Sources of Evidence for Vertical Selection

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

Fernando Díaz
Yahoo! Labs Montréal
1000 Rue de la Gauchetière
Suite 2400
Montréal, QC H3M4W5
díazf@yahoo-inc.com

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

Jean-François Crespo
Yahoo! Labs Montréal
1000 Rue de la Gauchetière
Suite 2400
Montréal, QC H3M4W5
jfcrespo@yahoo-inc.com

ABSTRACT

Web search providers often include search services for domain-specific subcollections, called *verticals*, such as news, images, videos, job postings, company summaries, and artist profiles. We address the problem of *vertical selection*, predicting relevant verticals (if any) for queries issued to the search engine's main web search page. In contrast to prior query classification and resource selection tasks, vertical selection is associated with unique resources that can inform the classification decision. We focus on three sources of evidence: (1) the query string, from which features are derived independent of external resources, (2) logs of queries previously issued directly to the vertical, and (3) corpora representative of vertical content. We focus on 18 different verticals, which differ in terms of semantics, media type, size, and level of query traffic. We compare our method to prior work in federated search and retrieval effectiveness prediction. An in-depth error analysis reveals unique challenges across different verticals and provides insight into vertical selection for future work.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

vertical selection, distributed information retrieval, resource selection, aggregated search, query classification

1. INTRODUCTION

In recent years, major search engines have extended their services to include search on specialized subcollections or *verticals* focused on specific domains (e.g., news, travel, and local search) or media types (e.g., images and video). There are currently two ways through which a user can access vertical content. If the user suspects that relevant content exists in a vertical, she may issue the query directly to a vertical search engine. On the other hand, if the user is unaware of a relevant vertical or prefers a portal interface, she may issue the query directly to a portal search engine. To address this, search engines can include summaries of relevant vertical results in web results, as shown in Figure 1. In the research community, this is referred to as *aggregated search* and has been implemented by many major search engines [12].

Vertical selection is the task of selecting the relevant verticals, if any, in response to a user's query. We focus on *single vertical selection*, defined as the task of predicting a single relevant vertical, if any. Figure 1 exemplifies a common action associated with single vertical selection—embedding a short summary of the relevant vertical's results above the first web result. We are conservative in predicting at most a single vertical, as some queries have multiple relevant verticals. However, as we will see later, most queries in our evaluation set were assigned zero or one relevant vertical by human annotators.

Vertical selection is related to the task of *resource selection* in *federated search* or *distributed information retrieval.* Resource selection is the task of deciding which collections to search given a user's query [4]. Similar to resource selection, vertical selection can be informed by the content of each vertical. However, vertical selection has a few distinguishing properties. First, verticals specialize on identifiable domains and types of media. This enables users to possibly express interest in vertical content explicitly, using key-

[*]work done while at Yahoo! Labs Montréal

SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.
Copyright 2009 ACM 978-1-60558-483-6/09/07 ...$5.00.

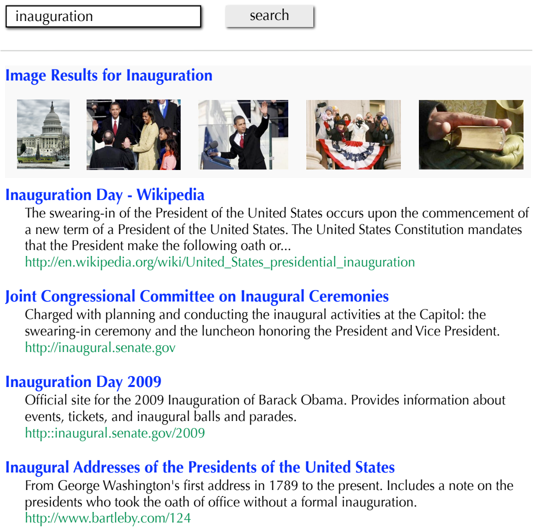

Image Results for Inauguration

Inauguration Day - Wikipedia

The swearing-in of the President of the United States occurs upon the commencement of a new term of a President of the United States. The United States Constitution mandates that the President make the following oath or...

http://en.wikipedia.org/wiki/United_States_presidential_inauguration

Joint Congressional Committee on Inaugural Ceremonies

Charged with planning and conducting the inaugural activities at the Capitol: the swearing-in ceremony and the luncheon honoring the President and Vice President.

http://inaugural.senate.gov

Inauguration Day 2009

Official site for the 2009 Inauguration of Barack Obama. Provides information about events, tickets, and inaugural balls and parades.

http::inaugural.senate.gov/2009

Inaugural Addresses of the Presidents of the United States

From George Washington's first address in 1789 to the present. Includes a note on the presidents who took the oath of office without a formal inauguration.

http://www.bartleby.com/124

Figure 1: A vertical selection system determines that the *images* vertical is relevant to query "inauguration".

words such as "news" for the *news* vertical or "pictures" for the *images* vertical. Therefore, a potentially useful source of evidence for vertical selection is the query string itself, independent of any other resource. Second, some verticals have a search interface through which users directly search for vertical content. Because a vertical selection system and its target verticals are operated by a common entity (e.g., search engine company), we assume access to vertical query-logs. Third, users do not always seek vertical content, but may prefer the default web results instead. In contrast to resource selection, where a resource is always selected in order to retrieve documents, in vertical selection we must decide when to not predict any vertical relevant.

