Patterns of Query Reformulation During Web Searching

Bernard J. Jansen and Danielle L. Booth
College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16802. E-mail: jjansen@acm.org; nephari@gmail.com

Amanda Spink
Faculty of Information Technology, Queensland University of Technology, Gardens Point Campus, 2 George Street, GPO Box 2434, Brisbane QLD 4001, Australia. E-mail: ah.spink@qut.edu.au

Query reformulation is a key user behavior during Web search. Our research goal is to develop predictive models of query reformulation during Web searching. This article reports results from a study in which we automatically classified the query-reformulation patterns for 964,780 Web searching sessions, composed of 1,523,072 queries, to predict the next query reformulation. We employed an n-gram modeling approach to describe the probability of users transitioning from one query-reformulation state to another to predict their next state. We developed first-, second-, third-, and fourth-order models and evaluated each model for accuracy of prediction, coverage of the dataset, and complexity of the possible pattern set. The results show that *Reformulation* and *Assistance* account for approximately 45% of all query reformulations; furthermore, the results demonstrate that the first- and second-order models provide the best predictability, between 28 and 40% overall and higher than 70% for some patterns. Implications are that the n-gram approach can be used for improving searching systems and searching assistance.

Introduction

Web studies have focused on *query reformulation* (also known as *query expansion* and *query modification*) to assist users in locating relevant information. Query reformulation is the process of altering a given query to improve search or retrieval performance. Prior work has shown that effective query reformulation can improve the outcome of user searches (Gauch & Smith, 1993; Rieh & Xie, 2006). For example, Belkin et al. (2003) reported that query-reformulation assistance may be helpful and improve searching performance, and researchers have reported success with query-expansion methods (cf. Fonseca, Golgher, Pôssas, Ribeiro-Neto, & Ziviani, 2005; Fonseca, Golgher, De Moura, & Ziviani, 2003). When implemented in real systems, perhaps unfortunately, users seldom utilize this system support, resulting in ineffective and inefficient searches (Anick, 2003).

Some researchers have attempted contextual help to assist in query reformulation (Meadow, Hewett, & Aversa, 1982b); however, users can become frustrated with information that is pushed to them by the system (Adamczyk & Bailey, 2004). One issue hindering the use of these contextual help systems may be a lack of understanding about when users desire system intervention. The cognitive load of information seeking and processing in complex contextual situations is high (Belkin, Oddy, & Brooks, 1982). The retrieval or interjection of assistance into the search process may be too much of a cognitive load, requiring a task switch from focusing on the search process to mentally processing the intervention. Therefore, the searcher may simply ignore any assistance offered.

What if the information system could more accurately predict what type of query reformulation the user was most likely to implement? What if the system could tell when the user was most open to system intervention? What if the system then could offer targeted query-reformulation assistance at the most receptive point in the search process? These questions motivate our research.

In the following sections, we first review prior work about searching patterns with a focus on query-reformulation literature, and we present our research questions. We then explain our use of a Web search engine log to investigate query-formulation patterns using n-grams and present our results. We end the article with implications for system design for Web searching.

Related Studies

In examining the searching process, various researchers have used the concept of states to model the sequence of

Received November 2, 2008; revised February 13, 2009; accepted February 14, 2009

 • Published online 25 March 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.21071

user–system interactions (Choo, Detlor, & Turnbull, 1998). These researchers have typically identified user actions on an information searching system and then classified these actions into states (cf. Penniman, 1975; Qiu, 1993). With this information, one then can build a state map or matrix of possible moves. Each pattern is a sequence of state changes. This use of states and transitions is a stochastic process from which one can compare patterns of various lengths to test the significance (i.e., to determine what length of pattern predicts arrival at a certain state).

Such stochastic processes are established as effective methods for analyzing users' searching patterns. Penniman (1975), for example, used this method to examine search–response patterns on a bibliographic database system. Penniman (1975) defined 11 states, merging them into four categories. He reported differences in both zero- and first-order models when users searched different databases as well as different search behaviors of novices and experienced users. Later, Penniman (1982) compared findings from various database systems reporting that session length is one of the variables that characterize user behavior and that the frequency distribution of usage patterns follows approximately a Zipfian curve (e.g., nearly 80% of activities are accounted for by about 20% of the activity types). Chapman (1981) used this method to compare groups of searchers based on group characteristics by constructing zero- through fourth-order models for each participant group and statistically testing for intergroup differences. Chapman reported that the lower order models appear to account for most of the differences in the higher order models. Tolle (1984) used the method to describe the use of various online catalogs. Tolle and Hah (1985) used the technique to compare the use of National Library of Medicine databases, building high-order transition Markov models. The researchers did not report which order model was best. Marchionini (1989) used the state transition approach to investigate the searching behavior of children using an electronic encyclopedia. Based on analysis of search patterns, the researcher found that novices used a heuristic and highly interactive search strategy. Using transition matrix analyses, Marchionini showed that younger searchers usually favored query-refining moves while older searchers favored title- and text-examination moves.

This line of state-based research focused primarily on the search actions and system responses (i.e., to what page of the system the searcher navigated). These studies did not develop an algorithmic approach to classify user actions at a more granular level than interactions (i.e., submit query, view result page, click result). Although addressing interaction patterns, these research studies did not investigate whether significant state-transition length can predict the next state.

Two exceptions to this last point are research studies conducted by Qiu (1993) and H.-M. Chen and Cooper (2002). Qiu investigated searching patterns in a hypermedia environment, and identified eight search states and conducted empirical testing of state-transition behavior. The investigator showed that a second-order Markov process best modeled the online-searching patterns. This means that the probability of arriving at a certain state depends only on the preceding two states. Qiu also reported that the second-order Markov model held for a variety of control variables. Qiu focused on user–system interactions patterns, however, and not specifically query reformulations. A Markov process must meet certain conditions, especially in the case of order; namely, homogeneity, stability, and order. However, Qui ran statistical tests to determine the difference among various ordered chains.

H.-M. Chen and Cooper (2002) conducted state-transition analysis, defining a state as a certain address of the viewed page. These researchers clustered users into six groups based on patterns of the states (H.-M. Chen & Cooper, 2001). Using six clusters of usage patterns, H.-M. Chen and Cooper (2001) showed that there were statistical differences among the groups. In related research, H.-M. Chen and Cooper (2002) used 126,925 sessions from an online library system, modeling access patterns using Markov models. In that study, the researchers found that a third-order Markov model explained the majority of the user clusters; they reported that a third-order model describes five of the groups, and a fourth-order model describes the remaining one cluster.

The Markovian approach also has been used for a variety of other studies in the Web searching and browsing areas. Spink (1997) examined search modifications in search terms and feedback states, and found that a first-order Markov process provided the best description of the data. Su, Yang, and Zhang (2000) investigated n-order models utilizing path profiles of users from a Web log to predict the users' future page requests. Shen, Dumais, and Horvitz (2005) used a Markov model for inferences of searcher topic interests by using visited uniform resource locators (URLs). M. Chen, LaPaugh, and Singh (2002) employed a user's history and frequency of access to predict future page requests. Lau and Horvitz (1999) manually tagged search engine queries using temporal boundaries and a Bayesian network to predict user query-reformulation patterns; however, a Bayesian network does not account for cyclic patterns (i.e., a searcher returning to a previously visited state). Zhang and Nasraoui (2007) showed that combining implicit search with Markov models was an effective design technique for a recommender system.

The aforementioned research studies focused primarily on the search or browsing actions and system responses and did not focus on query reformulation, which is a key area of research given that the query is the primary (albeit inexact) expression of the user's need (Croft & Thompson, 1987).

