

# Enhancing Web Search by Promoting Multiple Search Engine Use

Ryen W. White, Matthew Richardson, Mikhail Bilenko  
Microsoft Research  
One Microsoft Way  
Redmond, WA 98052  
{ryenw,mattri,mbilenko}@microsoft.com

Allison P. Heath  
Rice University  
6100 Main Street  
Houston, TX 77054  
aheath@rice.edu

## ABSTRACT

Any given Web search engine may provide higher quality results than others for certain queries. Therefore, it is in users' best interest to utilize multiple search engines. In this paper, we propose and evaluate a framework that maximizes users' search effectiveness by directing them to the engine that yields the best results for the current query. In contrast to prior work on meta-search, we do not advocate for replacement of multiple engines with an aggregate one, but rather facilitate simultaneous use of individual engines. We describe a machine learning approach to supporting switching between search engines and demonstrate its viability at tolerable interruption levels. Our findings have implications for fluid competition between search engines.

## Categories and Subject Descriptors

D.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – *search process, selection process.*

## General Terms

Measurement, Experimentation, Human Factors.

## Keywords

Search engine switching.

## 1. INTRODUCTION

Web search engines such as Google, Yahoo!, and Live Search provide users with keyword access to Web content. According to statistics aggregated by audience measurement and analysis firms such as Nielsen-NetRatings<sup>1</sup> and comScore Media Metrix<sup>2</sup>, although users occasionally use multiple search engines, they are typically loyal to a single one even when it may not satisfy their needs, despite the fact that the cost of switching engines is relatively low (*e.g.*, [19]). While most users appear to be content with their experience on their engine of choice, it is conceivable that many users dislike the inconvenience of adapting to a new engine, may be unaware how to change the default settings in their Web browser to point to a particular engine, or may even be unaware of other Web search engines that exist and may provide better service. Performance differences between Web search engines may be attributable to ranking algorithms and index size, among other factors. It is well understood in the Information Retrieval (IR) community that different search systems perform well for some queries and poorly for others [2,10], which suggests that excessive loyalty to a single engine may actually hinder searchers.

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

SIGIR'08, July 20–24, 2008, Singapore.

Copyright 2008 ACM 978-1-60558-164-4/08/07...\$5.00.

To address this problem, this paper describes a machine learning approach that allows users to leverage multiple search engines by unobtrusively recommending the most effective engine for a given query. The approach relies on a classifier to suggest the top-performing engine for a given search query, based on features derived from the query and from the properties of search result pages, such as titles, snippets, and URLs of the top-ranked documents. We seek to promote *supported search engine switching* operations where users are encouraged to temporarily switch to a different search engine for a query on which it can provide better results than their default search engine. Unsupported switching, whereby users navigate to other engines on their own accord, is a phenomenon that may occur for a number of reasons: users may be dissatisfied with search results or the interface, they may be lured to the engine by advertising campaigns or word of mouth, or they may switch by accident.<sup>3</sup> Results of a log-based study that we present in the paper show that only around 10% of search sessions currently involve more than one search engine. We conjecture that by proactively encouraging users to try alternative engines for appropriate queries (hence increasing the fraction of sessions that contain switching) we can promote more effective user searching for a significant fraction of queries. Empirical results presented in this paper support this claim.

We structure the remainder as follows. Section 2 describes related work and provides some evidence which motivates this research. Section 3 demonstrates the importance and potential benefit of search engine switching using large-scale behavioral datasets. Section 4 describes the machine learning approach to supporting switching behavior, which is empirically evaluated in Section 5. In Section 6 we discuss the implications of this research and future work, followed by conclusions in Section 7.

## 2. RELATED WORK

Prior work in search engine switching has sought to characterize the behavior with the goal of developing metrics for competitive analysis of engines in terms of estimated user preference and user engagement [16], or switching prediction [13]. Other work has focused on building conceptual and economic models of search engine choice. Telang et al. [24] proposed a qualitative model of search engine choice that is a function of the search engine brand, the loyalty of a user to a particular search engine at a given time, user exposure to banner advertisements, and the likelihood of a within-session switch from the engine to another engine. Mukhopadhyay et al. [18] develop an economic model of search engine

<sup>1</sup> <http://www.nielsen-netratings.com>

<sup>2</sup> <http://www.comscore.com>

<sup>3</sup> For example, when a query is typed into a browser's address bar, most browsers forward it to the default search engine.

competition assuming that the switching cost between engines is very low. These studies have focused on understanding and characterizing *existing* switching behaviors in Web search. Although we provide summary statistics on the nature of switching from our observations, our objective is not to characterize switching behavior. Instead, we demonstrate that the utilization of multiple search engines can be advantageous to users and propose a framework that proactively promotes switching.

Commercial meta-search engines such as Clusty<sup>4</sup> and Dogpile<sup>5</sup> attempt to provide access to multiple engines. Given the ranked lists of documents returned by multiple search engines in response to a given query, the objective of meta-search engines is to combine these lists in a way which optimizes the performance of the combination. The IR community has studied meta-search in great detail, with the emphasis on how to merge results from multiple engines (*e.g.*, [7,21,23]), rather than on encouraging people to switch engines as we do in this work. Proactive switching support is an attractive alternative to meta-search for the following reasons: (i) strong brand loyalty may discourage users from migrating to a meta-search engine, (ii) meta-search engines merge search results and obliterate the benefits of interface features of the individual engines, and (iii) meta-searching may be discouraged by search engines as it can negatively impact brand awareness and advertising revenue. We propose an approach whereby users can use their favorite engine but have an alternate engine suggested to them when it is expected to perform better for their current query. In some respects, this is similar to distributed IR (*c.f.* [4]), although we are interested in directing users to the best engine rather than the best collection of documents, and do not merge the search results, as is common practice in that sub-discipline.