We investigate a classification-based approach to vertical selection and exploit three feature types: (1) query string features, (2) corpus features, derived from vertical representative corpora, and (3) query-log features, derived from vertical query-logs. Corpus and query-log features enrich the query representation beyond the query string and focus on two potentially complementary sources of evidence—corpus features relate to content production (i.e., content in the vertical) and query-log features relate to content demand (i.e., content sought by users). With respect to corpus features, we make use of and compare against prior work in resource selection for federated search (i.e., scoring a collection by its expected number of relevant documents) and retrieval effectiveness prediction (i.e., scoring a collection by the predicted quality of its retrieval). We evaluate corpus features on two types of collections: collections of vertical-sampled documents and surrogate collections representative of verticals constructed by sampling a non-vertical resource, the Wikipedia. [1] We evaluate several simple baselines, each focused on a single source of evidence and a supervised approach that combines our three feature types. An error analysis shows the contribution of each feature type and reveals unique challenges in vertical selection.

2. RELATED WORK

If we consider verticals as external collections, we may view vertical selection analogous to resource selection in federated search. Most prior approaches to resource selection derive evidence solely from the target collections either directly or indirectly, using a sampling of documents as proxy for the collection [6, 17, 19, 20, 9]. Approaches such as CORI [6], CVV [20], and KL-divergence [19] treat collections (or their sampled documents) as "large documents" and adapt document scoring techniques to scoring collections. Because these techniques make no distinction between documents, they do not model the number of relevant documents in a collection [16]. Approaches such as GlOSS (and its variations) [9] as well as ReDDE [17] more explicitly model the distribution of relevant documents across resources. ReDDE issues the query to an index of documents sampled from the target collections and scores each collection proportional to the number of top-ranked documents originating from it, taking into account the difference between the size of the original collection and its sample size.

Some verticals are genre-specific. Therefore, some prior work in query-classification into topical categories is relevant to vertical selection [13, 14, 2, 1, 10]. Because queries are terse, many query-classification approaches augment the query with features beyond the query string, possibly derived from query-logs or corpora of documents associated with the target classes. Bietzel *et al.* use a large (unlabeled) query-log and a technique known as *selectional preference*—the query "interest rates" belongs to target category *finance* because "interest" and "rates" are distributionally similar to the term "finance" [1, 2]. Shen *et al.* [13] and other participants of the KDD 2005 Cup [11] use corpus-based evidence. These techniques resemble ReDDE in that the query is issued to an index of documents associated with the target categories and the query's membership to a category is proportional to the number of top-ranked documents associated with the category. In later work, Shen *et al.* derive a soft mapping from documents to target categories using term similarity [14]. The category representation is augmented with related terms using pseudo-relevance feedback.

There is some prior work on vertical selection. Li *et al.* focus on the *shopping* and *jobs* verticals [10]. They focus on query lexical features and use a query-click graph to propagate category labels to unlabeled queries. Our work differs from that of Li *el al.* in that we enrich the query representation beyond query string features, focus on more verticals, and, by formulating the task as single vertical selection, we examine vertical contention resolution rather than evaluate on each vertical independently. Diaz investigates vertical selection with respect to the *news* vertical [8]. Diaz focuses on features derived from the news collection and from web and vertical query-logs and incorporates click-feedback into the model. We extend the work of Diaz by exploring more features, focusing on more verticals, and evaluating on human relevance judgements rather than clicks.

3. PROBLEM DEFINITION

Throughout the paper, we will use the following notation.

$\mathcal{V}$	set of all verticals
$\mathcal{Q}$	set of all queries
$\mathcal{V}_q$	set of verticals relevant to query q
$\tilde{v}_q$	the single vertical *predicted* relevant to query q

[1] http://www.wikipedia.org

vertical	retrievable items
autos	car reviews, product descriptions
directory	web page directory nodes
finance	financial data and corporate information
games	hosted online games
health	health-related articles
images	online images
jobs	job listings
local	business listings
maps	maps and directions
movies	movie show times
music	musician profiles
news	news articles
reference	encyclopedic entries
shopping	product reviews and listings
sports	sports articles, scores, and statistics
travel	travel and accommodation reviews and listings
tv	television listings
video	online videos

Table 1: Vertical descriptions.

We define single vertical selection as the following problem. Given query q, the objective is to predict a single relevant vertical, $\tilde{v}_q \in \mathcal{V}_q$, if one exists, and to predict the "no relevant vertical" class, $\tilde{v}_q = \emptyset$, otherwise. Formally, we want to maximize single vertical precision,

$$\mathcal{P} = \frac{1}{|\mathcal{Q}|}\left(\sum_{q \in \mathcal{Q}|\mathcal{V}_q \neq \emptyset} \mathcal{I}(\tilde{v}_q \in \mathcal{V}_q) + \sum_{q \in \mathcal{Q}|\mathcal{V}_q = \emptyset} \mathcal{I}(\tilde{v}_q = \emptyset)\right), \quad (1)$$

where $\mathcal{I}$ is the indicator function. The first term is the number of queries for which a relevant vertical was correctly predicted. The second term is the number of queries for which the "no relevant vertical" class was correctly predicted. We investigated the 18 verticals described in Table 1.

4. FEATURES

We investigated three sources of evidence for vertical selection: the query string, vertical-representative corpora (not necessarily composed of vertical documents), and queries previously issued to the vertical.

4.1 Query String Features

Perhaps the lowest effort approach to vertical selection operates on the query string alone, disregarding hits on vertical collections or previous queries issued directly to the vertical. Query string features aim to capitalize on key phrases used in explicit requests for vertical content (e.g. "inauguration *pictures*") and a possible correlation between named entity types and a vertical (e.g., *music* vertical queries may mention a musician). We define two types of query string features: rule-based vertical triggers and geographic features.

