Ranking Classes of Search Engine Results

Zheng Zhu, Mark Levene
Department of Computer Science and Information Systems, Birkbeck College, University of London, Malet Street , London, UK
zheng@dcs.bbk.ac.uk, mark@dcs.bbk.ac.uk

Ingemar Cox
Department of Computer Science, University College London, Gower Street, London, UK
ingemar@cs.ucl.ac.uk

Keywords: Ranking, Classification

Abstract: Ranking search results is an ongoing research topic in information retrieval. The traditional models are the vector space, probabilistic and language models, and more recently machine learning has been deployed in an effort to learn how to rank search results. Categorization of search results has also been studied as a means to organize the results, and hence to improve users search experience. However there is little research to-date on ranking categories of results in comparison to ranking the results themselves.
In this paper, we propose a probabilistic ranking model that includes categories in addition to a ranked results list, and derive six ranking methods from the model. These ranking methods utilize the following features: the class probability distribution based on query classification, the lowest ranked document within each class and the class size.
An empirical study was carried out to compare these methods with the traditional ranked-list approach in terms of rank positions of click-through documents and experimental results show that there is no simpler winner in all cases. Better performance is attained by class size or a combination of the class probability distribution of the queries and the rank of the document with the lowest list rank within the class.

1 Introduction

Existing search engines such as Google, Bing and Yahoo return a list of retrieved documents based on each document's relevance to a user's query. Web users scroll through the result set to find a document that satisfies their information need. Since documents are only organized based on their relevance score, it is not uncommon for documents on one topic to be scattered throughout the result list. Thus, for example, a search on the keyword, "*jaguar*" will produce results in two topic areas, cars and animals, but all the documents on the topic of cars will not come before, or after, documents on animals.

Several researchers have proposed grouping together documents with a common theme. The rationale for this is the hypothesis that users should be able to quickly identify the topic area related to their query, and thereby avoid the need to look at documents in other topic areas.

There are two broad approaches to grouping. The first is based on clustering, while the other is based on classification. A key distinction between the two approaches is how groups are labeled. In clustering, a retrieved set of documents, the result set, is clustered into a set of groups based on a clustering algorithm. Each group is then assigned a labeled to describe its associated topic area. This label is automatically derived from the documents contained within the group. Not only might these labels be unfamiliar to a user, but the same topic area may be assigned different labels depending on the documents in the group. A recent survey of clustering of search results can be found in (Carpineto et al., 2009).

The classification approach has a predefined (Chen and Dumais, 2000; Zeng et al., 2004; Zhu et al., 2008) set of groups/topic areas, typically derived from an ontology. Thus, the label for a group/class is predefined and is independent of the documents contained within it. Of course, these labels may also be unfamiliar to a user. However, since the label for a class does not change, (i) users have an opportunity to learn the

meaning, and (ii) the labels can be carefully chosen to help users understand what the class represents.

There has been considerable work on document categorization, which is discussed in Section 2. However, this is not the main focus of our work.

Once a document result set has been categorized, the users is first presented with a list of the categories represented by documents. The user then selects a category and the documents within this class are then presented. In such an arrangement, it is necessary to rank classes presented to the user. However, despite the fact that document ranking remains one of the most active research areas in information retrieval, there has been surprisingly little research on ranking classes.

In this paper, we investigate various approaches to ranking classes and how such rankings affect the number of classes and documents a user must inspect prior to finding a documents satisfying his or her information need. Section 3 describes the details of the class rankings we consider. Section 4 then describes our experimental methodology, and Section 5 summarizes our experimental results. Finally, Section 6 provides a discussion of the results.

2 Related Work

There has been significant work on clustering and classification of search results (Carpineto et al., 2009; Chen and Dumais, 2000; Zeng et al., 2004; Zhu et al., 2008). Our interest is specific to how previous research ranked classes. There are two main approaches to doing so.