Supporting engine switching in real-time requires computationally efficient estimation of relative search result quality across several engines. Measuring quality of search results via metrics such as precision and recall has been central in driving research in IR algorithm design, particularly in the Text REtrieval Conference (TREC) community [11]. Hawking et al. [12] employed a methodology similar to TREC to compare the performance of multiple Web search engines. Others, such as Rorvig [22] and Cronen-Townsend et al. [8], have looked at techniques for predicting the quality of results using the dispersion of the top documents or computing the entropy between the language model for the results and the collection as a whole. Leskovec et al. [17] used properties of search result sets projected onto the Web graph to estimate result quality. Despite their effectiveness at computing result quality, some of techniques depend on relevance judgments, meaning that they cannot scale to unseen queries, and some are computationally expensive, meaning that real-time computation is unfeasible. One key distinction of our work from these approaches is that we directly model *relative* quality of multiple search result sets instead of the quality of any individual result set.

Our framework relies on a classifier to estimate the differences in search result quality between the engines using features computed based on the query and the result pages. Yom-Tov et al. [27] have proposed estimating query difficulty using a machine learning approach based on query-only features, validating it for a distributed IR setting with several collections of newswire documents, rather than Web search as we do in this work. Caption features have already been shown to be important to users in determining which search results to select [5], and query-caption features have

been used in the development of ranking algorithms to improve search [1]. As our empirical results demonstrate, utilizing multiple diverse feature sources is beneficial over query-only features, and is a key performance differentiator for accurate prediction of the most appropriate search engine for a given query in real-time.

### 3. THE CASE FOR MULTI-ENGINE USE

At the outset of our studies, we pursued general statistical clues that could provide insight into the extent to which users switched engines and the potential benefit to them of switching engines. To do so, we used the interaction logs of a large sample of consenting Web users. We begin by describing the statistical properties of search sessions extracted from the logs.

#### 3.1 Search Sessions

We used the interaction logs of over five million consenting Web users over a five-month period from May 2007 to September 2007. These logs were anonymized, and all personally identifiable information, including IP addresses, was removed. The logs gave us access to user interactions with all search engines. From these logs, we extracted *search sessions* that began with a query to Google, Yahoo!, or Live Search and terminated after 30 minutes of browsing inactivity.<sup>6</sup> A similar threshold has been used to demarcate search sessions in previous work on search engine switching [16] and in related studies of user search behavior [20,26]. These sessions are used to analyze switching behavior and give insight into the potential benefit of supporting switching.

#### 3.2 Overview of Switching Behavior

Our analysis showed that 36.4% of searchers used more than one search engine in the duration of the logs.<sup>7</sup> The findings also showed that 6.8% of all sessions and 12.0% of sessions containing more than one query involved a switch between two or more search engines. Although the aim of the paper is not to characterize the nature of search engine switching, a visual examination of search engine usage patterns in the logs revealed three salient classes of switching behavior: *within-session*, *between-session*, and *long-term*. We now describe these classes and provide summary statistics:

- **Within-session switching:** Users switch between Web search engines within a single search session and may use multiple engines concurrently. Such switches may be associated with a desire for topic coverage, dissatisfaction with any particular engine, and perhaps even automated applications that issue queries to multiple engines. Approximately 33.4% of the users in our sample exhibited this class of behavior.
- **Between-session switching:** Users switch engines for individual search sessions or groups of sessions. Switches of this nature may occur because a user feels that a particular engine is better suited for the current task due to an interface component or vertical supported. Approximately 13.2% of the users in our sample exhibited this type of switching behavior.
- **Long-term switching:** Users switch from one search engine to another and never return to the original engine. This appears to represent a change in their search engine preference. Approximately 7.6% of the users in our sample switched

<sup>4</sup> <http://www.clusty.com>

<sup>5</sup> <http://www.dogpile.com>

<sup>6</sup> At the time of writing, together these engines handle over 80% of worldwide Web search queries according to comScore.

<sup>7</sup> These users submitted five or more queries to at least two search engines. If we vary this threshold between one and ten queries the proportion of users that switch engines ranges between 54.0% and 26.7%.

search engines and never returned to their original engine in the duration of the study.<sup>8</sup>

Of these three classes, our component aims to support within-session switches, where it might be in a user’s interest to change search engines for the current query. While the above statistics demonstrate that search engine switching is a strategy employed by some users, the majority of users remain loyal to a single engine. Prior to describing our method for supporting search engine switching, the next section analyzes the potential benefit to users brought by utilizing multiple search engines.

### 3.3 Potential Benefit of Switching

To motivate our approach, we first quantify the potential benefit of multiple search engine use. That is, if a user is searching on a given engine, what is the likelihood that they would obtain better quality results if they were to issue the same query on a different engine. This is important, since encouraging users to switch when it is not in their interests to do so could lead to user dissatisfaction and ultimately distrust for our classifier.