In our research, we focus on the state transitions as users reformulate their queries during a session. Query reformulation has been an active area of research in the information searching and retrieval areas given that the query is the primary expression of the searcher's information need. Rieh and Xie (2006) also conducted a qualitative analysis of query reformulation using 313 sessions from a Web search engine log. The researchers reported three facets of query reformulation (content, format, and resource), with multiple subfacets of each of these given areas. He, Göker, and Harper (2002), in automatically classifying query reformulation, used

DOI: 10.1002/asi

contextual information from a Reuters transaction log for analysis of Web sessions. Employing a version of the Dempster–Shafer theory in an attempt to identify search engine session boundaries, the researchers also identified a series of query states to detect the session start and end states; however, the researchers' focus was on determining the average Web user session duration rather than mapping query-reformulation states. He et al. automatically tagged queries, but they did not investigate the prediction of moving from one state to another within a session.

In other research, Özmutlu and Çavdur (2005) attempted to duplicate the findings of He et al. (2002), but Özmutlu and Çavdur reported that there were issues relating to implementation, algorithm parameters, and fitness function. In parallel and follow-up research, Özmutlu, Çavdur, Spink, and Özmutlu (2004, 2005) and Özmutlu and Çavdur (2005) investigated the use of neural networks to automatically identify topic changes of queries within sessions, reporting relatively high percentages (72–97%) of correct identifications of topic shifts and topic continuations. Özmutlu, Çavdur, Spink, and Özmutlu (2005) reported that neural networks were effective at topic identification, even if the neural network application was trained with data from another search engine transaction log. This line of research involved the use of sophisticated algorithmic approaches and extensive amounts of training data for identification of query reformulation. Even so, the approaches were all primarily descriptive in nature. For our research, we were interested in methods where one can make predictions of future user query-reformulation states.

In attempting to develop a query-expansion algorithm based on related sessions, Huang, Chien, and Oyang (2003) noted some interesting observations concerning user sessions. First, these researchers noted that the query length, measured in terms, is longer at the end of a session relative to the beginning. They further noted that the query terms in the beginning of the sessions were more general than those at the end of the sessions. This suggests that these users go through a process of query reformulation to narrow their information need and that there may be a correlation between longer queries and more specific information expressions.

In summary, this line of research (H.-M. Chen & Cooper, 2001, 2002; Qiu, 1993) has illustrated that the use of state transitions can be an effective methodological approach for modeling user actions; however, there has been little use of this method for modeling and drawing inferences for query reformulations. In the current research, we employ a state-transition approach to model query reformulation during a searching session to predict with some degree of accuracy based on a probabilistic model the user's next query reformulation. With this knowledge, one can design systems to provide more tailored query-reformulation assistance. The next section outlines our research questions, followed by an explanation of the research design and methods. This study is a continuation of research by Jansen, Spink, Blakely, and Koshman (2007), who explored methods for defining Web session boundaries.

Research Questions

The following research questions are addressed in this study:

RQ1: What is the distribution of search states of query reformulations during Web searching?

For RQ1, using a transaction log from a Web search engine (i.e., Dogpile), we developed heuristics to classify each query into one of six unique query-reformulation states, implemented these heuristics in a program, and executed this program against the entire transaction log. With these results, we then could show the distribution of query-reformulation states of the entire dataset.

RQ2: What states are most likely to follow one another in Web searching?

For RQ2, we used the results from RQ1 to develop a probability transition matrix, which provides the percentage of transitions among each of the six query-reformulation states.

RQ3: What order of state transition provides the best predictability for query reformulation during Web searching?

For RQ3, we used an n-gram approach to determine what order of states provides the best prediction of future states. We were interested in how much session history for a particular user the system would need to predict with an acceptable degree of certainty the user's next query reformulation state. Building off our probability transition matrix, we measure the probability of transitioning through a sequence of states.

RQ4: When are users most receptive to system assistance?

For RQ4, we investigated the use of searching assistance by users. The Web search engine used in this research offers a query-reformulation feature. By analyzing our state-transition patterns, we could detect when in the query-reformulation sequence users sought out assistance from the system. We assume that at this point in the searching process, users would be most receptive to some form of automated assistance (Jansen, 2006) or contextual help (Meadow, Hewett, & Aversa, 1982a; Xie & Cool, 2009) for query reformulation.

We discuss our research design in more detail in the following sections.

Research Design

Web Data

For this research study, we collected data from the Dogpile (http://www.dogpile.com/) meta-search engine. A search engine within the Infospace network, Dogpile integrates the results from four leading Web search indices (i.e., Ask, Google, MSN Live, and Yahoo!) along with results from 18 other search engines into an integrated search results listing. Dogpile.com provides indices for searching *Web*, *Images*, *Audio*, and *Video* content using tabs on the search engine interface. In addition to spelling suggestions, Dogpile also offers query-reformulation assistance with alternate query suggestions listed in an *Are You Looking for?* area of the interface. The interested reader can visit http://www.dogpile.com for an illustration of the interface with query box, tabbed indices, and the *Are You Looking for?* feature.

In terms of generalizability of the dataset, Jansen and Spink (2005) showed that Web searchers exhibit similar searching characteristics across search engines from an analysis of nine Web search engine transaction logs. Jansen and Spink (2005) also demonstrated that Web searcher interactions are consistent across days and search engines, with the exception of the specific term usage. Reports from other studies of Web search engines have reported similar characteristics (Park, Bae, & Lee, 2005; Silverstein, Henzinger, Marais, & Moricz, 1999; Wang, Berry, & Yang, 2003). Therefore, we believe the data sample is representative of not only Dogpile users but also the larger Web searching population.

Data Collection and Preparation

On May 6, 2005, we collected records of Web searcher–system interactions in a transaction log that represents a portion of the searches executed on Dogpile.[1] The terminology and procedure that we used in this research is similar to that used in other Web transaction log studies (cf. Jansen & Pooch, 2001; Park et al., 2005). The original transaction log contained 4,056,374 records, with each record containing seven fields:

- *User Identification:* a code to identify a particular computer based on the computer's Internet Protocol address.
- *Cookie:* an anonymous cookie automatically assigned by the Dogpile.com server to identify unique users on a particular computer based on a browser.
- *Time of Day:* measured in hours, minutes, and seconds as recorded by the Dogpile.com server on the date of the interaction.
- *Query Terms:* the terms exactly as entered by the given user.
- *Source:* the content collection that the user selects to search (e.g., *Web*, *Images*, *Audio*, *News*, or *Video*), with *Web* being the default.
- *Feedback:* a binary code denoting whether the query was generated by the *Are You Looking for?* query-reformulation assistance provided by Dogpile.com

Once we had recorded the data, we imported the original flat ASCII transaction log file of 4,056,374 records into a relational database and generated a unique identifier for each record. We used four of the fields in the search log (*Time of Day*, *User Identification*, *Cookie*, and *Query*) to locate the initial query and then to recreate the temporal sequential series of interactions of a particular user. The fields *User Identification* and *Cookie* determined a user or, more correctly, a given computer browser. Naturally, there is no guarantee that one and only one person is using said browser. An analysis of the dataset shows that the interactions of Dogpile searchers were generally similar to Web searching on other Web search engines (Jansen, Spink, Blakely, & Koshman, 2006).

For this research, we were interested in queries submitted by humans, and the transaction log contained queries from both human users and agents. In prior published work, researchers used either a temporal or interaction cutoff for identifying human from nonhuman submissions in a search log (Silverstein et al., 1999). For this research, we selected the interaction cutoff approach by removing all sessions with 100 or more queries. Since this cutoff is substantially greater than the reported mean number of queries for human Web searchers (Silverstein et al., 1999), it increased the probability that we were not excluding any human searches. Although this cutoff most likely introduced some agent sessions, we wanted to be reasonably certain that we had included most of the queries submitted by human searchers.

For this research, we define the following key concepts:

- *Term:* a series of characters within a query separated by white space or other separator.
- *Query:* a string of terms submitted by a searcher in a given instance of interaction with the search engine.
- *Session:* a series of queries submitted by a user and related interactions during an episode of interaction between the user and the Web search engine around a single topic.
- *Search Episode:* one or more sessions by an individual user within a given period.
- *Query Reformulation:* the process of altering a given query to improve search or retrieval performance.
  - *Search:* the process of a searcher interacting with an information system.
  - *Retrieval:* the algorithmic behaviors of an information system.