The first approach (Chen and Dumais, 2000) ranks classes based on their size, i.e. the top-ranked class contains more documents than the other classes. This approach is simple and assumes that a class's relevance is purely a function of the number of documents it contains.

The second approach (Zamir and Etzioni, 1999; Zeng et al., 2004) ranks classes based on various properties associated with the documents in each class. For example, in (Zamir and Etzioni, 1999) the authors ordered the clusters by their "estimated coherence", defined as a function of the number of documents each phrase contains and the number of words that make up its phrase. And in (Zeng et al., 2004), all n-gram ($n \leq 3$) were first extracted from search results as candidate phrases, and salient scores were calculated from a regression model trained from manually labeled data. The features included phrase frequency/inverted document frequency, phrase length, intra-cluster similarity, cluster entropy and phrase independence. The salient scores are used to rank the clusters.

A second important consideration is how documents are ranked within a class. Several alternatives have been proposed. However, in the work described here, the relative ranking of documents within a class is the same as their relative ranking in the original result set. We believe that this is an important experimental design consideration. In particular, if the ranking of documents in altered within a class, then it is very difficult to determine whether any improvement is due to (i) the class ranking, (ii) the new document ranking, or (iii) a combination of (i) and (ii). Thus, in order to eliminate this potential ambiguity, we maintained relative document rankings within classes. Thus, any improvement must only be due to the class ranking.

3 Class-Based Ranking Method

Before we discuss the various class ranking algorithms we examined, it is useful to first describe how we evaluated performance. For a conventional system, in which the result set is displayed as a ranked list of documents, i.e. we have a list of documents $\{d_1, d_2, \cdots d_N\}$, if the k-th document is the desired document, then the user must look at k documents $(d_1 \cdots d_k)$. We refer to k as the "list rank" of the document, since it was ranked k in a one-dimensional list of retrieved documents. Clearly, the lower the list rank, the quicker the user will find the desired document.

The performance of a classification-based system is slightly more complicated to define. Consider the case where the user is looking for document, $d_{i,j}$, where i denotes the rank of the class the document is contained in, and j is the document's rank within this class. Thus, a user must look at i class labels and then j document snippets in order to find document, $d_{i,j}$, a total of $(i+j)$ classes and documents. We refer to $(i+j)$ as the "classification rank" (CR) of document $d_{i,j}$.

For any classification-based system, we compare a document's classification rank to its original, corresponding list rank, k. We say that the classification-based system outperforms the list-based system if $i+j<k$, i.e., the user looks as few classes and documents.

Note that we have implicitly assumed that (i) documents are correctly assigned to classes, and (ii) that users always choose the correct class. In practice, this will not always be the case. However, the assumption simplifies our analysis and permits us to de-

termine the best-case performance of class-based retrieval systems. The reader is directed to (Zhu et al., 2008) for more discussion of when this assumption does not hold.

Given an initial retrieval set, $D = \{d_1 \cdots D_{|D|}\}$, that has been grouped into a set of classes, $C = \{c_1 \cdots c_{|C|}\}$, we now wish to determine the relative ranking of each class. The information we have available is the query, q, and the documents, D. We take a straightforward Bayesian approach, i.e. we wish to estimate the probability, $P(c_i|q)$, that class, c_i is relevant, conditioned on the query, q,

$$P(c|q) \approx P(c)P(q|c). \quad (1)$$

We now consider how each of these two terms might be estimated.

3.1 Query-dependent Classification

The probability, $P(q|c)$, is the likelihood that class c generates query q. It is related to solving the query classification problem (Shen et al., 2006). This problem has received significant recent attention (Broder et al., 2007; Cao et al., 2009) since the 2005 KDD Cup competition (Li and Zheng, 2005). Solutions to this problem typically enrich the query terms using keywords from the top-ranked documents in the result set[1].