To quantify the potential benefit from switching, we studied user search behavior in the interaction logs described in the previous section. We used two measures to evaluate engine performance for a given query: *relevance score* and *result click-through rate*:

- **Relevance score:** The Normalized Discounted Cumulative Gain (NDCG) [15] on each of the engines for a particular query. We can compute NDCG at different rank positions (*e.g.*, 1, 3, 10). In this paper, we elect to compute it at position three unless otherwise stated, since it captures the value of the top-ranked search results, which matter most to users.
- **Click-through rate:** The proportion of searches on an engine for a query that lead to a click on any of the returned search results. Users may fail to click on search results for a number of reasons that are not attributable to topical relevance (as measured through NDCG). Average click-through rate may give us a reasonable estimation of search result utility

From each of the search sessions described in Section 3.1, we extracted the queries that users issued. We identified a set of 4,921 queries that were submitted at least five times to each of the three engines in this study: Google, Yahoo!, and Live Search. These queries were originally drawn from a larger set of queries obtained by randomly sampling by frequency a one month query log of one of the search engines (*i.e.*, each query had a chance of being selected proportional to its frequency). For each of these queries, trained human assessors assigned judgments to result pages from the live Web (on a six point scale) based on their perceived relevance to the query. This judged set provided the basis for evaluation.

We computed the relevance scores (NDCG) and the click-through rates on all three engines for each of these queries. For each query in this set, we ranked the three engines based on the relevance score and their click-through rate to give us two independent rankings for each query. The direct comparison of quality between these three engines is beyond the scope of this paper. Nonetheless, in Table 1 we present the number (and percentage) of queries in our query set where each of the three engines – represented in random order as X, Y, and Z to preserve anonymity – outperformed the two other engines in terms of relevance and result click-through rate.

**Table 1. Number of queries for which engine performs best.**

Search engine	Relevance (NDCG)	Result click-through
X	<b>952</b> (19.3%)	<b>2,777</b> (56.4%)
Y	<b>1,136</b> (23.1%)	<b>1,226</b> (24.9%)
Z	<b>789</b> (16.1%)	<b>892</b> (18.1%)
No difference	<b>2,044</b> (41.5%)	<b>26</b> (0.6%)

These findings demonstrate that engine choice for a particular query is important, and that a classifier to help users select the most effective engine for each of their search queries is likely to improve a user’s overall search effectiveness, since all engines perform best at some subset of the query set.

To estimate the potential for supported search engine switching, we once again used the search sessions described in Section 3.1. We extracted all instances of the 4,921 queries from these sessions and computed the proportion of all query instances for which the relevance score or click-through rate were higher on an engine other than that selected by the user. Since it considers all query instances this more accurately captures the potential benefit of switching than the findings presented in Table 1. Results show that users could benefit from switching engine for around half of their queries (*i.e.*, click-through rate higher on alternate engine for 54.5% of query instances, relevance score higher on alternate engine for 52.3% of query instances). To ensure that we were not simply advocating a switch to a single dominant engine, we computed the distribution of search engines recommended across all query instances. This analysis showed that all three engines were recommended approximately equally, alleviating our concerns.

These results quantify the benefits of switching, demonstrating that any given engine performs best for at least some fraction of search queries. As loyalty and familiarity may discourage users from switching, our aim is to automatically determine when it is in users’ interest to try another search engine. The principal challenge for a generic solution to this problem lies in achieving real-time accurate performance for previously unseen queries. In the next section, we present our proposed machine learning methodology that utilizes features of the query, the results, and the titles/snippets/URLs of the top-ranked pages.

## 4. SUPPORTING SWITCHING

As the results in the previous section demonstrate, search engine switching was detected for around half of our five million users, and in 10% of all search sessions. The analysis shows that around 50% of all searches may have more accurate results if the query is issued on a different engine. Therefore, a user’s search experience could be enhanced if they were notified when an alternate search engine is likely to provide better results or different results of same quality, obviating the need to attempt the query on an alternate engine manually and broadening awareness of other engines.

Achieving this requires automatically detecting whether the results for the current query on an alternate engine are better (or equivalently good but different) than the results for the currently used engine. The following subsections describe our approach for solving this prediction problem.

### 4.1 Switching as Classification

Comparison of search result sets from any two engines can be modeled in several ways. One approach is to predict the quality of the results for each engine independently and subsequently compare the two scores. An alternative is to consider the two engines simultaneously, where the single prediction objective is to determine whether one engine produces results of better or equal quality

<sup>8</sup> The criterion for a long-term switch was a switch followed by no further queries on the prior engine. More relaxed variants would lead to the identification of more long-term switchers.

ty than the other engine. Since the underlying problem facing the user is a decision task based on the pair of result sets, this “coupled” approach is a more appropriate abstraction, and hence is the direction we pursue.

Modeling the difference in quality between two sets of search results can be viewed as a regression task (predicting the real-valued difference in quality between the two result sets), or as binary classification (where the prediction is equivalent to deciding whether switching to a different engine is worthwhile, without directly learning to quantify the expected difference in result quality). Among these options, binary classification is a more suitable choice, since it most closely mirrors the switching decision task. The actual utility of switching for a given user depends on such factors as the relative costs of interruption and benefits of obtaining better and/or different search results, which can be incorporated into the classification task via the concept of a *margin* in quality between the two result sets (by assigning “positive” labels to pairs of results sets where the difference in quality is above the minimum margin corresponding to switching utility).