4.1.1 Rule-based vertical triggers

Rule-based vertical triggers are based on a set of 45 classes aimed to characterize the query's vertical intent (e.g.,*local phone*, *product*, *person*, *weather*, *movies*, *driving direction*, *music artist*). Some of these 45 triggers map conceptually one-to-one to a target vertical (e.g., *movies* → *movies*, *autos* → *autos*). Others map many-to-one (e.g., {*sports players*, *sports*} → *sports*, {*product review*, *product*} → *shopping*). Others do not map directly to a vertical, but may provide (positive or negative) evidence in a supervised classification framework (e.g., *patent*, *events*, *weather*). Each trigger class is associated with hand-crafted rules using regular expressions and dictionary lookups. A query may be associated with multiple classes, each triggered if at least one rule in its inventory matches the query.

4.1.2 Geographic features

Geographic features were produced using a rule-based geographic annotation tool that outputs a probability vector for a set of geographic entities possibly appearing in the query. We focused on the following geographic entities: *airport*, *colloquial* (i.e., location information associated with a named entity, such as "North Shore Bank"), *continent*, *country*, *county*, *estate*, *historical county*, *historical state*, *historical town*, *island*, *land feature*, *point of interest* (e.g, Eiffel Tower), *sports team*, *suburb*, *supername* (i.e., a region name, such as Middle East), *town*, and *zip code*. We used the probability of each entity being present in the query as a separate feature. Geographic features are intended to inform classification into verticals whose queries often mention a location name, such as *local*, *travel*, and *maps*.

4.2 Query-Log Features

Query-log features use evidence from the queries previously issued to the vertical, which reflect the topics in the vertical that are of interest to users. For each vertical, we compute the query likelihood given by a unigram language model constructed from the vertical's query-log. Our query-log features (one per vertical) are defined by,

$$\mathrm{QL}_q(\mathcal{V}_i) = \frac{1}{\mathcal{Z}} P(q|\theta^{\mathrm{qlog}}_{\mathcal{V}_i}), \quad (2)$$

where $\theta^{\mathrm{qlog}}_{\mathcal{V}_i}$ is vertical $\mathcal{V}_i$'s query-log language model and $\mathcal{Z} = \sum_{\mathcal{V}_j \in \mathcal{V}} P(q|\theta^{\mathrm{qlog}}_{\mathcal{V}_j})$.

We collected a year's worth of vertical query-logs for the year preceding the gathering of our evaluation query set. In addition, to inform classification into the "no relevant vertical" class, we also collected Web query-logs. Since Web search sees much more traffic than vertical search, we collected only a month's worth of Web query-logs. We used the CMU-Cambridge Language Modeling Toolkit [2] to build a unigram language model from each query-log. Each language model's vocabulary was defined by its most frequent 20000 unigrams and we used Witten-Bell smoothing [18].

Query-log features were evaluated under two conditions: allowing and disallowing zero probabilities from out of vocabulary (OOV) terms. In the first condition, a single OOV query term results in a zero probability from the vertical. In the second condition, $P(\mathrm{OOV}|\theta_{\mathcal{V}_i})$ was estimated proportional to the frequency of terms not in the top 20000 in vertical $\mathcal{V}_i$'s query-log.

Some of our target verticals did not have query-logs predating the collection of our evaluation query set. These included *autos*, *maps*, *sports*, and *tv*.

4.3 Corpus Features

Corpus features are derived from document rankings obtained by issuing the query to different collections. Conducting a retrieval allows us to compare, for example, the number of retrieved documents from different verticals. In practice, issuing a query to all verticals can incur unnecessary query load on the vertical retrieval system. Therefore,

[2] http://svr-www.eng.cam.ac.uk/ prc14/toolkit.html

we construct smaller, representative corpora of vertical content local to the vertical selector.

4.3.1 Constructing Representative Corpora

Before discussing our corpus-based features, we will describe two methods for creating representative corpora: sampling from the vertical and using surrogate corpora.

Direct Sampling from the Vertical.

Query-based sampling [5] is a technique for sampling documents from collections assumed to provide only a query-in-documents-out interface. The most general query-based sampling approach iterates over the following steps. A single-term query is used to retrieve documents from the collection. Then, the collection's content description is updated and a new single-term sampling query is selected from the updated content description. As documents are retrieved, the evolving resource description and, indirectly, new sampling queries are derived from retrieved documents. Shakouhi *et al.* show that using high-frequency query-log queries for sampling can produce more effective resource descriptions than queries derived from the sampled documents [15]. We follow a similar approach. While Shakouhi *et al.* use the same set of queries to sample from every collection, we use queries from vertical query-logs. Sampling with query-log queries has two effects. First, it decouples the sampling query from the sampled documents. Second, the sampled documents are biased towards those more likely to be seen by users. This is important when constructing small samples of large corpora if a significant part of the corpus is not of interest to users.

We used the following procedure to sample documents. First, we collected the top 100 documents returned by running each of the 1000 most frequent non-stopword query-log unigrams as a query. Then, we uniformly sampled at most 25000 documents from the union of these documents. Because we used vertical query-logs to sample vertical documents, verticals without query-logs preceding our evaluation queries also lacked a vertical-sampled collection. The "no relevant vertical" class does not have a vertical collection for sampling. We denote the set of documents sampled from vertical $\mathcal{V}_i$ by $\mathcal{S}_i^{\text{vertical}}$.