For a test query, the class probability distribution is given by

$$P(q|c) \approx \frac{1}{1+exp(-(w_c^T x+b))}, \quad (2)$$

where, x is term vector containing the query and enrichment terms, and the weights w_c and intercept b are derived from L2-regularized logistic regression (Lin et al., 2007), based on a set of labeled examples. Alternative estimates for $P(q|c)$ are possible, but are not considered in this paper.

3.1.1 Query-based rank (QR)

If each class is only ranked based on Equation (2), i.e. we ignore the query-independent term, $P(c)$, in Equation (1), we refer to it as *query-based rank (QR)*.

3.2 Query-Independent Classification

To estimate $P(c)$, we make use of the available documents in retrieved result set.[2] We further assume that only documents contained in the class, D_c, affect the probability of the class, i.e.

$$P(c) \approx P(c|D_c). \quad (3)$$

We believe this assumption is reasonable as the class probability is mainly determined by the information within the class, not by the other classes. Thus, Equation (1) becomes

$$P(c|q) \approx P(c|D_c)P(q|c). \quad (4)$$

We now considered several ways to estimate the conditional probability $P(c|D_c)$[3].

3.2.1 Document-based Rank (DR)

One approach to estimating $P(c_i|D_c)$ is to base the probability on the top-ranked document in the class, c_i. The reader is reminded that the original rank order of documents in the result set is retained within a class.

The j-th ranked document in class, c_i, is denoted $d_{i,j}$. The document's corresponding list rank, i.e. its rank prior to classification, is denoted $s(d_{i,j}) = s(d_k) = k$.

We then define the conditional probability, $P(c|D_c)$ as

$$P(c|D_c) = f(s(d_{i,1})), \quad (5)$$

where $c = c_i$, and $s(d_{i,1})$ is the list rank of the top document in class c_i. The function, $f(x)$, can be any monotonically decreasing function in the value x. In this paper we consider the inverse function defined by

$$f(x) = \frac{1}{x} \quad (6)$$

and the logistic function defined by

$$f(x) = \frac{1}{1+exp(x)}. \quad (7)$$

If we only rank classes based on the query-independent factor of Equation (1), then both functions, $f(x)$, will rank the classes in the same order. In the subsequent experiments, we therefore only consider the inverse function, and rank classes according to

$$P(c|D_c) = \frac{1}{s(d_{i,1})} \quad (8)$$

We refer to this as the *document-based rank* (DR).

[1] The implicit assumption, shared with pseudo-relevance feedback, is that the top-ranked documents are relevant.

[2] Note that we can estimate $P(c)$ based on the class distribution of the collection, but this is beyond our scope and it is not as accurate as the retrieved result set.

[3] More precisely, we presents functions to approximate the likelihood of the class c to be examined rather than probability, as the results do not sum to 1.

3.2.2 Size Rank (SR)

In contrast to ranking classes based on the top-ranked document in the class, we also consider the case where the conditional probability $P(c|D_c)$ is based on the class size. That is, the bigger the class, i.e. the more documents assigned to the class, the more important the class is considered to be. Thus, we have

$$P(c) \approx P(c|D_c) = \frac{|c|}{\sum_i |c_i|}, \quad (9)$$

where $|c|$ is the number of elements in the class c, and the denominator is the size of result set.

Again, if we only rank classes based on the query-independent factor of Equation (1), and the class ranks are based on the size of the classes, as defined in Equation (9), then we refer to this as the *Size Rank* (SR).

3.3 Additional Class Ranking Models

From Equation (2) and the definitions for $P(q|c)$ and $P(c)$, we can now define a variety of different models for ranking classes based on *both* the query-dependent and query-independent probabilities..

3.3.1 Query/ Inverse Rank (QD_IR)

If the class ranking is determined by the product of Equations (2) and (6), then we obtain

$$P(c|q) \approx \frac{1}{1+exp(-(w_c^T x+b))} \times \frac{1}{s(d_{c,1})}. \quad (10)$$

We call this rank the *Query/Inverse Rank(QD$_I$R)*.