Formally, let a given problem instance consist of a query  $q$  and two search engine result pages:  $R$  from the current search engine, and  $R'$  for an alternative search engine. Let query  $q$  have a human-judged result set  $R^* = \{(d_1, s_1), \dots, (d_k, s_k)\}$  consisting of  $k$  ordered URL-judgment pairs, where each judgment reflects how well the URL satisfies the information need expressed in the query, on a scale from 0 (*Bad*) to 5 (*Perfect*). The utility of each engine for the query can be represented as the NDCG score of the returned results set:  $U(R) = NDCG_{R^*}(R)$  and  $U(R') = NDCG_{R^*}(R')$ . Suppose that the user benefits from switching support if the alternative search engine provides utility that is higher by at least  $\epsilon \geq 0$ . Then, a dataset of queries  $Q = \{(q, R, R', R^*)\}$  yields a set of training instances,  $D = \{(x, y)\}$ , where each instance  $x = f(q, R_1, R_2)$  is comprised of features derived from the query and result pages as described in Section 4.2, and the binary label  $y$  encodes whether the alternative search engine provides performance that is higher than that for the current engine by at least  $\epsilon$ :  $y = 1$  iff  $NDCG_{R^*}(R') \geq NDCG_{R^*}(R) + \epsilon$ .

While any binary classifier can be used for this task, minimizing computational and memory costs is a key consideration for selecting an appropriate algorithm. Upon every search executed in the browser, the switching support framework must execute the same search on alternative engines in the background, subsequently computing features for the classifier, which then predicts whether alternative engines should be suggested. Furthermore, users’ interaction with the switching support system may provide additional training information for the classifier, which calls for classifiers that can be trained in online fashion, where learning is performed using a continuously incoming stream of instances with labels derived from user interaction (e.g., using such indicators of user satisfaction as click-through on the search results page or dwell time on result pages). In this work, we employ maximum-margin averaged perceptron [6] as the classifier, since it readily satisfies the constraints above and has previously shown excellent empirical performance in many domains from natural language to vision.

## 4.2 Features

For each query submitted to a current search engine, the classifier must predict in real-time whether the user would benefit from utilizing a different search engine based on features derived from the query and the two sets of results from the two engines. Thus, features can be separated into three broad categories: (i) features derived from the two result pages, (ii) features based on the query, and (iii) features based on the matching between the query and the

results page. The subsequent sections describe each of these feature sets in detail, while Table 2 provides a comprehensive list of all features. We employ only generic text-based features that can be obtained for any search result page; the space of features was determined before running any experiments, and we did not perform any feature selection. In Section 5.1.3, we measure the utility of each feature category to determine their relative contributions to the classification task.

### 4.2.1 Results Page Features

Each engine’s result page contains a ranked list of search results, where each result is described by a title, a snippet (a short summary), and its URL. The results page features capture the following properties of each result:

- Textual statistics for the title, URL, and the snippet, such as the number of characters, number of tokens, number of ellipses, etc.
- Properties of the URL, such as whether it comes from a .com or .net domain, whether the page has a .html or .php file extension, the number of directories in the URL path, presence of special characters, etc.

Furthermore, there are features of the results page not captured by the result lists themselves. For example, search engines typically inform the user how many total pages in their index contain the given query terms (e.g., “Results 1-10 of 64,500”). This number is also a feature. Other features encode such results page properties as whether spelling correction was engaged, features of any query-alteration suggestions, and features based on any advertisements also found on the page.

### 4.2.2 Query Features

Different search engines may have ranking algorithms that perform particularly well (or particularly poorly) on certain classes of queries. For example, one engine may focus on answering rare (“long-tail”) queries, while another may focus on common queries. Thus, features can also be derived from query properties, such as the length of the query, presence of stop-words (common terms like “the”, “and”, etc.), presence of named entities, etc.

### 4.2.3 Match Features

We designed the third set of features to capture how well the results page matches the query. For example, these features encode how often query words appear in the title, snippets, or result URLs, or how often does the entire query, or bigrams within the query appear in these segments. Since search engines attempt to create a snippet that represents the most relevant piece of a document, one expects that snippets that contain many matches of the query are indicative of a relevant result, while few or no matches likely correspond to a less relevant result.

### 4.2.4 Higher-Order Features

Following a common practice in machine learning applications, we provide non-linear transforms of each feature to the learner, so it can directly utilize the most appropriate feature representation. In this paper, we add the logarithm and the square of each feature value as two additional features. Another group of meta-features are based on combinations of feature values for the two engines, e.g., a binary feature indicating whether the number of results that contain the query is at least 50% greater in the alternative engine than in the current engine. Note that simple differences between features (e.g., the number of results on the alternate engine minus the number on the current engine) are unnecessary, as the perceptron can model such features by assigning a higher positive weight

**Table 2. Features employed in classification.**

Results Page Features
10 binary features indicating whether there are 1-10 results
Number of results
For each title and snippet:
# of characters
# of words
# of HTML tags
# of “...” (indicate skipped text in snippet)
# of “. ” (indicates sentence boundary in snippet)
# of characters in URL
# of characters in domain (e.g., “apple.com”)
# of characters in URL path (e.g., “download/quicktime.html”)
# of characters in URL parameters (e.g., “?uid=45&p=2”)
3 binary features: URL starts with “http”, “ftp”, or “https”
5 binary features: URL ends with “html”, “aspx”, “php”, “htm”
9 binary features: .com, .net, .org, .edu, .gov, .info, .tv, .biz, .uk
# of “/” in URL path (i.e., depth of the path)
# of “&” in URL path (i.e., number of parameters)
# of “=” in URL path (i.e., number of parameters)
# of matching documents (e.g., “results 1-10 of 2375”)
Query Features
# of characters in query
# of words in query
# of stop words ( <i>a, an, the, ...</i> )
8 binary features: Is $i^{th}$ query token a stopword
8 features: word lengths (# chars) from smallest to largest
8 features: word lengths ordered from largest to smallest
Average word length
Match Features
For each text type (title, snippet, URL):
# of results where the text contains the exact query
# of top-1, top-2, top-3 results containing query
# of query bigrams in the top-1, top-2, top-3, top-10 results
# of domains containing the query in the top-1, top-2, top-3

to the first component of the difference, and a higher negative weight to the second.