Sampling from Wikipedia.

An alternative to sampling directly from the verticals is to sample from an external collection, if documents can be mapped conceptually to verticals. We sampled documents from Wikipedia, making use of Wikipedia categories to map documents to verticals using regular expressions. Each article in Wikipedia belongs to one or more categories. For instance, a sample of documents representative of the *autos* vertical was gathered from articles assigned a Wikipedia category containing any of the terms "Automobile", "Car", and "Vehicle".

We do not claim that our mapping of Wikipedia documents to verticals is optimal. The risk of associating documents from an external collection to a vertical is misrepresenting the vertical's contents. However, sampling from Wikipedia may provide several advantages. First, Wikipedia is rich in text. Our corpus features, discussed next, are dependent on text richness. Documents typical of some verticals (used to represent the vertical in direct vertical sampling), such as *images* and *video*, tend to be text poor. Second, Wikipedia articles have a consistent format, which makes comparing rankings across collections easier. Third, Wikipedia articles are usually semantically coherent and on topic.

For practical reasons, some verticals were not mapped to Wikipedia content. The *directory* vertical and the "no relevant vertical" class are too broad to be sensibly characterized by a set of Wikipedia categories while the *maps* vertical intersects semantically with *local* and *travel*. We denote the set of Wikipedia articles mapped to vertical $\mathcal{V}_i$ by $\mathcal{S}_i^{\text{wiki}}$.

4.3.2 Corpus-Based Features

Retrieval Effectiveness Features.

Predicting retrieval effectiveness is the task of automatically assessing the quality of a retrieval without human relevance judgements. We applied an existing approach to predicting retrieval effectiveness, Clarity [7], to vertical selection. Our motivation is that a collection's predicted retrieval effectiveness with respect to a query may correlate with the collection's relevance to the query. Clarity measures the similarity between the language of the top ranked documents and the language of the collection, estimated using the Kullback-Leibler divergence between the query and collection language model,

$$\text{Clarity}_q(C) = \sum_{w \in V} P(w|\theta_q) \log_2 \frac{P(w|\theta_q)}{P(w|\theta_C)}, \tag{3}$$

where V is the vocabulary of collection C and $P(w|\theta_q)$ and $P(w|\theta_C)$ are the query and collection language models, respectively. The query language model was estimated from the top 100 documents, $\mathcal{R}_{100}$, according to,

$$P(w|\theta_q) \quad = \tfrac{1}{\mathcal{Z}} \textstyle\sum_{d \in \mathcal{R}_{100}} P(w|\theta_d) P(q|\theta_d), \tag{4}$$

where $P(q|\theta_d)$ is the query likelihood score of document d, and $\mathcal{Z} = \sum_{d \in \mathcal{R}_{100}} P(q|\theta_d)$. The Clarity score becomes smaller as the top ranked documents approach a random sample from the collection (i.e., an ineffective retrieval).

We used two sets of Clarity features: one using collections of vertical-sampled documents and one using collections of Wikipedia-sampled documents. The final Clarity score for vertical $\mathcal{V}_i$ is given by,

$$\text{Clarity}^*_q(\mathcal{V}_i) = \frac{1}{\mathcal{Z}^*} \text{Clarity}_q(\mathcal{S}^*_i), \tag{5}$$

where $\mathcal{S}^*_i$ denotes either $\mathcal{S}_i^{\text{vertical}}$, the set of documents sampled from $\mathcal{V}_i$, or $\mathcal{S}_i^{\text{wiki}}$, the set of Wikipedia documents mapped to $\mathcal{V}_i$ and $\mathcal{Z}^* = \sum_{\mathcal{V}_j \in \mathcal{V}} \text{Clarity}_q(\mathcal{S}^*_j)$.

ReDDE Features.

As previously mentioned, in federated search, resource selection is the problem of deciding which collections to search given a query. We adapted an existing approach to resource selection, ReDDE [17], to the task of vertical selection. ReDDE scores a target collection based on its expected number documents relevant to the query. It derives this expectation from a retrieval of a index that combines documents sampled from every target collection. Given this retrieval, ReDDE accumulates a collection's score from its document scores, taking into account the difference between the size of the original collection and sampled set size. As with Clarity features, we generated two sets of ReDDE features: one using vertical-sampled documents and one using Wikipedia-sampled documents. ReDDE scores vertical $\mathcal{V}_i$ according to,

$$\text{ReDDE}^*_q(\mathcal{V}_i) = |\mathcal{V}_i| \sum_{d \in \mathcal{R}_{100}} \mathcal{I}(d \in \mathcal{S}^*_i) P(q|\theta_d) P(d|\mathcal{S}^*_i), \quad (6)$$

where,

$$P(d|\mathcal{S}^*_i) = \frac{1}{|\mathcal{S}^*_i|}. \quad (7)$$

The term $|\mathcal{V}_i|$ is the number of documents in vertical $\mathcal{V}_i$ and $\mathcal{S}^*_i$ denotes either $\mathcal{S}^{\text{vertical}}_i$, the documents sampled directly from $\mathcal{V}_i$, or $\mathcal{S}^{\text{wiki}}_i$, the Wikipedia documents mapped to $\mathcal{V}_i$. We normalize across vertical- and Wikipedia-sampled ReDDE features such that $\sum_{\mathcal{V}_j \in \mathcal{V}} \text{ReDDE}^*_q(\mathcal{V}_j) = 1$.