3.3.2 Query/Logistic Rank (QD_LR)

The *Query/Logistic Rank (QD$_L$R)* is correspondingly defined as

$$P(c|q) \approx \frac{1}{1+exp(-(w_c^T x+b))} \times \frac{1}{1+exp(s(d_{c,1}))}. \quad (11)$$

3.3.3 Query/Size Rank (QSR)

Similarly, if the class ranking is determined by the product of Equations (2) and (9), then we have

$$P(c|q) \approx \frac{1}{1+exp(-(w_c^T x+b))} \times \frac{|c|}{\sum_i |c_i|}. \quad (12)$$

We call this rank the *Query/Size Rank (QSR)*.

3.3.4 Summary of Ranks

We distinguish the methods for estimating $P(c|q) \approx P(q|c)P(c|D_c)$ according to the different methods presented above. The *list rank* (LR) is the original rank of a document in the result set, i.e. before any classification. We then consider two query-independent methods of ranking classes, based on (i) the class size, i.e. *size rank* (SR), and (ii) the top-ranked document in each class, i.e. *document rank* (DR). We also consider ranking classes based only of the query-dependent term, i.e. the *query-based rank* (QR). Finally, we consider ranking classes based on a combination of query-dependent and query-independent terms. In all these cases, the query-dependent term is based on Equation (2), and we vary the query-independent term. Specifically, we consider (i) *query/size rank* (QSR) in which the conditional probability, $P(c|D_c)$ is based on the size of a class, and (ii) *query inverse rank* (QD_IR) and *query logistic rank* (QD_LR), both of which are based on a function of the top-ranked document in each class, and where this function is the inverse function or the logistic function, respectively. The various methods are summarized in Table 1.

Table 1: The summary of the ranks

Notation	Meaning
LR	List Rank, the rank of the results returned by the search engine.
SR	Size-based Rank computed according to (Equation (9))
DR	Document-based Rank computed according to (Equation (6)).
QR	Query-Based Rank computed according to (Equation (2))
QSR	Query/Size Rank computed according to (Equation (12))
QD$_I$R	Query/Inverse Rank computed according to (Equation (10)).
QD$_L$R	Query/Logistic Rank computed according to (Equation (11)).

4 Experimental Set Up and Data

Evaluation of information retrieval systems requires knowledge of a document's relevance with respect to a query. One indirect source of such information is query logs. These logs consist of queries together with associated click-through data. Previous research (Liu et al., 2007) showed that retrieval evaluation based on query logs yields similar per-

formance to retrieval evaluation based on traditional human assessors. We used a subset (the RFP 2006 dataset) of an MSN query log, collected in spring 2006. The log contains approximately 15 millions queries. Each query has associated with it either (i) no click-through data (*no-clicks*), (ii) one click-through data (*one-click*), or (iii) multiple click-through data (*multiple-click*).

We ignore queries for which there is no associated click-through data (approximately 6.1 million queries). For *one-click* queries, of which there are approximately 7.2 million, we assume the query was satisfied by the document clicked on. For *multiple-click* queries, of which there are approximately 1.6 million, we assume that the query was satisfied by the last document clicked on. We realize that this will not always be true, but assume that it is true sufficiently often to provide us with reliable results. Note that this assumption has been partially justified by other researchers (Joachims et al., 2005), in the context of *multiple-click* queries.

The query log does not include information describing the result set returned in response to the query. Rather, the click-through data only identifies those documents in the result set that the user clicked on. Of course, in order to evaluate the various classification based systems, we need access to the complete result set. We acquired this information by issuing the query to a search engine, specifically Microsoft Live Search on May 2009, which was subsequently replaced by Bing. Note that for some queries, the url's clicked on in the query log are not returned by the search engine, either because its rank is beyond our retrieved result set or the url is no longer available. We discarded such queries. In the case where the url *is* returned in the result set, we assume the the result set returned by Live Search is similar to the result set observed by the user during the collection of the log. We acknowledge that this is a major assumption, which cannot be verified, and future work is needed to repeat these experiments on a more recent data set.

