279 (-13.18%)**
2	0.402	0.394 (-2.00%)	0.390 (-3.01%)	0.392 (-2.50%)	0.388 (-3.51%)	**0.379 (-5.84%)**
3	0.442	0.438 (-0.91%)	0.428 (-3.04%)	**0.423 (-4.26%)**	0.431 (-2.43%)	0.435 (-1.52%)
4	0.457	**0.449 (-1.76%)**	0.455 (-0.44%)	0.469 (2.64%)	0.450 (-1.62%)	0.456 (-0.29%)
5	0.479	0.477 (-0.42%)	0.472 (-1.40%)	0.474 (-0.84%)	0.480 (0.28%)	**0.465 (-2.81%)**

P@30

k	all.features	no.cori	no.gavg	no.redde.top	no.cats	no.click
1	0.223	0.224 (0.70%)	0.219 (-1.41%)	0.206 (-7.34%)	0.228 (2.41%)	**0.191 (-14.37%)**
2	0.312	0.309 (-1.22%)	0.300 (-4.08%) †	0.301 (-3.51%)	0.295 (-5.52%)	**0.283 (-9.46%)** †
3	0.347	0.338 (-2.51%)	0.333 (-3.99%)	0.335 (-3.54%)	0.330 (-4.90%) †	**0.319 (-8.12%)** †
4	0.367	0.356 (-2.93%)	0.364 (-0.79%)	0.370 (0.98%)	0.349 (-4.70%)	**0.349 (-4.82%)**
5	0.390	0.380 (-2.52%)	0.387 (-0.63%)	0.383 (-1.72%)	0.381 (-2.29%)	**0.372 (-4.65%)**

Table 3: Feature type ablation study. A significant drop in performance, using a paired t-test on queries, is denoted with a † at the $p < 0.05$ level and a ‡ at the $p < 0.005$ level.