Data Analysis and Session Identification

The transaction log covered a complete 24-hr period with the possibility that certain users will have made several visits to the search engine. Therefore, we had to define a "session" within the transaction log. Although some researchers have used no boundary or an arbitrary temporal cutoff (cf. Su et al., 2000), we believe that this is inconsistent with reported studies of Web searching sessions (He et al., 2002; Jansen & Spink, 2003). Instead, we used a contextual method to define a session using the searcher's IP address and the browser cookie to determine the *initial query* and *subsequent queries*. Specifically, the occurrence of a new IP address and cookie combination always denoted the start of a new session. However, we also examined the query terms for possible new searching episodes. If a query had no terms in common with the user's previous query, we classified this also as the start of a new session. We present an evaluation of this approach later in this article. We then identified content changes in the sequence of queries for each user in the dataset. To implement this method, we assigned each query to a mutually exclusive group based on an IP address, cookie, query content, use of the feedback feature, and query length. The groups are generally consistent with prior work in query classification

[1]We expect to make this Web search engine transaction log available to the research community once the current nondisclosure agreement expires and upon successful negotiation with Infospace.

DOI: 10.1002/asi

(cf. He et al., 2002; Lau & Horvitz, 1999). The classifications of query reformulation that we used are defined as:

- *New:* The query is the first query from a unique *User Identification–Cookie*, or the query is on a new topic from this searcher. We considered the query on a new topic if there were no terms in common with the previous query from a particular user. Although this approach for defining a new topic is not foolproof, from a systems viewpoint, a new query is a new execution against the inverted file index. *New* is the first classification applied.
- *Assistance:* This query is generated by the searcher's selection of an *Are You Looking for?* feature. Many Web search engines have features such as Google's *Did You Mean?*, which focus on spellchecking, and AltaVista's *Prisma* (Anick, 2003), which is a specific query-reformulation feature. The *Assistance* field was the second classification checked for after *New*. The *Assistance* field was helpful in illustrating when the searcher sought assistance from the system and related directly to RQ4.
- *Content Change:* The user executed a query on another content collection. The available content collections were *Web*, *Images*, *Audio*, *News*, and *Video*. This was the third condition checked for during data analysis. Although it is possible to simultaneously change the query and the content collection, we could locate no occurrences of it in the dataset.
- *Generalization:* The current query is on the same topic as the searcher's previous query, but the searcher is now seeking more general information. We determined a query reformulation to be *Generalization* if the query contained fewer terms than the previous query by a particular user. Naturally, we acknowledge that a reformulated query with fewer terms than a previous query by a user will not always be an effort at generalization.
- *Reformulation:* The current query is on the same topic as the searcher's previous query, and both queries contain common terms. We determined a query reformulation to be *Reformulation* if the query contained the same number of terms as the previous query by a particular user with at least one term being in both queries. Naturally, we acknowledge that there may be other methods of reformulating a query.
- *Specialization:* The current query is on the same topic as the searcher's previous query, but the searcher is now seeking more specific information. We determined a query reformulation to be *Specialization* if the query contained more terms than the previous query by a particular user. Naturally, we acknowledge that a reformulated query with more terms than a previous query by a user will not always be an effort at specialization.

In implementing our classification algorithm, the *initial query* (Q_i) from a unique IP address and cookie always identified a new session. In addition, if a *subsequent query* (Q_{i+n}) by a searcher contained no terms in common with the previous query (Q_i), we also deemed this the start of a new session. Therefore, for this research, a session is the user's sequence of queries for a specific search topic. We used common terms across queries to identify this specific topic (and evaluate the effectiveness of this approach). Obviously, from an underlying information-need perspective, these sessions may be related at some level of abstraction. Nevertheless, with no terms in common, one also can credibly make the case that the information state of the user changed, either based on the results from the Web search engine or from other sources (Belkin et al., 1982). In addition, from a system perspective, two queries with no terms in common represent totally different executions to the inverted file index and content collection. Two queries with no terms in common also represent a deviation from the classic building-block approach to query reformulation (Siegfried, Bates, & Wilde, 1993). Huang, Chien, and Oyang (2003) used a somewhat similar approach.

Other researchers have taken simpler approaches, usually involving just the use of the IP address and cookie (Jansen & Spink, 2003; Shi & Yang, 2007) or in conjunction with some temporal cutoff (Silverstein et al., 1999). Fonseca et al. (2003) used the temporal cutoff approach and 10-query limitation for sessions. Fonseca et al. (2005) also used a time limit (i.e., 10 min between queries) to delimit sessions.

We used an automated program that we developed for this research to classify each query in each record in the database using an approach similar to that used by He et al. (2002) to identify temporal sessions in Web searching. The algorithm for the application is:

Algorithm: *Query Reformulation Classification*
Assumptions:

1. Null queries and page request queries are removed.
2. Transaction log is sorted by IP address, cookie, and time (ascending order by time).

Input: Record R_i with IP address (IP_i), cookies (K_i), query Q_i, feedback F_i, and query length QL_i; and record R_{i+1} with IP address (IP_{i+1}), cookies (K_{i+1}), query Q_{i+1}, feedback F_{i+1}, and query length QL_{i+1}.

Variables:
$B = \{t | t \in Q_i \wedge t \in Q_{i+1}\}$//terms in common
$C = \{t | t \in Q_i \wedge t \notin Q_{i+1}\}$//terms that appear in Q_i only
$D = \{t | t \notin Q_i \wedge t \in Q_{i+1}\}$//terms that appear in Q_{i+1} only
$E = \{1 \text{ if } QL_i = QL_{i+1}\}$//queries QL_i and QL_{i+1} are the same length; default is 0.
$G = \{1 \text{ if } QL_i > QL_{i+1}\}$//query QL_i has more terms than QL_{i+1}; default is 0.
$H = \{1 \text{ if } QL_i < QL_{i+1}\}$//query QL_i has fewer terms than QL_{i+1}; default is 0.

Output: Search pattern, *SP*
begin
Move to R_i
Store values for IP_i, K_i, Q_i, F_i, *and* QL_i
$SP =$ New//*default value for first* R_i *in record set*
While not end of file
Move to R_{i+1}
If ($IP_i \neq IP_{i+1}$ *and* $K_i, \neq K_{i+1}$) *then* $SP =$ New
Elseif
{*Calculate values for B, C, D, F, G, and H*
If $F_{i+1} = 1$ *then* $SP =$ Assistance
Elseif ($B \neq \emptyset \wedge C \neq \emptyset \wedge D = \emptyset \wedge E = 0 \wedge G = 1 \wedge H = 0$) *then* $SP =$ Generalization

DOI: 10.1002/asi

TABLE 1. Snippet from search log with fields and query reformulations classified.