We now describe our experimental methodology. The total number of unique queries which have a click-through is 3,545,500. Among them, there are 658,000 multiple-click queries whose top-20 search results have been downloaded by us before Microsoft upgraded Live to Bing. We took a random sample of 20,000 one-click queries and 20,000 multiple-click queries, whose click-through occurred both in the query log and in our retrieved data set.

For each query, the list rank of the relevant document (i.e. the document clicked on for *one-click* or the final document clicked on for *multiple-click*) were recorded. Next, the documents in the result set were classified into one of 27 classes; these classes are enumerated in Table 4 in the appendix; see (Bar-Ilan et al., 2009) for more detail about this ontology.

In order to classify the documents we compute $P(c|q)$ from Equation (2), using logistic regression. The training data is obtained from two manually classified subsets of an AOL search log (Beitzel et al., 2005). The first one contains 9,913 manually classified queries, resulting from a Master's level Information Science class assignment at Bar-Ilan University during 2007 (Bar-Ilan et al., 2009). The second one is a labeled log file from AOL's research lab. We enriched the query with the top-10 snippets in the result set, the titles of the top-10 documents and their urls to form the vector x. Then for the test query, we enriched the query with the same information and predict the probability via Equation (2).

To keep our data consistent, for a given query, we record the list rank of the given click-through in the result set, as it may be different from the one recorded in the log data. Then query classification is carried out by enriching the query with the top-10 results. In this manner we attain the class probability distribution. After that we assign each result into its class.

5 Experimental results

In Section 5.1, we present the results for multiple-click queries (m-clicks). The results for one-click queries (1-click) are presented in Section 5.2.

5.1 Results for Multiple-Click Queries

Each target document has an original list rank from 1 to 20. Table 2 shows for each list rank, the mean value of the corresponding classification rank. Column 1 of Table 2 provides the list rank of the target documents for the top-10 documents, i.e., there is no classification, only a traditional list of retrieved documents. Column 2-7 provide the equivalent classification ranks (CR), i.e., the total number of classes and documents a user must examine in order to find the target document. If the CR is less than the LR, then the classification based system outperforms a traditional system. The smallest value indicates the least number of documents that a user must inspect before finding the desired document. Columns 8-14 provide the same data for list ranks between 11 and 20.

We can see that for list ranks between 1 and 4, all CR values in the corresponding row are greater than the list rank. For a list rank of 5 and higher, there is always a CR that is less than the corresponding list

Table 2: The comparison of class based rank for the last click through according to list rank(m-clicks).

LR	*QR*	*DR*	*SR*	*QSR*	QD_LR	QD_IR	*LR*	*QR*	*DR*	*SR*	*QSR*	QD_LR	QD_IR
1	3.00	**2.00**	2.88	2.79	2.08	2.13	11	7.46	7.69	**7.16**	7.18	7.60	7.57
2	3.55	3.00	3.39	3.32	**2.93**	3.01	12	7.89	8.07	**7.52**	7.59	8.01	7.93
3	3.95	3.64	3.73	3.68	**3.58**	3.63	13	8.38	8.58	**7.99**	8.05	8.50	8.41
4	4.45	4.23	4.20	4.20	**4.15**	4.18	14	8.41	8.66	**8.07**	8.13	8.57	8.50
5	4.81	4.80	4.63	**4.58**	4.71	4.70	15	8.99	9.17	**8.56**	8.68	9.10	9.05
6	5.28	5.26	**4.95**	4.99	5.19	5.18	16	9.55	9.84	**9.25**	9.32	9.74	9.64
7	5.65	5.86	**5.41**	5.42	5.74	5.68	17	9.64	10.03	**9.23**	9.29	9.94	9.83
8	6.11	6.24	5.86	**5.85**	6.15	6.12	18	10.50	10.82	**10.12**	10.17	10.72	10.63
9	6.37	6.58	**6.13**	**6.13**	6.49	6.44	19	10.69	10.95	**10.27**	10.36	10.88	10.80
10	6.83	7.05	**6.54**	6.56	6.97	6.89	20	11.27	11.52	10.98	**10.95**	11.41	11.32