These features can all be computed at run time and are all readily available with minimal overhead, and are only a subset of all features that could be used. If efficiency constraints were relaxed, the feature set could be enhanced to leverage the hyperlink structure of top documents (as done in [17]), search result clickthrough logs, and search engine response times, among many others.

## 5. EVALUATING PERFORMANCE

As demonstrated by the analysis in Section 3, no matter what search engine is employed by a given user, there are always some queries for which other engines provide better results. The objective of providing switching support is to identify such queries automatically using the machine learning methodology described in Section 4. Therefore, our goal is to evaluate the accuracy of the proposed switching support mechanism independently of popularity or absolute accuracy of individual search engines to assess the viability of recognizing queries for which an alternative engine provides better performance.

### 5.1 Dataset and Methodology

To evaluate the proposed approach for recognizing queries for which switching search engines is beneficial, we employ a labeled corpus of queries randomly sampled from search engine logs. For

each query, a panel of human judges evaluated several dozen top-ranked results returned by the three most popular search engines, assigning them relevance scores on a six-point scale that range from *Bad* to *Perfect*. Human evaluation is performed without any information that may identify engines to remove any individual biases that judges may have. Table 3 below provides some summary statistics for the labeled dataset.

**Table 3. Query dataset properties.**

Total number of queries	17,111
Total number of judged pages	4,254,730
Total number of judged pages labeled <i>Fair</i> or higher	1,378,011

Given the labeled dataset, the quality of results returned by search engines for each query can be evaluated by computing NDCG against the human judgments as described in Section 3.3. To evaluate the machine learning approach to switching support described in Section 4.1, we transform the corpus of queries, judgments and search engine results into multiple labeled datasets of feature vectors and labels as described in Section 4.2. For every pair of search engines and any predefined margin  $\epsilon$  in NDCG scores required to justify switching for the user, the sampled dataset includes an equal number of positive and negative instances corresponding to queries for which switching is beneficial, and queries for which it is not.

Classification experiments are performed using 10-fold cross-validation by separating the dataset for every pair of search engines into ten folds of equal size, and repeatedly computing accuracy on one (testing) fold after training on the remaining folds. The process is repeated over 100 runs with randomized fold assignment.

There are fundamental trade-offs between recall, interruption, and error cost to the user that switching support must address. If the confidence threshold is low, the user will be informed of possibly better results provided by the alternative engine more frequently, however some suggestions may be erroneous, which coupled with the increased interruption cost is likely to upset the user. Therefore, it is preferable to interrupt the user less frequently, while providing high-accuracy suggestions. Evaluation can reflect these considerations by employing precision-recall curves in place of single-point accuracy measurements, where precision and recall are defined as the proportion of true positives (queries for which switching is desirable) among (1) all predicted positives for precision, and (2) all true positives for recall. We construct precision-recall curves by varying the confidence threshold of the classifier, starting with a high value, where switching is advised in very few cases, resulting in high precision (few erroneous suggestions) but low recall. Through lowering the confidence threshold, it is possible to suggest switching for more queries at the cost of more errors and increased interruption to the user.

#### 5.1.1 Precision-Recall Results

Figure 1 shows the precision-recall curves that summarize the performance of the classifier-based approach with respect to NDCG@3 with  $\epsilon = 0$  (in other words, equally or more accurate but different results on a different engine comprise a positive example for predicting switching). These results demonstrate that the proposed approach can attain very high precision at lower recall levels, which are most important if the costs of user interruption are viewed as non-negligible. Precision decreases sharply at higher recall levels, eventually dropping to the random prior, which is above 50% because with  $\epsilon = 0$ , queries on which engines obtain equally accurate but different results are viewed as

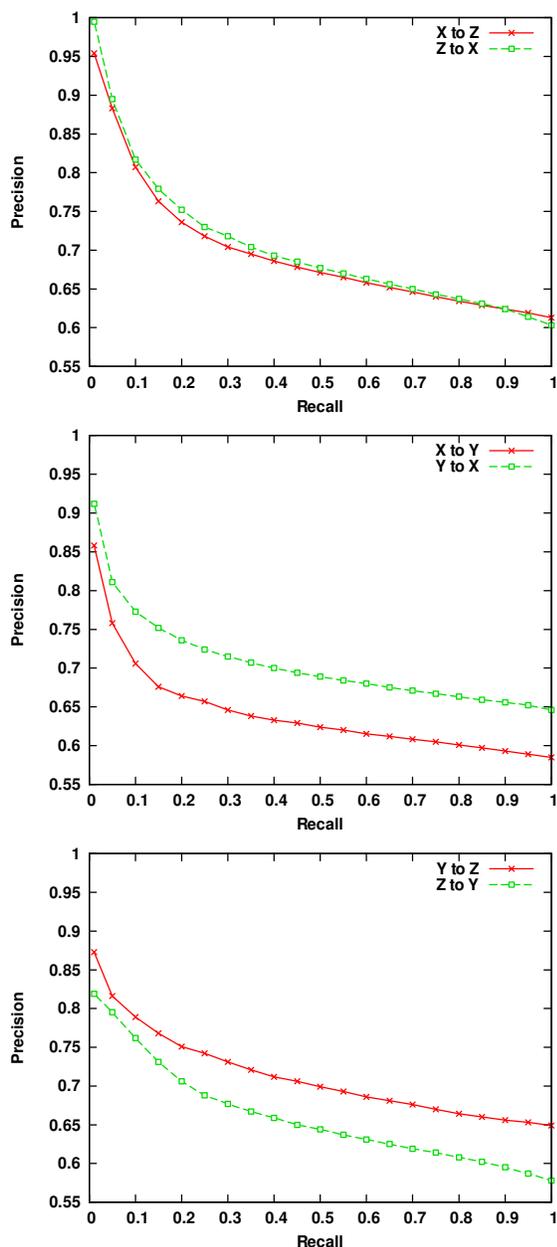


Figure 1. Prediction accuracy for different engine pairs.