Soft.ReDDE Features.

ReDDE requires a hard assignment of documents to verticals. When sampling from verticals, this assignment is trivial—a document represents the vertical from which it originates. When sampling from non-vertical collections (e.g., Wikipedia), this assignment is not trivial, and we risk misrepresenting a vertical's contents. We experimented with a novel approach similar to ReDDE. Soft.ReDDE computes a soft membership of a document to a vertical, $\phi(d, \mathcal{V}_i)$, based the correlation between the document language model, θ_d, and vertical language model, $\theta_{\mathcal{V}_i}$, estimated using the vertical's query-log. We used the Bhattacharyya correlation [3], defined by,

$$\mathcal{B}(d, \mathcal{V}_i) = \sum_w \sqrt{P(w|\theta_d) P(w|\theta_{\mathcal{V}_i})}, \quad (8)$$

and normalize across verticals,

$$\phi(d, \mathcal{V}_i) = \frac{\mathcal{B}(d, \mathcal{V}_i)}{\sum_{\mathcal{V}_j \in \mathcal{V}} \mathcal{B}(d, \mathcal{V}_j)}. \quad (9)$$

Soft.ReDDE scores a vertical by the sum of documents scores, $P(q|\theta_d)$, weighted by the document's similarity to the vertical,

$$\text{Soft.ReDDE}_q(\mathcal{V}_i) = \sum_{d \in \mathcal{R}_{100}} \phi(d, \mathcal{V}_i) \times P(q|\theta_d). \quad (10)$$

We normalize Soft.ReDDE features across verticals such that $\sum_{\mathcal{V}_j \in \mathcal{V}} \text{Soft.ReDDE}_q(\mathcal{V}_j) = 1$.

Compared to ReDDE, Soft.ReDDE has two potential benefits. First, every document in the ranking contributes, more or less, depending on its correlation, to a vertical's score. Second, it is not necessary to manually map documents to verticals, so external collections can be used more freely. In our implementation, we used the full English Wikipedia.

Clarity, ReDDE, and Soft.ReDDE features used the Indri IR toolkit. [3]

Categorical Features.

Categorical features were derived from the topical categories automatically assigned to the top 100 documents returned when issuing the query to a general Web index. Each document in the index is assigned, using a maximum entropy text classifier, at most three categories, resembling nodes from the Online Directory Project (ODP) ontology (e.g., "recreation/sports/basketball"). Categorical features were divided into two distinct sets: *general* (depth one) category features (e.g., "recreation", "science", "health") and *specific* (depth two) category features, each which describes a subcategory of a general category (e.g. "recreation/travel", "recreation/sports", "health/nutrition"). Each category prediction on a document is associated with a confidence value. We set the value of category feature y_i (of depth x) to be the sum of confidence values over all occurrences of the category in the top 100 documents,

$$\text{CAT}_q(y_i) = \sum_{d \in \mathcal{R}_{100}} \sum_{y_j \in \mathcal{Y}_d} \mathcal{I}\big(y_i = \text{depth}_x(y_j)\big) \times P(y_j|D), (11)$$

where $\mathcal{R}_{100}$ denotes the top 100 documents, $\mathcal{Y}_d$ denotes the categories associated with document d, $P(y_j|d)$ is the confidence of predicted category y_j on document d, and function $\text{depth}_x(y_j)$ returns the depth x ancestor of category y_j. For example, depth_1("recreation/sports") returns "recreation". We focused on 14 general category features and 42 specific category features—the union of categories for the queries in our training set. In general, we expect the set of category features to depend on the queries the system is likely to encounter and the target verticals.

5. VERTICAL SELECTION ALGORITHMS

5.1 Single Feature Runs

We evaluated 8 single-evidence baselines: The four combinations of Clarity and ReDDE with vertical- and Wikipedia-sampled collections, the query likelihood given the vertical's query-log language model (allowing and disallowing zero probabilities), Soft.ReDDE, and an approach that always predicts the "no relevant vertical" class. These vertical scoring functions were uniformly adapted for single vertical selection by normalizing across vertical scores and selecting the top vertical, $\tilde{v}$, if its score exceeds a threshold, τ, or else predicting "no relevant vertical".

$$\tilde{v} = \begin{cases} \text{argmax}_{\mathcal{V}_i} \text{score}_q(\mathcal{V}_i) & \text{if } \max_{\mathcal{V}_i} \frac{1}{\mathcal{Z}} \text{score}_q(\mathcal{V}_i) > \tau \\ \emptyset & \text{otherwise} \end{cases},$$

where $\mathcal{Z} = \sum_{\mathcal{V}_j \in \mathcal{V}} \text{score}_q(\mathcal{V}_j)$ and the empty set $\emptyset$ denotes a "no relevant vertical" prediction. Parameter τ was set using a 500 query validation set

5.2 Feature Combination Run

For our multiple feature approach, we trained a multiclass classifier using all features. We trained 19 one-versus-all logistic regression models (one for each of our 18 verticals and one for the "no relevant vertical" class) using the liblinear toolkit [4]. We complemented the "no relevant vertical" classifier using the confidence of the 18 binary vertical classifiers using parameter τ.