rank. However, it is not the case that a single particular classification-based system is superior, although in most cases, ranking classes by class size (SR) has best performance.

One reason why the classification-based rankings do not yield an improvement for list ranks of 5 or less, is that classification ranks introduce a small overhead, i.e. an extra click to examine the class the result is in. Thus, if the desired document is ranked first, i.e. its list rank is 1, and, for classification-based ranking, this document is the first document in the first class, the user must examine one class and one document, thereby incurring a cost of 2.

It is interesting to note that for an initial list rank of 5 or less, the best classification-based methods are provided by document-based ranking methods, specifically DR and QD_IR. However, for list ranks greater than 5, classification methods based on class size perform best. For an initial list ranks between 5 and 10, we observe that SR and QSR are the best, and for an initial list rank greater than 10, SR most usually performs best. This might be due to the fact that for list ranks greater than 10, the initial query is, by definition, poor, and therefore ranking classes based only on the query-independent component is usually superior. However, the difference in performance between the two methods is actually quite small.

Figure 1 shows the cumulative distribution of target documents for each method. We see that for list rank, approximately 25% of target documents are at list rank of 1, and 35% have a list rank less than or equal to 2. No class rank system has a class rank of 1 because of the overhead it introduces. Approximately 25% of clicked document have a classification rank of 2. The list rank and classification rank cross at rank 4. Approximate 50% of documents have a rank of 4 or less for all systems. Conversely, 50% of clicked documents have a rank greater than 4, and in those cases, a class based system performs better.

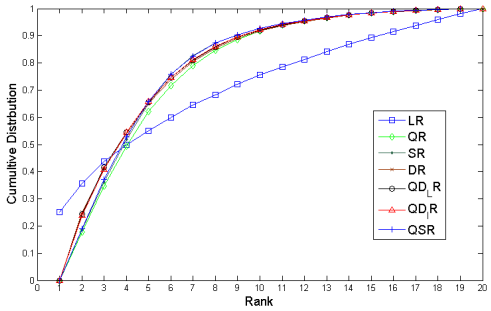

Figure 1: The cumulative distribution of the rank of target documents for m-clicks queries.

5.2 Results for One-Click Queries

The performance for one-click queries is very similar to the multiple-click queries.

Table 3 shows the mean value of the respective classification rank for each list rank. Column 1 of Table 3 provides the list rank of the clicked document in the list-ranked results set. We can see that all classification ranks perform worse than the list rank when list rank is less than or equal to 4, which is similar to the results in Table 2. The classification rank outperforms the list rank when the list rank is greater than 4. In those cases, once again, ranking classes based on class size, i,e, SR and QSR, exhibit best results.

Figure 2 shows the cumulative distribution of target documents for each method, for 1-click queries.

Compared to the Figure 1, the list rank more strongly dominates the top ranks, i.e., approximate 47% of target documents are at list rank of 1 and about 70% of target documents are at ranks of three or less. As before, ranking classes based on a document-based ranking provides the best performance for the classification methods when the list rank is less than 5.

If the initial query is good, i.e. the target docu-

Table 3: The comparison of class based rank for the one click through according to list rank(1-click).