switch-worthy, since the alternative search engine provides the user with novel results of equal quality.

The sharp decline in precision at higher recall levels demonstrates that discriminating between search engines using only the query and their result pages is a very difficult learning task. However, since the goal is to only suggest alternative search engines when they are likely to provide additional value over the current search engine, high performance at low recall levels is still highly valuable as it allows the provision of accurate suggestions to the user for a number of queries, while not interrupting them too often. Table 4 summarizes precision at recall of 0.05 for all engine pairs.

These results demonstrate that the machine learning approach we propose for supporting search engine switching can achieve high accuracy, and therefore can be used for providing useful search engine suggestions to users. The table shows that there are significant distinctions in performance between different engine pairs:

Table 4. Summary of precision at recall=0.05.

		To		
		X	Y	Z
From	X		0.758	0.883
	Y	0.811		0.816
	Z	0.860	0.795	

*e.g.*, performance is much higher when identifying queries on which users of engine X will benefit when switching to engine Z versus switching to engine Y. These differences are caused by two factors: (i) the degree to which ranking algorithms employed by the engines differ, and (ii) the prior probability of obtaining better performance on the alternate engine when switching from the given default engine.

Because both of these factors can be controlled by varying the margin parameter,  $\epsilon$ , which specifies the minimum difference in result quality considered acceptable for providing the user with a switching suggestion. We investigate the effect that  $\epsilon$  has on accuracy by varying the value of  $\epsilon$ , and correspondingly changing the classification task to have fewer/more positive examples. Figure 2 demonstrates the precision values at 0.05 recall averaged over all search engine pairs, for different values of  $\epsilon$  alongside the prior probability of obtaining better results on the alternate engine. The two values at  $\epsilon = 0$  denote either labeling queries on which the engines produce different but equally accurate results as positive (switching is beneficial for novel results), or negative (switching is only desired for higher-quality results).

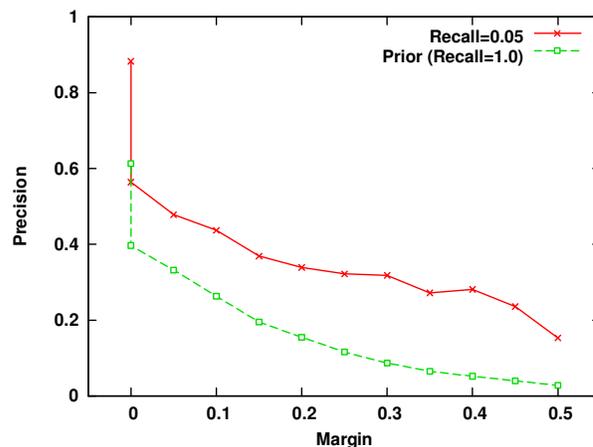


Figure 2. Prediction accuracy for different margin values.

These results demonstrate that while our approach is able to improve over the baseline for all margin values, the task becomes significantly harder for larger margin values, since the number of queries for which one engine is better than another by a large margin decreases with margin size.

### 5.1.2 Avoiding Querying the Alternate Engine

So far, we have seen that we can, with reasonably high precision, suggest alternative search engines to users for appropriate queries. Doing so requires not only analyzing the content of the current result page, but also querying the alternate search engine in the background. For some users and/or search engines, the resulting extra network traffic may be undesirable. One way to avoid this is by classifying whether a switch would be beneficial, but using only features based on the current engine's result page. The results for this are given in Figure 3, averaged across both alternate engines for each of X, Y, and Z.

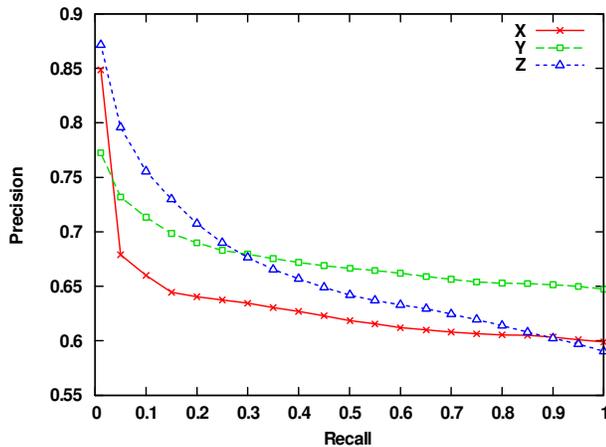


Figure 3. Prediction accuracy using only one engine's features.

As expected, the accuracy is lower than when using the results from both engines. Whether the networking cost to the user and the alternate search engine is worth the boost in accuracy is an empirical question that would have to be answered by the particular user and search engine in question. One interesting approach would be to use the single-engine classifier as a filter to exclude queries least likely to be served better by the alternate engine.