User ID	Cookie	Time	Query	Modification	Source	Assistance
1120237134	2MUT282A3UW73OG	12:07:25 P.M.	dead or alive	new	Web	0
1120237134	2MUT282A3UW73OG	12:14:00 P.M.	sonny capone dead or alive	specialization	Web	0
120245	MV2ED3A4BTRYPS	6:41:44 A.M.	how come	new	Video	0
120245	MV2ED3A4BTRYPS	6:42:04 A.M.	how come eminem	specialization	Video	0
120245	MV2ED3A4BTRYPS	6:42:14 A.M.	how come d12	assistance	Video	1
120245	MV2ED3A4BTRYPS	6:43:20 A.M.	how come d12	content change	Images	0
120245	MV2ED3A4BTRYPS	6:43:22 A.M.	how come d12	content change	Audio	0
120245	MV2ED3A4BTRYPS	6:47:07 A.M.	git up d12	reformulation	Audio	0

Elseif $(B \neq \emptyset \wedge C \neq \emptyset \wedge D \neq \emptyset \wedge E = 0 \wedge G = 1 \wedge H = 0)$
then $SP =$ *Generalization with Reformulation*
Elseif $(B \neq \emptyset \wedge C = \emptyset \wedge D \neq \emptyset \wedge E = 0 \wedge G = 0 \wedge H = 1)$
then $SP =$ *Specialization*
Elseif $(B \neq \emptyset \wedge C \neq \emptyset \wedge D \neq \emptyset \wedge E = 0 \wedge G = 0 \wedge H = 1)$
then $SP =$ *Specialization with Reformulation*
Elseif $(B \neq \emptyset \wedge C \neq \emptyset \wedge D \neq \emptyset \wedge E = 1 \wedge G = 0 \wedge H = 0)$
then $SP =$ *Reformulation*
Elseif $(B \neq \emptyset \wedge C = \emptyset \wedge D = \emptyset \wedge E = 1 \wedge G = 0 \wedge H = 0)$
then $SP =$ *Content Change*
Elseif $SP =$ *New}*
(R_{i+1} now becomes R_i)
Store values for R_{i+1} as IP_i, K_i, Q_i, F_i, and QL_i
end loop

As an example of the output, Table 1 contains a snippet of the transaction log consisting of two sessions, including the normal fields from the search log and fields containing the query-reformulation classifications from the algorithm. In Table 1, we see that the first user (User ID 1120237134 and Cookie 2MUT282A3UW73OG) submitted the query "dead or alive" against the Web content at 12:07:25 P.M., and then 7 min later submitted the query "sonny capone dead or alive." This subsequent query, according to the query-classification algorithm, is a specialization. The second user (User ID 120245 and Cookie MV2ED3A4BTRYPS) submitted the query "how come" against the video-content collection; 20 s later, this user specialized the query, then used the assistance feature, switched to the image-content collection, then audio, and then reformulated the query.

Once we had executed the program against the search log and our algorithm classified the query reformulations within all sessions, we then focused on developing a probability transition matrix using a stochastic approach. This effort specifically supports RQ2. A stochastic model can mathematically describe the sequence of states through which searchers progress via transition probability matrices. The value in each cell of a transition probability matrix is the probability of going from the row state to the corresponding column state. Therefore, a transition probability matrix describes a pattern of movements through a state space. Conceptually, for this research, the probability matrix is a map of user query reformulations during Web searching. An analysis of user behavior in this manner not only describes a search state (i.e., the particular query reformulation) at a given point in the sequence but also suggests which states are most likely to follow one another (i.e., what particular query-reformulation state will come next).

A stochastic model is both descriptive and predictive. The number of states that one uses to predict future states is referred to as the order of the stochastic model. A zero-order stochastic process refers to the probability of being at a single state (i.e., no prediction). A first-order stochastic process refers to the probability of arriving at a certain state given a certain number of preceding states. A second-order stochastic process refers to using two previous states and transitions to determine the probability of arriving at a given state. Therefore, at various orders of analyzing a state transition pattern, there is a descriptive component (i.e., the predictive pattern) and predicted component (i.e., the state most likely to follow the predictive pattern).

Once we had developed our probability transition matrix, we then could begin calculating the state–transition pattern for each session (i.e., the number of states and transitions for a given session) and a hash table (H_s) for the entire dataset (i.e., all session patterns). We used n-grams to develop the probabilistic patterns. N-grams are a probabilistic modeling approach used for predicting the next item in a sequence and are (n − 1) order Markov models, where n is the gram (i.e., subsequence or pattern) from the complete sequence or pattern. An n-gram model predicts state x_i using states x_{i-1}, $x_{i-2}, x_{i-3}, \ldots x_{i-n}$. The probabilistic model is then presented as: $P(x_i|x_{i-1}, x_{i-2}, x_{i-3}, \ldots x_{i-n})$, with the assumption that the next state depends only on the last $n - 1$ states, which is, again, an (n − 1) order Markov model.

We used the following algorithm to identify the probability of various state transitions. Let H_s be the prediction model of the next query reformulation based on the previous states (S). Our algorithm is as follows:

Algorithm: *Search Pattern Identification*
Assumptions: Query reformulations are classified.
Input: Record R_i with modification pattern (RP_i); hash table (H_s)
Variables: $S =$ number of states in the model.

Output: Next pattern, *NP*
begin
Move to R_i

DOI: 10.1002/asi

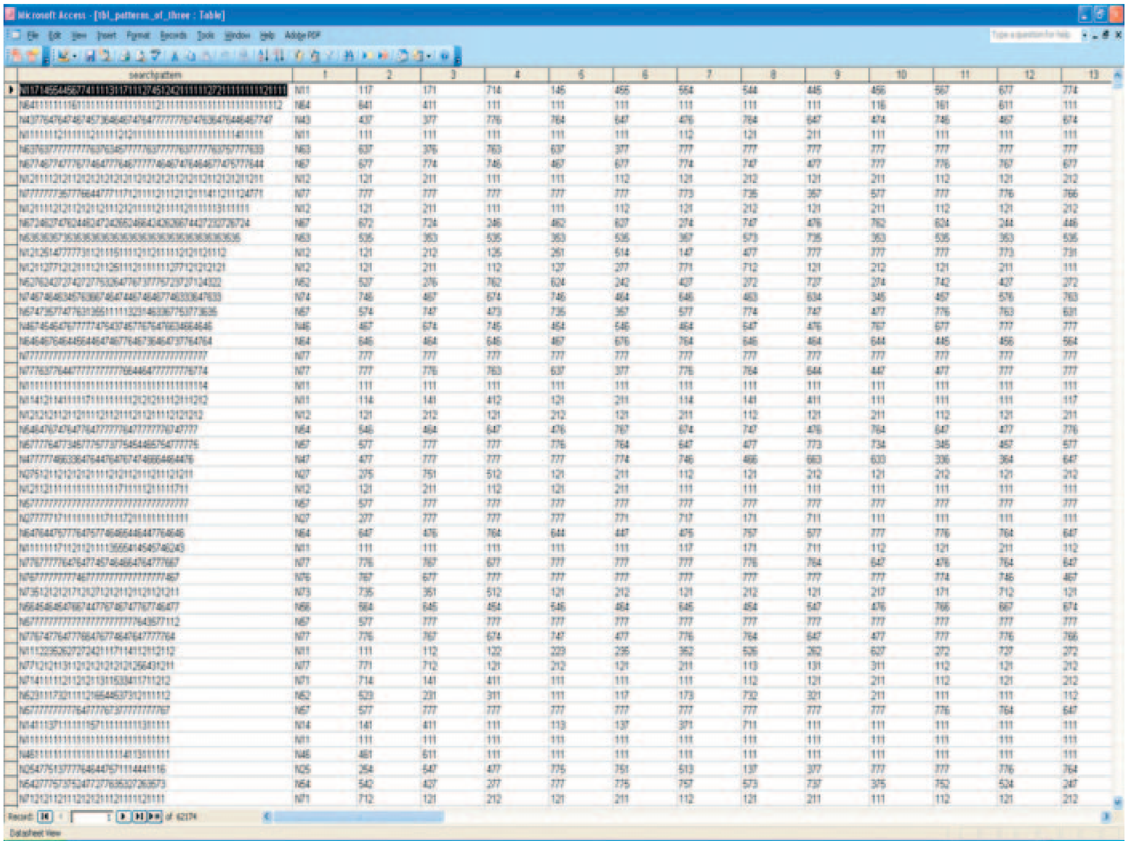

FIG. 1. Example of a hash table with a second-order model.