LR	*QR*	*DR*	*SR*	*QSR*	QD_LR	QD_IR	*LR*	*QR*	*DR*	*SR*	*QSR*	QD_LR	QD_IR
1	3.11	**2.00**	2.85	2.79	2.08	2.14	11	8.03	8.06	**7.54**	7.67	8.03	8.03
2	3.57	3.00	3.32	3.28	**2.94**	3.04	12	7.95	8.03	**7.38**	7.47	7.98	7.93
3	4.13	3.68	3.74	3.76	**3.61**	3.68	13	8.23	8.25	**7.78**	**7.78**	8.17	8.14
4	4.56	4.25	4.20	4.21	**4.18**	4.22	14	8.71	8.67	**8.28**	8.29	8.60	8.56
5	4.98	4.78	**4.62**	4.65	4.71	4.73	15	9.04	9.18	**8.61**	8.67	9.13	9.10
6	5.40	5.31	**5.04**	5.05	5.25	5.24	16	9.55	9.85	9.28	**9.26**	9.72	9.57
7	5.87	5.85	**5.51**	5.52	5.77	5.74	17	10.05	10.37	9.81	**9.79**	10.31	10.17
8	6.22	6.16	**5.88**	5.92	6.06	6.06	18	9.88	10.14	9.65	**9.60**	10.09	9.91
9	6.40	6.51	**6.05**	6.09	6.43	6.36	19	10.95	11.12	**10.38**	10.60	11.05	11.01
10	7.01	7.11	**6.63**	6.64	7.04	7.01	20	10.95	11.17	**10.51**	10.58	11.14	11.03

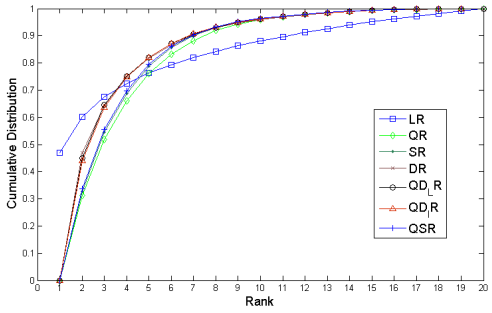

Figure 2: The cumulative distribution of the rank of target documents for 1-click queries

ment has a list rank less than 5, then displaying results as a traditional one-dimensional list is superior. However, for queries where the initial list rank is 5 or more, classification based ranking offers better results. It would therefore be interesting to investigate a hybrid method for displaying the result set, in which the top-ranked document is displayed first, followed by categorized results.

6 Concluding Remarks

We proposed a probabilistic model for ranking classes, and derived six ranking functions from this model. Two models, SR and DR, were query-independent, and one model, QR, was query-dependent. A combination of these resulted in the models QSR, and QD_IR and QD_LR.

Within each class, the rank order of documents was identical to that in the original list rank. We believe this is an important experimental control in order to be certain that any improvements in ranking are solely due to the classification methods under investigation.

We examined a subset of queries derived from an MSN log recorded in Spring 2006. This subset consisted of 20,000 queries for which 1-click was associated with each query, and 20,000 queries for which multiple-clicks were associated with each query. The two data sets were examined independently, but experimental results are consistent across both. In particular, we observed that for target documents with an initial list rank less than 5, the classification-based methods offered no advantage. This is partly due to the fact that these methods introduce a small overhead, i.e. to even examine the first document in the first class requires two, rather than one click. However, for target documents with an initial list rank of 5 or more, classification methods are better. Of the six methods examined, the two based on class size, SR and QSR, performed best. The difference between these two methods is also small.

For the case where the target document has an initial scroll rank of 5 or less, the document-based classification methods performed best. However, they were inferior to traditional list rank, i.e. no classification.

The fact that traditional list rank performs well for good queries, i.e. where the initial rank of target documents is less than 5, while classification-based methods perform well for poorer queries, i.e. where the initial rank of target documents is greater than 4, suggests that some form of hybrid method should be investigated. For example, one could display the top-ranked document followed by categorized results. This would be an interesting line of future investigation.