### 5.1.3 Contribution of Features

In order to better understand the utility of various features to the overall task performance, we conducted an ablation study in which we removed each of the three feature sets, retrained the classifier, and observed the decrease in performance. In Figure 4, we show the results of these ablations. Results page features are denoted as R, query-based features are denoted as Q, and match features are denoted as M. As can be seen from these results, every set of features is contributing to the overall accuracy to some degree. However, it is clear from the figure that features obtained from results pages are providing the most benefit, since performance decreases most substantially when they are removed.

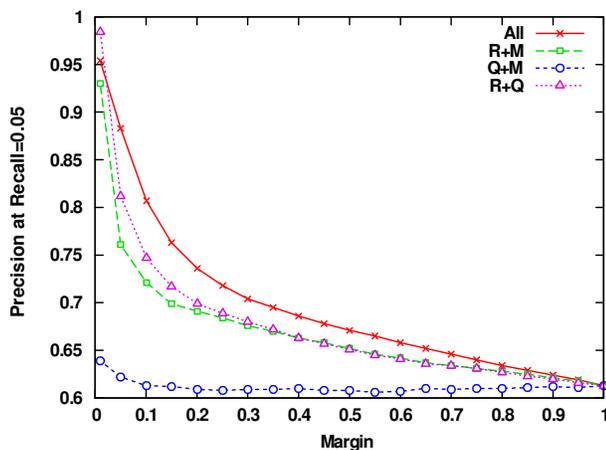


Figure 4. Performance with reduced feature sets.

We also tested the performance of the classifier when it is provided with only one feature group at a time; results of these experiments are shown in Figure 5, averaged across all search engine pairs. Confirming the ablation study, we again observe that features based on results pages yield most accurate predictions among the three groups.

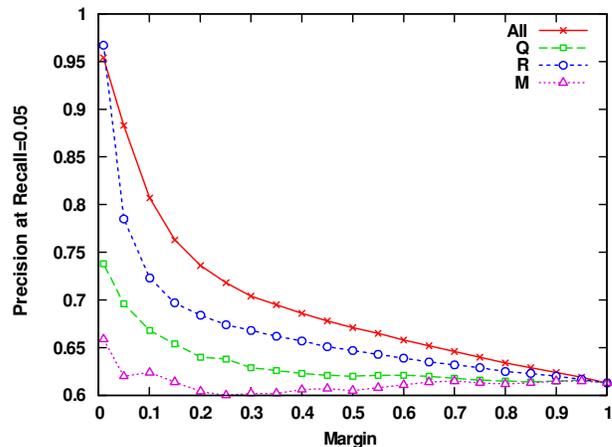


Figure 5. Performance of individual feature groups.

The features we used were designed to be efficient in terms of computational and memory requirements, so there is little to gain by removing any of them from the classifier, especially given that the above results demonstrate that each of the feature groups improves performance. The analysis here mostly serves as a guide for investigating new features, and while most benefit comes from analysis of the results page itself, investigating the utility of additional, possibly server-based features, is interesting future work.

## 6. DISCUSSION AND IMPLICATIONS

We have described a method for automatically determining when user's interests would be best served by switching engines for a given query. The precision and recall values provided in the previous section demonstrate that it is possible to accurately predict whether another engine has better quality results based solely on features of the query and search results from alternate engines. Additional analysis revealed that if an oracle switcher could perfectly predict which engine had better results for the queries in our test set, a ten-point gain in NDCG would be observed. This improvement would significantly impact users' search effectiveness.

The machine learning framework we proposed in this paper could be implemented as a browser plug-in that would notify users in real time when they should consider switching engines. The tool would alert users whenever another engine could provide a better set of search results, or when the user appeared dissatisfied with the current result set (per negative interaction behaviors such as requesting the next page of search results, not clicking on any results, or reformulating their current query). It is important to emphasize that our approach can be implemented completely client-side without the need for server-side link-graphs or log-based information that would make meeting the real-time performance constraints difficult.

As part of the study of switching behavior described earlier, we identified three classes of search engine switching: within-session, between-session, and long-term. Although our emphasis has been on supporting users who may be willing to switch between engines within a single search session, it is also important to consider how to support users in selecting a different engine for different search sessions. Automatic detection of search task intent and switches between search tasks has been studied extensively in the human factors community [3,9], leaving it as an exciting challenge for future research to develop techniques that would provide users with the best multi-engine support at search task level.

While we have found that for approximately half of all searches users could retrieve more accurate search results if they switched

search engines, we did not impose any constraints on the accuracy margin beyond analyzing performance for different margin values in Section 5.1.1. Thus, we established an upper bound for accuracy improvements enabled by switching engines. Previous work has shown that users may not notice small differences in quality of search results, even though these have been detected by evaluation metrics [25]. We hope to investigate the relative benefits of accuracy increases versus the cost of user interruption in future user studies, so that our methodology could provide maximum value to users. As well as predicting the existence of higher-quality search results on alternative engines, factors that must be considered include understanding users' focus of attention, workload, and willingness to be interrupted, so as to present recommendations at an appropriate time [14].

It is worth noting that the objective measures of switching utility do not consider the additional cognitive burden and associated temporal costs on users of this activity. Web search engines exhibit differences in their user interfaces, the query syntax they support, and the collection of Web pages they index. These distinctions may adversely affect users' ability to locate relevant information when changing engines. Further work is required to understand the cognitive costs to users in multiple search engine use.