While not end of file
If [Length of RP < (S + 1)] then NP = NULL (i.e., no prediction)
—Note: This means that the state transition chain does not exist. For example, if a session contains only two queries, there is only one state to state transition.—
Else
For S up to {max [len(RP_{i-s}*)]} do*
$RP_s = (RP_{i-s})$
Find RP index in hash table H_s
$NP = H_s\ (NP_{max})$ *for RP*
End For
EndIf
Move to R_{i+1}
end loop

The hash table for the entire dataset allowed us to address RQ3. With the hash table (H_s) of all occurrences of a given pattern, we could calculate the state transitions during each session at any order model. Figure 1 is a snapshot of a hash table.

Mathematically, for H_s, let S represent a number of states; then the set of states is $S = (0, 1 \ldots n)$, where n is the maximum pattern length for the longest session state transitions in the dataset. We used the maximum requested next state (NP_{max}) at each S as the predicted outcome for that state transition probability. Using this approach, we can then make predictions on a user's next state transition. Additionally, this straightforward approach has the advantage of being implementable in real time. In addition, it allows for evaluating which order is the best predictor for query reformulations within the dataset.

As an example of our algorithm's output, assume the current modification pattern log consists of the pattern "ABCDE," and the order of the pattern is three. In this case, the prediction algorithm checks H_s for the pattern "ABC" and "BCD," and the predicted patterns are "D" and "E," respectively. If the order of the pattern is two, then the algorithm would check for the pattern "AB," "BC," and "CD," returning "C," "D," and "E," respectively. For an order of four, "ABCD" would predict "E."

Evaluation Metrics

There were two areas of focus for evaluation: the classification algorithm and the prediction algorithm. For the classification algorithm, we conducted a verification of our approach by manually classifying 2,000 queries, developing categories of errors a posteriori. For the manual verification, we examined each of the 2,000 queries to evaluate whether the classification was correct in accordance with the intent of the category.

For evaluation of the prediction algorithm, we were interested in three metrics: (a) the accuracy of the predictions, (b) the portion of the dataset that the model covers, and (c) the complexity of the model. For each order of model used in our research, we evaluated its accuracy on each dataset

DOI: 10.1002/asi

TABLE 2. Search log (T).

ABCF
ABCDE
ABCDE
A
AB
AC

TABLE 3. N-gram hash table H_s with $S = 2$.

Predictive pattern	NP_{max}	Prediction accuracy
AB	C	100%
BC	D	66%
CD	E	100%

and calculated the percentage of the dataset that the model addressed as well as the possible number of patterns that a system would have to evaluate if the approach was implemented in real time.

As an illustration, consider a transaction log (T) of session patterns consisting of six individual records (R), each of which represents a user interaction and session containing state modification patterns (RP): Each state (A, B, C, D, or E) in R represents a query-reformulation state (Table 2).

Using these six sessions in T, a second order model (i.e., two states to predict the third) would generate the hash table in Table 3.

Tables 2 and 3 show that our accuracy of prediction varies from 66 to 100%. However, there also are three *sessions* that our model does not represent (i.e., "A," "AB," and "AC") since the length of these patterns is less than $S + 1$. Therefore, the *coverage* (i.e., the percentage of the collection that the path length addresses) is less. Additionally, as the order of our model increases (i.e., as we add more states), the *complexity* (i.e., the cost of performing the calculation) also increases exponentially. Therefore, we evaluate our prediction algorithm in terms of three metrics: *precision*, *coverage*, and *complexity*. *Coverage* is a measure of the applicability of the model to the entire dataset (i.e., how much of the dataset a model of a particular order can accurately address). *Precision* is a measure of the accuracy of the model's prediction. *Complexity* is a measure of the possible patterns of the model relative to the maximum possible patterns of the model complexity under investigation.

We define each of these metrics algorithmically as:

> *Coverage:* Let T be the set of all sessions in the transaction log. Let T_s be the subset of T where the length of RP is greater than $S + 1$. Coverage is then defined as:
>
> $$Coverage = T_s/T$$
>
> *Precision:* Let P^+ be the number of correct predictions and P^- be the possible incorrect predictions at some given S. The union of P^+ and P^- is T_s, which is a subset of T, where the length of RP is greater than $S + 1$. Precision is then defined as:
>
> $$Precision = (P^+)/(P^+ + P^-) = (P^+)/T_s$$
>
> *Complexity:* Let S be the number of states in the model. Let $S = (1, 2, \ldots S_{max})$. Let M be the number of possible transactions at each state. The complexity C of the model at S is $C(S) = M^S$ (i.e., M to the S power). Therefore, one can measure the complexity of the model at some S to the complexity of the model at S_{max}:
>
> $$Complexity = [C_S/C\ (S_{max})]$$

TABLE 4. Aggregate statistics from the Dogpile search log.

Sessions	964,780	
Queries	1,523,793	
Terms		
Unique	298,796	7.03%
Total	4,250,656	
Mean terms per query	2.79 $SD = 1, 54$	
Session size		
1 query	691,672	71.64%
2 queries	153,056	15.85%
3+ queries	120,052	12.51%
	964,780	100.0%

All three of these measures are bounded within the set $(0, \ldots 1)$, so they allow for comparison across systems and path-calculation methods.

For RQ4, we examined when in the search pattern that the users sought out system assistance using the prior searching states. We examine the state prior to the seeking of assistance order to help determine what leads searchers to seek system help.

Results

We now return to our research questions and presentation of the results of our analysis. We first conducted an overall analysis of the dataset, shown in Table 4. There were 2,465,145 interactions during the data-collection period. Of these interactions, there were 1,523,793 queries submitted by 534,507 users (identified by unique IP address and cookie) containing 4,250,656 total terms. Our classification algorithm identified 964,780 unique sessions. There were 298,796 unique terms in the 1,523,793 queries. The mean query length was 2.79. Nearly 71.74% of the sessions contained only one query. These statistics for query length, session length, and term usage are in line with those reported in prior work (cf. Park et al., 2005; Silverstein et al., 1999; Wang et al., 2003; Wolfram, 1999).

RQ1

Results for our first research question (*What is the distribution of search states of query reformulations during Web*

DOI: 10.1002/asi

TABLE 5. Occurrences of query reformulation.

Search patterns	Occurrence	%	Occurrence (excluding *New*)	% (excluding *New*)
New	964,780	63.34	–	–
Reformulation	126,901	8.33	126,901	22.73
Assistance	124,195	8.15	124,195	22.25
Specialization	90,893	5.97	90,893	16.28
Content change	65,949	4.33	65,949	11.81
Specialization w/reformulation	55,531	3.65	55,531	9.95
Generalization w/reformulation	54,637	3.59	54,637	9.78
Generalization	40,186	2.64	40,186	7.20
	1,523,072	100.00	558,292	100.00

searching?) are shown in Table 5. Table 5 shows that the state *New* represents the majority of query classifications (>63%). Nearly 40% of query submissions were some sort of query reformulation. We also see in Table 5 that more than 8% of the query reformulations were for *Reformulation*, with another approximately 8% of query reformulations resulting from system *Assistance*. If we exclude the *New* queries, *Reformulation* and *Assistance* states accounted for nearly 45% of all query reformulations. This indicates two important characteristics of these Web searchers. First, they seem at first to have an unclear idea as to how to express their information need, so they try a series of queries to zero in on the proper domain. Second, it shows that searchers are open to system assistance in query reformulations, given the rate of implementation of system assistance.

A chi-square statistical test of significance (Greenwood & Nikulin, 1996) showed that the distribution of query reformulations is not uniform across the modification states, $\chi^2(6) = 90.842$, $p < .01$. This finding would seem to indicate that a substantial portion of searchers go through a process of defining their information need by exploring various terms or using system feedback to modify the query. Another 16% of query reformulations are *Specialization*, supporting prior reports that precision is a primary concern for Web searchers (Jansen & Spink, 2005). Nearly 12% of Web users searched multiple content collections in their quest for relevant information, illustrating the need for not only the right content at the right time but also content in the right format.