A key assumption of our experimental results is that the retrieved results obtained using Live Search are similar to those observed by users at the time the query log was collected in Spring 2006. It is not possible to verify this assumption, and it would be interesting to repeat our experiments on more recent data.

In our work, we also assume that classification is perfect, i.e. that documents are correctly classified

and that users correctly identify the target class. In practice, this will not be the case, and our experimental results must be considered a best case. Nevertheless, we are optimistic that classification errors can be kept small. In particular, documents could be classified during indexing, when considerably more information than just the result set is available. And, over time, users are likely to learn the classification ontology and increase the frequency of choosing the correct class.

A Appendix

The list of classes, See Table 4.

REFERENCES

Bar-Ilan, J., Zhu, Z., and Levene, M. (2009). Topic-specific analysis of search queries. In *WSCD '09: Proceedings of the 2009 workshop on Web Search Click Data*, pages 35–42, New York, NY, USA. ACM.

Beitzel, S. M., Jensen, E. C., Frieder, O., Grossman, D., Lewis, D. D., Chowdhury, A., and Kolcz, A. (2005). Automatic web query classification using labeled and unlabeled training data. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 581–582, New York, NY, USA. ACM.

Broder, A. Z., Fontoura, M., Gabrilovich, E., Joshi, A., Josifovski, V., and Zhang, T. (2007). Robust classification of rare queries using web knowledge. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 231–238, New York, NY, USA. ACM.

Cao, H., Hu, D. H., Shen, D., Jiang, D., Sun, J.-T., Chen, E., and Yang, Q. (2009). Context-aware query classification. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, New York, NY, USA. ACM.

Carpineto, C., Osiński, S., Romano, G., and Weiss, D. (2009). A survey of web clustering engines. *ACM Comput. Surv.*, 41(3):1–38.

Chen, H. and Dumais, S. (2000). Bring order to the web: Automatically categorizing search results. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 145–152, New York, NY, USA. ACM Press.

Joachims, T., Granka, L., Pan, B., Hembrooke, H., and Gay, G. (2005). Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA. ACM.

Li, Y. and Zheng, Z. (2005). Kdd cup 2005. Online at http://www.acm.org/sigs/sigkdd/kdd2005/kddcup.html.

Lin, C.-J., Weng, R. C., and Keerthi, S. S. (2007). Trust region newton methods for large-scale logistic regression. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 561–568, New York, NY, USA. ACM.

Liu, Y., Fu, Y., Zhang, M., Ma, S., and Ru, L. (2007). Automatic search engine performance evaluation with click-through data analysis. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1133–1134, New York, NY, USA. ACM.

Shen, D., Pan, R., Sun, J.-T., Pan, J. J., Wu, K., Yin, J., and Yang, Q. (2006). Query enrichment for web-query classification. *ACM Trans. Inf. Syst.*, 24(3):320–352.

Zamir, O. and Etzioni, O. (1999). Grouper: A dynamic clustering interface to web search results. *Computer Networks*, pages 1361–1374.

Zeng, H. J., He, Q. C., Chen, Z., Ma, W. Y., and Ma, J. W. (2004). Learning to cluster web search results. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 210–217, New York, USA. ACM Press.

Zhu, Z., Cox, I. J., and Levene, M. (2008). Ranked-listed or categorized results in ir: 2 is better than 1. In *NLDB '08: Proceedings of the 13th international conference on Natural Language and Information Systems*, pages 111–123, Berlin, Heidelberg. Springer-Verlag.

Table 4: The 27 classes.

1	Art	8	Entertainment	15	Home	22	Religion
2	Auto	9	Finance&Economy	16	Law &Legislation	23	Science
3	Companies&Business	10	Food and drink	17	Nature	24	Shopping
4	Computing	11	Games	18	News	25	Society&Community
5	Directories	12	Government Organization	19	People	26	Sports
6	Education	13	Health&Medicine	20	Places	27	Technology
7	Employment	14	Holiday	21	Pornography		