These classifiers were combined by predicting vertical $\tilde{v}$ according to,

$$\tilde{v} = \begin{cases} \text{argmax}_{\mathcal{V}_i} P_{\mathcal{V}_i}(Y = 1|q) & \text{if } \max_{\mathcal{V}_i} P_{\mathcal{V}_i}(Y = 1|q) > \tau \\ \emptyset & \text{otherwise} \end{cases},$$

[3] http://www.lemurproject.org/indri/

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

autos	3.0%	images	6.0%	reference	15.4%
directory	4.4%	local	19.1%	shopping	20.3%
finance	2.6%	maps	1.1%	sports	3.3%
games	2.6%	movies	2.3%	travel	8.7%
health	4.3%	music	4.6%	tv	2.7%
jobs	1.5%	news	5.1%	video	3.1%
				no.rel.vertical	26.3%

Table 2: Percentage of queries assigned each vertical. Percentages do not sum to one because queries can be assigned more than one relevant vertical

where $P_{\mathcal{V}_i}(Y = 1|q)$ is the probability of a positive prediction from vertical $\mathcal{V}_i$'s classifier. If the most confident vertical classifier predicts its vertical with confidence below τ, we default to the "no relevant vertical" class. All features were scaled to zero minimum and unit maximum. Features associated one-to-one with a vertical (Clarity, ReDDE, the query likelihood given the vertical's query-log and Soft.ReDDE) were normalized across verticals before scaling. Supervised training/testing was done via 10-fold cross validation. Parameter τ was tuned for each training fold on the same 500 query validation set used for our single feature baselines.

6. DATA

Our evaluation data consisted of 25195 unique queries obtained from a commercial search engine's query-log. Human editors were instructed to assign between zero and six relevant verticals per query based on their best guess of the user's vertical intent. About 70% of queries were assigned either a single relevant vertical or no relevant vertical. About 26% of queries, mostly navigational (e.g., "myspace"), were assigned "no relevant vertical" and 44% were assigned a single relevant vertical. Some queries assigned multiple relevant verticals were ambiguous in terms vertical intent (e.g., query "hairspray" was assigned verticals *movies*, *video*, and *shopping*). Table 2 shows the vertical distribution.

7. EVALUATION

We evaluated single vertical selection in terms of precision, $\mathcal{P}$ (see Equation 1), defined as the percentage of queries for which we either correctly predict a relevant vertical or correctly predict "no relevant vertical". Because we make a single prediction when there are potentially multiple relevant verticals, a recall-flavored performance measure has undesirable properties. For example, if two verticals are perfectly correlated in terms of the queries for which they are relevant, then a classifier that chooses the same vertical each time maximizes our objective (i.e, it selects a correct vertical each time) but recall would be perfect for one vertical and zero for the other. We also show % coverage (% cov), defined as the percentage of queries for which a vertical was predicted (correctly or incorrectly). Significance was tested using a 2-tailed paired t-test on queries.

8. RESULTS

Results for single vertical selection are shown in Table 3.

	$\mathcal{P}$	% cov
clarity.vertical	0.254	3.4%
clarity.wiki	$0.256^{\dagger}$	2.7%
no.rel	$0.263^{\ddagger}$	0.0%
redde.wiki	$0.293^{\ddagger}$	54.4%
q.log	$0.312^{\ddagger}$	61.9%
soft.redde	$0.324^{\ddagger}$	43.6%
redde.vertical	$0.336^{\ddagger}$	45.7%
q.log (zero probs)	$0.368^{\ddagger}$	51.0%
LR	$0.583^{\ddagger}$	64.3%

Table 3: Single Vertical Precision ($\mathcal{P}$). Approaches are listed in ascending order of $\mathcal{P}$. A significant improvement over all worse-performing approaches is indicated with a † at the $p < 0.05$ level and a ‡ at the $p < 0.005$ level.

The `no.rel` approach obtained $\mathcal{P} = 0.263$ because 26.3% of queries had no true relevant vertical. Both Clarity using vertical- and Wikipedia-sampled collections performed significantly worse than `no.rel`. Clarity scores for a given query may not be directly comparable across collections with different corpus statistics. In prior work, Clarity has been used to compare retrievals from different queries on the same collection, but not retrievals from the same query on different collections. Further experiments are needed to determine whether Clarity can be adapted for vertical selection. ReDDE using vertical-sampled documents outperformed ReDDE using Wikipedia-sampled documents, in spite of more verticals having a Wikipedia-sampled collection than a vertical-sampled collection. We examined the types of classification errors each algorithm performed. Both approaches performed comparably with respect to the "no relevant vertical" class. However, `redde.wiki` more often predicted a wrong vertical. Precision on queries for which a vertical was predicted was 0.382 for `redde.vertical` and 0.284 for `redde.wiki`. Our mapping of Wikipedia categories to verticals may have misrepresented one or more vertical.

The query likelihood given the vertical's query-log language model was the best single-evidence predictor. This method performed better when allowing than when disallowing zero probabilities. This may have been due to the non-uniformity of $P(\text{OOV})$ estimates across vertical language models. Each vertical's $P(\text{OOV})$ estimate was based on the frequency of terms not in its top 20000, expected to be greater for verticals with a more open vocabulary. A vertical's $P(\text{OOV})$ estimate affects the probability estimates of within vocabulary terms through discounting. Different $P(\text{OOV})$ estimates across verticals may have made the query likelihood given by different vertical language models less comparable.