We conducted a verification of our query-classification algorithm by manually classifying 2,000 queries. We arrived at four categories of errors, developed posteriori:

- *Misspelling:* a word was misspelled or a previously misspelled word caused a change resulting in a misclassification (causes a false *New* or *Reformulation*).
- *Cookie:* Either the cookie was not defined or there was a change in the cookie, but not a change in the user (causes a false *New*).
- *Special character change:* The original query contained special characters (causes a false *New* or *Reformulation*). These special characters were a mix of items, such as "+," "−," "?," "," and so on.
- *Other:* a miscellaneous collection of other reasons (causes a false *New*).

TABLE 6. Misclassifications from 2,000-query sample.

Type of misclassification	Occurrences	Errors (%)
Misspelling	52	58.5
Cookie	23	25.8
Special character change	5	5.6
Other	9	10.1
Total	89	100.00

Table 6 shows that of the 2,000 queries manually classified, there were 89 deemed mistakes; furthermore, most of the errors were due to misspellings (i.e., the classification algorithm identified the word as a *new* term when in reality the user had misspelled a term in the original query and corrected the term in the subsequent query. Most misspellings occurred due to missing spaces in words. However, the sum total of all misclassifications was 4.4%, resulting in a 95.6% accuracy rate for the algorithm. Therefore, we deemed the approach valid.

In addition, we manually evaluated 1,000 queries and the category algorithmically assigned to evaluate our underlying assumptions about each of the categories, most notably that a subsequent query with no terms in common with the previous query represents a new session. From the 1,000, 4.8% ($n = 48$) were improperly assigned. The most common occurrence that caused errors was the use of an identical query term for different topics ($n = 15$). This was most common with a navigational sequence of queries (e.g., the use of .com or .edu or .net) or with the use of natural language queries. Spelling corrections ($n = 11$) were the next common source of errors, followed by parallel/multitasking searching ($n = 9$). Hierarchy searching ($n = 8$) was an interesting cause of misclassification. This occurred when the searcher entered a series of queries that used no term in common, but the queries had an obvious hierarchical relationship (e.g., the following query sequence: *reptiles, snake, cobra, king cobra, usa snakes*). The use of related terms ($n = 4$) and synonyms ($n = 1$) (e.g., *ipod*, *mp3 player*) rounded out the error list. Again, with a 95.2% accuracy rate, the underlying assumptions of the algorithmic approach seem valid.

TABLE 7. Transition probability matrix.

	New	Content change (%)	Reformulation (%)	Generalization (%)	Generalization w/reformulation (%)	Specialization (%)	Specialization w/reformulation (%)	Assistance (%)	Total (%)
New	–	13	21	7	7	**22**	9	**21**	100
Content change	–	0	17	11	8	16	7	**41**	100
Reformulation	–	11	0	14	**18**	19	**23**	15	100
Generalization	–	10	18	0	5	**37**	12	18	100
Generalization w/reformulation	–	6	**32**	6	0	18	27	11	100
Specialization	–	9	**32**	16	**22**	0	12	9	100
Specialization w/reformulation	–	6	**28**	14	**36**	8	0	7	100
Assistance	–	**58**	12	7	11	5	8	0	100

TABLE 8. Occurrences of query reformulation.

	New	Content change (%)	Reformulation (%)	Generalization (%)	Generalization w/reformulation (%)	Specialization (%)	Specialization w/reformulation (%)	Assistance (%)	Total (%)
New	–	13	21	7	7	**22**	9	**21**	100
Content change	–	0	17	11	8	16	7	**41**	100
Reformulation	–	11	0	14	**18**	19	**23**	15	100
Generalization	–	10	18	0	5	**37**	12	18	100
Generalization w/reformulation	–	6	**32**	6	0	18	27	11	100
Specialization	–	9	**32**	16	**22**	0	12	9	100
Specialization w/reformulation	–	6	**28**	14	**36**	8	0	7	100
Assistance	–	**58**	12	7	11	5	8	0	100

RQ2

Results concerning our second research question (*What states are most likely to follow one another in Web searching*?) are shown in the transition probability matrix (i.e., first-order analysis) in Table 7. The value in each cell is the probability of going from the row state to the corresponding column state. The most frequently occurring states for each row (i.e., the start state) are bolded, directly answering our research question. By definition, *New* is always the first state in a session.

Table 7 shows that there appears to be a connection between the searcher shifting content collections and the use of system assistance, with a majority (58%) of assistance usage occurring just before a content change or just after a content change (41%). The use of assistance during these transitions accounted for 25% of all assistance usage. Users also appear to be somewhat receptive to *Assistance* at the start of the session (21%), just after their initial query. As stated previously, we see high occurrences of *Reformulation* after *New* (21%) and after *Specialization* (32%), with a variety of modification variations on this base pattern. This would indicate that searchers use interactions with the system, probably the results listings, to explore the information space with new query terms; however, searchers typically do not broaden or narrow queries until they focus on the content area. There also appears to be a tendency to go from *Generalization* to *Specialization* (37%), representing a standard building-block methodology of searching. *Specialization* also appears to be a tendency immediately after the initial query, with 28% of searchers immediately moving to narrow their queries. These are probably the searchers who have a well-defined expression of their information need.

RQ3

For results concerning our third research question (*What order of state transition provides the best predictability for query reformulation during Web searching?*), we refer to Table 8. We analyzed the entire dataset and also divided the log file into five separate subsets and individually analyzed each subset. Results for the entire dataset and each of the subsets were similar, so we report results only for the entire dataset in this article.

Table 8 presents the results from our analyses of the dataset at five different model orders, zero to four (i.e., one–five states, respectively). Although some prior work has excluded smaller order sessions of the collected data from model analysis (cf. Su et al., 2000), we believe it is important to apply all models to the entire dataset to obtain an accurate evaluation of the models' performance, applicability, and

DOI: 10.1002/asi

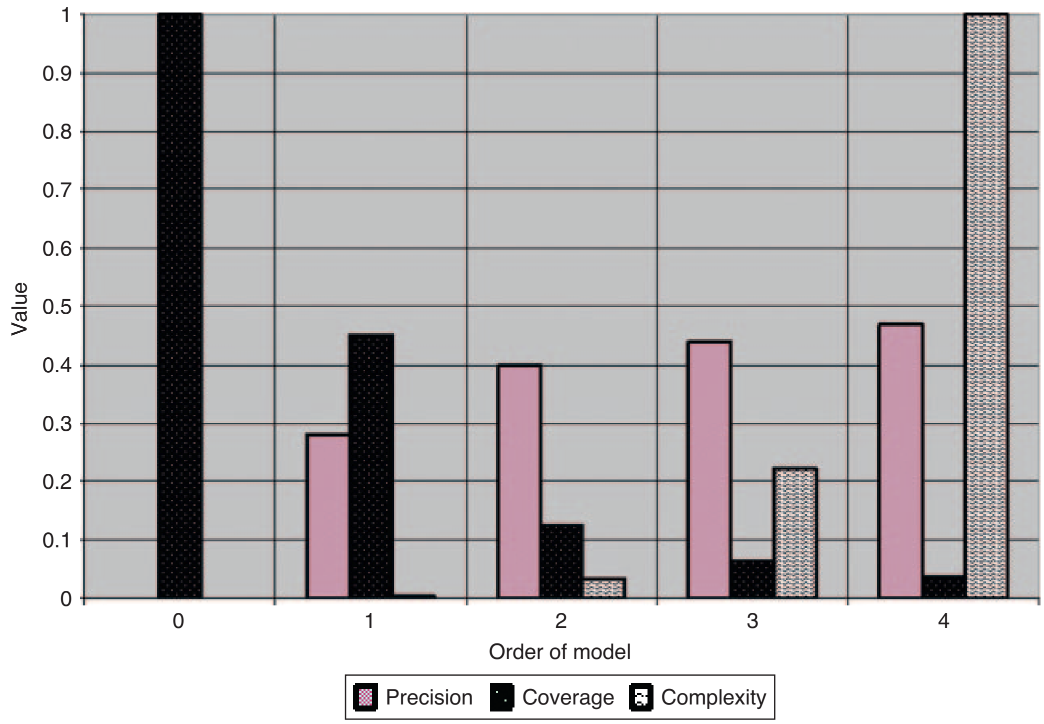

FIG. 2. Results of evaluation metrics of the first- through fourth-order models.

effectiveness. Additionally, we present the results for the zero-order model. Although providing little predictability, the zero-order model provides a baseline for comparison.