The research described in this paper has shown that it is possible to facilitate switching between search engines in real-time; the next step is develop methods that will make the transition between engines maximally smooth for the user. We hope that future user studies will help to evaluate the performance of the classifier with human subjects engaged in realistic task scenarios and quantify the above factors in computing switching utility.

## 7. CONCLUSIONS

In this paper, we advocated for the use of multiple search engines to empower users to search more effectively. We described a log-based study of Web search behavior with a particular emphasis on multiple search engine use, which demonstrated that search engine switching can substantially improve retrieval effectiveness. We proposed a machine learning-based approach for supporting switching that estimates in real time whether more accurate results exist on alternate search engines. Estimation is based on features of the query, the result set, and the titles, snippets, and URLs of the top-ranked search results. An empirical analysis of classification performance demonstrates that it is accurate at predicting when users would benefit from switching between engines at low recall levels. The promotion of multiple search engine use through application components such as that described has the potential to improve the retrieval experience for users of all search engines.

## 8. ACKNOWLEDGEMENTS

We wish to thank Resa Roth for her invaluable support and constructive comments on drafts of this paper.

## 9. REFERENCES

- [1] Agichtein, E., Brill, E. & Dumais, S. (2006). Improving web search ranking by incorporating user behavior information. In *Proc. SIGIR*, 19–26.
- [2] Buckley, C. & Walz, J. (1999). Overview of the TREC-8 query track. In Voorhees, E.M. & Harman, D.K. (Eds.) *Proc. TREC-8*, 65-76.
- [3] Budzik, J. & Hammond, K.J. (2000). Users interactions with everyday applications as context for just-in-time information access. In *Proc. IUI*, 44-51.
- [4] Callan, J.P., Lu, Z. & Croft, W.B. (1995). Searching distributed collections with inference networks. In *Proc. SIGIR*, 21-28.
- [5] Clarke, C., Agichtein, E., Dumais, S. & White, R.W. (2007). The influence of caption features on clickthrough patterns in web search. In *Proc. SIGIR*, 135-142.
- [6] Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proc. EMNLP*, 1-8.
- [7] Craswell, N., Hawking, D. & Thistlewaite, P. (1999). Merging results from isolated search engines. In *Proc. Australasian Database Conference*, 189-200.
- [8] Cronen-Townsend, S., Zhou, Y. & Croft, W.B. (2002). Predicting query performance. In *Proc. SIGIR*, 299-306.
- [9] Czerwinski, M., Horvitz, E. & Wilhite, S. (2004). A diary study of task switching and interruptions. In *Proc. SIGCHI*, 175-182.
- [10] Gordon, M. & Pathak, M. (1999). Finding information on the world wide web: The retrieval effectiveness of search engines. *Inf. Proc. Manage.* 35(2): 141-180.
- [11] Harman, D. (1993). Overview of the first text retrieval conference. In *Proc. SIGIR*, 36-47.
- [12] Hawking, D., Craswell, N., Bailey, P. & Griffiths, K. (2001). Measuring search engine quality. *Inf. Ret.*, 4: 33-59.
- [13] Heath, A.P. & White, R.W. (2008). Defection detection: Predicting search engine switching. In *Proc. WWW*, in press.
- [14] Horvitz, E. & Apacible, J. (2003). Learning and reasoning about interruption. In *Proc. ICMI*, 20-27.
- [15] Järvelin, K. & Kekäläinen, J. (2000). Information retrieval evaluation methods for retrieving highly relevant documents. In *Proc. SIGIR*, 41-48.
- [16] Juan, Y.-F. & Chang, C.-C. (2005). An analysis of search engine switching behavior using click streams. In *Proc. WWW*, 1050-1051.
- [17] Leskovec, J., Dumais, S.T. & Horvitz, E. (2007). Web projections: Learning from contextual subgraphs of the web. In *Proc. WWW*, 471-480.
- [18] Mukhopadhyay, T., Rajan, U. & Telang, R. (2004). Competition between internet search engines. In *Proc. HICSS*, p. 80216a.
- [19] Pew Internet & American Life Project. (2005). *Search Engine Users*. Accessed October 16, 2007. Available at: [http://www.pewinternet.org/pdfs/PIP\\_Searchengine\\_users.pdf](http://www.pewinternet.org/pdfs/PIP_Searchengine_users.pdf)
- [20] Pitkow, J.E. & Pirolli, P. (1999). Mining longest repeating subsequences to predict world wide web surfing. In *Proc. USENIX Symposium*, 139-150.
- [21] Rasolofo, Y., Abbaci, F. & Savoy, J. (2001). Approaches for collection selection and results merging for distributed information retrieval. In *Proc. CIKM*, 191-198.
- [22] Rorvig, M. (2000). A new method of measurement for question difficulty. In *Proc. ASIS*, 372–378.
- [23] Si, L. & Callan, J. (2003). A semi-supervised learning method to merge search engine results. *TOIS*, 21(4). 457-491.
- [24] Telang, R., Mukhopadhyay, T. & Wilcox, R. (1999). An empirical analysis of the antecedents of internet search engine choice. In *Proc. Workshop on Information Systems and Economics (WISE, Charlotte NC)*.
- [25] Turpin, A. & Hersh, W. (2001). Why batch and user evaluations do not give the same results. In *Proc. SIGIR*, 225-231.
- [26] White, R.W. & Drucker, S.M. (2007). Investigating behavioral variability in web search. In *Proc. WWW*, 21-30.
- [27] Yom-Tov, E., Fine, S., Carmel, D. & Darlow, A. (2005). Learning to estimate query difficulty. In *Proc. SIGIR*, 512-519.