Finally, our multiple-feature supervised approach (`LR`) outperformed all single-feature baselines by a large margin—a 58% improvement over the best single-evidence predictor, `q.log`. Such a performance improvement may justify the cost of producing training data in the form of vertical relevance judgements on queries. Our supervised framework has several potential advantages. First, as our results show, it can integrate multiple sources of evidence. Second, by combining vertical-specific classifiers, single-evidence scores (e.g., ReDDE or Clarity) need not be directly comparable across verticals. For example, a classifier may learn to ignore a high ReDDE score if it is unreliable, perhaps due to poor sampling. Third, by sharing all features among vertical-specific base classifiers, a classifier may benefit from another vertical's score if they are correlated.

9. DISCUSSION

In this section, we explore the contribution of different fea-

Leave one feature type out

feature variation	$\mathcal{P}$	% diff	% cov
all	0.583		64.30%
no.q.log	0.583	0.03%	64.36%
no.triggers	0.583	-0.03%	64.30%
no.clarity	0.582	-0.10%	63.68%
no.geo	$0.577^{\ddagger}$	-1.01%	65.30%
no.cat.general	$0.572^{\ddagger}$	-1.84%	63.91%
no.redde	$0.568^{\ddagger}$	-2.60%	60.27%
no.soft.redde	$0.567^{\ddagger}$	-2.67%	62.47%
no.cat.specific	$0.552^{\ddagger}$	-5.33%	64.19%

Leave one sampled corpus out

feature variation	$\mathcal{P}$	% diff	% cov
no.vertical.corpus	$0.577^{\ddagger}$	-3.1%	62.06%
no.wiki.corpus	$0.574^{\ddagger}$	-3.5%	62.67%

Table 4: Feature Set Contribution. A ‡ denotes a significant improvement ($p < 0.005$) over `all`, our classifier using all features

ture sets on precision ($\mathcal{P}$). A feature set is said to contribute significantly if the classifier's performance drops significantly upon removing the feature set.

9.1 Final Prediction Precision

We analyze each feature set's contribution to final prediction precision. We divide this analysis into two parts. First, we do a "leave one feature type out" analysis. Second, we do a "leave one sampled corpus out" analysis, where we compare the contribution of corpus features using vertical- vs Wikipedia-sampled documents.

Table 4 shows the change in precision associated with each feature type. Keep in mind that features were not evaluated in isolation. A non-significant performance drop in $\mathcal{P}$ does not necessarily mean the feature captures no useful evidence, as features may be correlated.

In terms of feature types, omitting `q.log`, `triggers`, and `clarity.*` features did not produce a significant drop in $\mathcal{P}$. It is possible that `q.log` features, the best single-evidence predictor, did not contribute significantly because they are correlated with `soft.redde` features, which did contribute significantly. A positive trigger class was predicted only for 4367 (18%) queries, suffering from low coverage. Clarity scores for the same query may not be directly comparable across collections.

Corpus-based features contributed the most. The largest contribution came from `cat.specific` features. Interestingly, categorical features are not derived from resources associated with a vertical (i.e., vertical documents or queries). The classifier learns to associate these features with a vertical from training data. The contribution of `cat.specific` features was significantly greater than that of `cat.general` features because `cat.general` categories were too coarse to discriminate between some verticals. For example, the general category *recreation* conflates *recreation/sports*, *recreation/auto*, and *recreation/travel*, which map conceptually to different verticals. The second and third most helpful features were `soft.redde` and `redde.*` features, respectively.

To evaluate the usefulness of evidence derived directly from the vertical, we omitted ReDDE and Clarity features using our vertical-sampled collections (`no.vertical.corpus`). Likewise, to evaluate the usefulness of evidence derived from surrogate corpora, we omitted ReDDE and Clarity features using our Wikipedia-sampled collections (`no.wiki.corpus`). Removing either set of features produced a significant drop in $\mathcal{P}$. Vertical- and Wikipedia-sampled collections were sampled using different techniques and have a different collection size distribution. Thus, we cannot (and did not intend to) directly compare one against the other. This result, however, shows that surrogate collections can provide evidence complementary to that derived directly from the vertical.

	$\mathcal{P}$	% true	% cov
travel	0.842	8.70%	6.10%
health	0.788	4.30%	3.40%
games	0.771	2.60%	2.10%
music	0.772	4.60%	3.80%
autos^{+}	0.730	3.00%	2.00%
sports^{+}	0.726	3.30%	2.30%
tv^{+}	0.716	2.70%	1.50%
movies	0.688	2.30%	1.40%
finance	0.655	2.60%	1.40%
local	0.619	19.10%	16.70%
jobs	0.570	1.50%	0.60%
shopping	0.563	20.30%	16.80%
images	0.483	6.00%	1.90%
no.rel.vertical$^{+,*}$	0.481	26.30%	35.70%
video	0.459	3.10%	0.40%
news	0.456	5.10%	0.80%
reference	0.348	15.40%	3.10%
maps$^{+,*}$	0.000	1.10%	0.00%
directory*	0.000	4.40%	0.00%
$\mathcal{P}_{\text{macro}}$	0.561		

Table 5: Per vertical precision ($\mathcal{P}$). $\mathcal{P}_{\text{macro}}$ is the average of per vertical $\mathcal{P}$. Verticals without a query-log are marked with $+$. Verticals without a Wikipedia-sampled surrogate corpus are marked with $*$.

9.2 Per Vertical Precision

Table 5 shows precision per vertical/class using all our features, listed in descending order of precision. Column "% true" is the percentage of queries with the vertical as a true relevant vertical. Column "% cov" is the percentage of queries with the vertical as the predicted relevant vertical. Note that the "% true" column does not sum to one because queries may have more than one true relevant vertical. Column "% cov" does sum to one because a single vertical/class was predicted per query. Although "% cov" can be expected to be less than "% true", ideally they should be comparable.