Table 8 shows that the models greatly varied based on the metric used. The first-order model has coverage of 45%, but a precision of only 28%. The fourth-order model has the highest precision (47%), providing the best predictability; however, the coverage of the fourth-order model is only 4%, and the complexity is 0.195. Figure 2 illustrates the relationship among precision, coverage, and complexity for the first- through fourth-order models [i.e., $S = (2, 3, 4, 5)$].

Based on precision, it appears that the second-order model generally may be the best to use. For second-order models, precision is high (40%), and complexity is low. The second-order model also represents a fair coverage of the dataset (13%).

TABLE 9. Occurrences of assistance usage.

Starting or ending state	To Assistance (%)	From assistance (%)
New	**21**	0
Content change	**41**	**58**
Reformulation	15	12
Generalization	18	7
Generalization w/reformulation	11	11
Specialization	9	5
Specialization w/reformulation	7	8
Assistance	0	0

RQ4

Concerning our fourth research question (*When are users most receptive to system assistance?*), we examine findings in Table 9, which shows each of our possible states. Column two shows the percentage transition from each of the states to the use of system assistance (e.g., from *New* to *Assistance* is 21%). Column three shows the percentage transition from *Assistance* to each of the other possible states (e.g., *Assistance* to *Reformulation* is 12%).

From Table 9, some interesting findings appear. First, there is a preference for using assistance at the start of the searching sessions (i.e., 21% of assistance usage occurred as the next state immediately after the submission of the first query). This would seem to indicate that users prefer system assistance to get started in the search process. This would indicate further research in leveraging implicit feedback (Oard & Kim, 2001) and contextual help (Xie & Cool, 2009) at the beginning of the searching episode and less later. Second, 41% of assistance usage occurred in the next state after a content change. This would again point to users seeking contextual help in formulating these new queries for the specific type of media (i.e., *Images*, *Video*, *Web, Audio*, or *News*). Finally, after the use of assistance, 58% of the time, the following pattern was a *Content Change*. This is interesting because the search engine assistance does not offer recommendations on switching content collections. Therefore, this content switch was an independent action by the searchers. Again, this would indicate a critical junction in the information-searching session where the searcher deemed it appropriate to change from a

DOI: 10.1002/asi

previous searching strategy. It seems reasonable that systems would tailor searching assistance for these critical junctions.

Discussion

Our research findings indicate that there were high occurrences of *Reformulation*, *Assistance*, and *Specialization* query-reformulation states. These states were especially prevalent immediately after the submission of the initial query (approximately 22% for each). Searchers appeared to execute a great deal of *Reformulation* as they tried to express more precisely their information need. They typically moved to narrow their query at the start of the session, moving to *Reformulation* in the mid- and latter portions of the sessions. Implications for designing contextual help are that it appears that assistance to narrow the query and alternate query terms would be beneficial immediately after the initial query submission. As the session progresses, the openness or benefit of such assistance would decrease.

There were low rates of implementing system assistance in conjunction with these states in the sessions. Instead, the most usage of systems assistance occurred immediately after *Content Changes*. The use of system *Assistance* at this state indicates that searchers are more open to system intervention during these content-collection shifts. As for why they are less likely to implement *Assistance* immediately after query submission, it may be that they are too cognitively focused on correctly expressing their information need to attend to anything else. The implication is that system assistance should be most specifically targeted to the user making a cognitive shift (i.e., using a different content vertical), when it appears searchers are open to system intervention.

We assessed zero- through fourth-order models to gauge the predictive quality of the models. We implemented this query classification in an approach that does not require user profiles as in Pazzani, Muramatsu, and Billsus (1998) and is not acyclic as in Lau and Horvitz (1999), in that our approach allows for searchers to return to previous states. The only constraint is that user queries must be logged successfully with a time stamp. These fields are common to most search engine transaction logs, so the method is reasonable for real-time use. We evaluated each of the models in terms of precision, coverage, and complexity. The implication is that this approach is implementable in real time.

The findings show that using a second-order model provides substantial prediction accuracy, good coverage of the data, and reasonable complexity. Although higher order models do provide some increases in accuracy of prediction, they also result in decreases in coverage and drastic increases in complexity. Additionally, if we exclude the sessions for which the second-order model could not cover (i.e., a session of only two states, respectively), the coverage of this model increases greatly. Nearly 72% of the sessions contained only one query. Results also indicate that for certain patterns, some models can provide substantial degrees of predictability—up to 70 to 80% in some cases. These results point to the possible need of the simultaneous use of n-order models in predicting query reformulation. Web search engines can use this approach to time system intervention into the search process, especially at the start of the session and when it appears that a content shift is about to occur.

There are both limitations and strengths of this research. For limitations, the detection of session boundaries is a significant challenge in Web searching. Combined with the aspects of lack of concrete user identification and fuzziness in detecting human versus agent searching, this leaves room for some ambiguity in calculations. However, the results of log analysis for Web searching has paired nicely with results from laboratory studies of Web searching (cf. Hargittai, 2002; Jansen & McNeese, 2005), so these concerns appear to have limited practical impact. In addition, there may be more sophisticated methods for detecting session boundaries than we used here, such methods that incorporate multiple tasking and parallel searches; however, our evaluation shows that these more sophisticated methods would lead to incremental improvements relative to the approach in this research, although this effort would be valid to increase the performance from approximately 90% to closer to 100%.

Concerning strengths, research on the detection of query-reformulations patterns during Web searching sessions is a critical area for designing more advanced searching systems, especially in the more multifaceted searching contexts of exploratory, successive, and multitasking searching situations. The methods presented in this research rely on the content of searchers' queries, along with other data normally collected by the search engine to identify and predict query reformulations. Therefore, this methodology is advantageous for real-time system implementation.

The implications from these research results are promising for the design of information retrieval and Web searching systems. Given that one can predict future states of query formulation based on previous and present states with a reasonable degree of accuracy, one can design information systems that provide query-reformulation assistance (Anick, 2003), automated searching-assistance systems (Jansen, 2006), recommender systems (Callan & Smeaton, 2003), and exploratory searching systems (White, Kules, Drucker, & Schraefel, 2005), among others.

Conclusion

The aim of this research was to develop a workable method of predicting future query reformulation of Web searchers with the intention of detecting when these searchers would be open to system assistance and what type of assistance they would most likely require to help in query reformulation. Our goal was to develop predictive models from which we could calculate future actions of Web searchers. We algorithmically classified queries into eight mutually exclusive categories and generated state-transition patterns of query reformulation using n-gram models during searching sessions. We then presented the distribution of different query-reformulation states.

DOI: 10.1002/asi

In this research, we used a search log of 4,056,374 records to investigate query-reformation patterns. We then developed state-transition chains for each searching episode. We applied an n-gram approach from zero to the fourth order to determine which order was best for predicting query reformulation. Our findings showed that as the order increases, so does the predictive power. However, this increase in prediction comes at significant cost of complexity and coverage of the dataset. Therefore, a first- or second-order model seems to have the best overall set of metrics (i.e., precision, complexity, and coverage).