As previously noted, some verticals lacked query-logs ($+$) and/or a Wikipedia-sampled surrogate corpus ($*$). Verticals *autos*, *sports*, and *tv* performed well in spite of lacking features derived from query-logs. Verticals *video*, *news*, and *reference* performed poorly in spite of having all resources. Therefore, the difference in performance across verticals cannot be attributed only to missing features.

The system performed best on verticals that focus on a coherent topic with identifiable vocabulary (i.e., *travel*, *health*, *games*, *music*). The vocabulary associated with these verticals may have been the least confusable with that of other verticals. Precision was higher for these verticals than verticals *shopping* and *reference* and the "no relevant vertical" class, which had more positive examples for training.

The system performed worst on the verticals *images*, *video*, *news*, *reference*, *maps*, and *directory*. The *maps* vertical had the fewest positive instances for training, was featureimpoverished, and probably confusable with *local* and *travel*. Verticals *images* and *video* focus on a type of media rather than a specific genre. Queries related to *reference* and *directory* characterize broad encyclopedic information needs. The *news* vertical tends to be highly dynamic and may require features related to bursts in content demand, possibly derived from same-day vertical query-logs.

With respect to the "no relevant vertical" class, coverage was high and precision was below 50%. Although our evaluation metric weights all false positive errors equally, in some cases a "no relevant vertical" false positive may be less costly than a vertical false positive. A user may be more annoyed by seeing a non-relevant vertical display than by not seeing a relevant vertical display. Of our misclassifications on queries with at least one true relevant vertical ($|\mathcal{V}_q| > 0$), 57% of the time we incorrectly predicted "no relevent vertical" and 43% a non-relevant vertical.

10. CONCLUSIONS

In the context of resource selection for federated search, this work contributes several meaningful results. First, most prior work in resource selection has studied corpus-based evidence derived from the target collections. The use of collection-specific query-logs for resource selection has not been previously studied. This is in part because in an uncooperative environment, query-logs of searchable collections may be inaccessible. Our results show that in vertical selection, a type of cooperative federated search, query-logs are useful. Ranking verticals by the query likelihood given the vertical's query-log language model was the best single-evidence predictor. In our supervised model, query-logs were used successfully to sample from vertical collections and to associate non-vertical documents (i.e., Wikipedia articles) with vertical collections. Second, some verticals (e.g., *video*) are likely to be text-impoverished. We presented methods for successfully associating non-vertical, text-rich documents with verticals, which makes it possible to use existing techniques (e.g., ReDDE) for vertical selection. Finally, most prior work in resource selection has focused on unsupervised or weakly supervised collection ranking methods. Our classification-based approach to vertical selection allows us to combine features without manually associating them with a vertical. For example, our categorical and geographic features, which are not derived from a vertical resource, contribute significantly to prediction accuracy.

This work could be extended in several directions. Our corpus and query-log features are derived from external resources. Although the proposed approach requires training data, it may not be necessary to retrain the model frequently, as long as the external resources used to compute these features reflect changes in the vertical's relevance to a topic. Models that use only query string features may have to be retrained more frequently. Future work might empirically evaluate the robustness of non-lexical features derived from external resources in a dynamic environment. Also, some verticals are bound to be resource-impoverished (e.g., lack query-logs or text-rich documents) and may require incorporating user feedback into the selection model.

11. ACKNOWLEDGMENTS

We would like to thank Jean François Beaumont, Daniel Boies, Hugues Bouchard, Deborah Donato, Rosie Jones, Remi Kwan, Vanessa Murdock, Jian-Yun Nie, Jean François Paiement, and Alexandre Rochette for helpful discussions and feedback. This work was supported in part by the NSF grant IIS-0841275 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.

12. REFERENCES

[1] 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, 2005.

[2] S. M. Beitzel, E. C. Jensen, D. D. Lewis, A. Chowdhury, and O. Frieder. Automatic classification of web queries using very large unlabeled query logs. *TOIS*, 25(2):9, 2007.

[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. Distributed information retrieval. In W. B. Croft, editor, *Advances in Information Retrieval*, pages 127–150. Kluwer Academic Publishers, 2000.

[5] J. Callan and M. Connell. Query-based sampling of text databases. *TOIS*, 19(2):97–130, 2001.

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

[7] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *SIGIR 2002*, pages 299–306, 2002.

[8] F. Diaz. Integration of News Content Into Web Results. In *WSDM 2009*, pages 182–191, 2009.

[9] L. Gravano, H. Garca-molina, A. Tomasic, I. Rocquencourt, and N. L. Gravano. Gloss: Text-source discovery over the internet. *Transactions on Database Systems*, 24:229–264, 1999.

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

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

[12] V. Murdock and M. Lalmas, editors. *SIGIR 2008 Workshop on Aggregated Search*, 2008.

[13] 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.

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

[15] M. Shokouhi, J. Zobel, S. Tahaghoghi, and F. Scholer. Using query logs to establish vocabularies in distributed information retrieval. *Inf. Process. Manage.*, 43(1):169–180, 2007.

[16] L. Si. *Federated Search of Text Search Engines in Uncooperative Environments*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2006.

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

[18] I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *Transactions on Information Theory*, 37, 1991.

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

[20] B. Yuwono and D. L. Lee. Server ranking for distributed text retrieval systems on the internet. In *DASFAA 1997*, pages 41–50. World Scientific Press, 1997.