In future research, we aim to use enhanced versions of the algorithms and evaluation metrics presented here to facilitate cross-system investigations. As for potential enhancements, we would like to investigate the application of temporal characteristics to see if the time interval between query submissions affects the next state. Based on findings from the results reported here, we also will investigate cascading stochastic models (Brants, 1999) to make predictions where multiple-order stochastic models are organized in a stepwise manner. Each slice of the search log structure is represented by its own order of stochastic model, and output of the aggregate slices can represent the entire search log. By limiting the application of various ordered models (i.e., first–fifth) to specific searching episodes, we might be able to achieve high precision while limiting increases in complexity and obtaining complete coverage of the search log. Finally, we plan to investigate whether the patterns vary based on the query domain (i.e., health, ecommerce, and entertainment).

Overall, the results presented here are promising for the application of n-grams to query reformulation in Web searching to enhance algorithmic applications of automated assistance and contextual help.

Acknowledgments

Our sincere thanks to Infospace, Inc. for providing the dataset used for this research project. We encourage other Web search engine companies to seek ways to collaborate with the academic research community. The Air Force Office of Scientific Research (AFOSR) and the National Science Foundation (NSF) funded portions of this research.

References

Adamczyk, P.D., & Bailey, B.P. (2004). If not now, when? The effects of interruption at different moments within task execution. In E. Dykstra-Erickson & M. Tscheligi (Chairs), Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 04) (pp. 271–278). New York: ACM Press.

Anick, P. (2003, July/August). Using terminological feedback for Web search refinement—A log-based study. In C. Clarke, G. Cormack, J. Callan, D. Hawking, & A. Smeaton (Chairs), Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 88–95). New York: ACM Press.

Belkin, N., Cool, C., Kelly, D., Lee, H.-J., Muresan, G., Tang, M.-C., et al. (2003, July/August). Query length in interactive information retrieval (pp. 205–212). Paper presented at the 26th annual International ACM Conference on Research and Development in Information Retrieval, Toronto, Canada.

Belkin, N., Oddy, R., & Brooks, H. (1982). Ask for information retrieval, Parts 1 and 2. Journal of Documentation, 38(2), 61–71, 145–164.

Brants, T. (1999). Cascaded Markov models. In H. Thompson & A. Lascarides (Eds.), Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics (pp. 118–123). San Fransciso, CA: Morgan Kaufmann Publishers (for the Association for Computational Linguistics).

Callan, J., & Smeaton, A. (2003). Personalisation and recommender systems in digital libraries. Retrieved January 1, 2009, from http://www-2.cs.cmu.edu/~callan/Papers/Personalisation03-WG.pdf

Chapman, J. (1981). A state transition analysis of online information seeking behavior. Journal of the American Society for Information Science, 32(5), 325–333.

Chen, H.-M., & Cooper, M.D. (2001). Using clustering techniques to detect usage patterns in a Web-based information system. Journal of the American Society for Information Science and Technology, 52(11), 888–904.

Chen, H.-M., & Cooper, M.D. (2002). Stochastic modeling of usage patterns in a web-based information system. Journal of the American Society for Information Science and Technology, 53(7), 536–548.

Chen, M., LaPaugh, A.S., & Singh, J.P. (2002). Predicting category accesses for a user in a structured information space. In K. Järvelin, M Beaulieu, R. Baeza-Yates, & S. H. Myaeng (Chairs), Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 65–72). New York: ACM Press.

Choo, C., Detlor, B., & Turnbull, D. (1998). A behavioral model of information seeking on the Web: Preliminary results of a study of how managers and it specialists use the Web. In C.M. Preston (Ed.), Proceedings of the 61st Annual Meeting of the American Society for Information Science (ASIS) (pp. 290–302). Medford, NJ: Information Today.

Croft, W.B., & Thompson, R.H. (1987). I3: A new approach to the design of document retrieval systems. Journal of the American Society for Information Science, 38(6), 389–404.

Fonseca, B.M., Golgher, P., Pôssas, B., Ribeiro-Neto, B., & Ziviani, N. (2005). Concept-based interactive query expansion. In A. Chowdhury, N. Fuhr, M. Ronthaler, H.-J. Schek, & W. Teiken (Eds.), Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM'05) (pp. 696–703). New York: ACM Press.

Fonseca, B.M., Golgher, P.B., De Moura, E.S., & Ziviani, N. (2003). Using association rules to discovery search engines' related queries. Proceedings of the 1st Latin American Web Congress (LA-Web 2003) (pp. 66–71). Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Gauch, S., & Smith, J. (1993). An expert system for automatic query reformulation. Journal of the American Society for Information Science, 44(3), 124–136.

Greenwood, P.E., & Nikulin, M.S. (1996). A guide to chi-squared testing. New York: Wiley.

Hargittai, E. (2002). Beyond logs and surveys: In-depth measures of people's web use skills. Journal of the American Society for Information Science and Technology, 53(14), 1239–1244.

He, D., Göker, A., & Harper, D.J. (2002). Combining evidence for automatic Web session identification. Information Processing & Management, 38(5), 727–742.

Huang, C.-K., Chien, L.-F., & Oyang, Y.-J. (2003). Relevant term suggestion in interactive web search based on contextual information in query session logs. Journal of the American Society for Information Science and Technology, 54(7), 638–649.

Jansen, B.J. (2006). Using temporal patterns of interactions to design effective automated searching assistance systems. Communications of the ACM, 49(4), 72–74.

Jansen, B.J., & McNeese, M.D. (2005). Evaluating the effectiveness of and patterns of interactions with automated searching assistance. Journal of the American Society for Information Science and Technology, 56(14), 1480–1503.

Jansen, B.J., & Pooch, U. (2001). Web user studies: A review and framework for future work. Journal of the American Society of Information Science and Technology, 52(3), 235–246.

Jansen, B.J., & Spink, A. (2003, June). An analysis of Web information seeking and use: Documents retrieved versus documents viewed. Paper presented at the 4th International Conference on Internet Computing, Las Vegas, Nevada.

Jansen, B.J., & Spink, A. (2005). How are we searching the World Wide Web? A comparison of nine search engine transaction logs. Information Processing & Management, 42(1), 248–263.

Jansen, B.J., Spink, A., Blakely, C., & Koshman, S. (2006). Web searcher interactions with the dogpile.com meta-search engine. Journal of the American Society for Information Science and Technology, 58(4), 1875–1887.

Jansen, B.J., Spink, A., Blakely, C., & Koshman, S. (2007). Defining a session on Web search engines. Journal of the American Society for Information Science and Technology, 58(6), 862–871.

Lau, T., & Horvitz, E. (1999, June). Patterns of search: Analyzing and modeling Web query refinement (pp. 119–128). Paper presented at the 7th International Conference on User Modeling, Banff, Alberta, Canada.

Marchionini, G. (1989). Information-seeking strategies of novices using a full-text electronic encyclopedia. Journal of the American Society for Information Science, 40(1), 54–66.

Meadow, C.T., Hewett, T.T., & Aversa, E. (1982a). A computer intermediary for interactive database searching: I. Design. Journal of the American Society for Information Science, 33, 325–332.

Meadow, C.T., Hewett, T.T., & Aversa, E. (1982b). A computer intermediary for interactive database searching ii: Evaluation. Journal of the American Society for Information Science, 33, 357–364.

Oard, D., & Kim, J. (2001). Modeling information content using observable behavior. Proceedings of the 64th Annual Meeting of the American Society for Information Science and Technology, (pp. 38–45). Medford, NJ: Information Today.

Özmutlu, H.C., & Çavdur, F. (2005). Application of automatic topic identification on Excite Web search engine data logs. Information Processing & Management, 41(5), 1243–1262.

Özmutlu, H.C., Çavdur, F., Spink, A., & Özmutlu, S. (2004). Neural network applications for automatic new topic identification on Excite Web search engine data logs. Proceedings of the 67th Annual Meeting of the American Society for Information Science and Technology (pp. 1–10). Medford, NJ: Information Today.

Özmutlu, H.C., Çavdur, F., Spink, A., & Özmutlu, S. (2005, October/November). Cross validation of neural network applications for automatic new topic identification (pp. 1–10). Paper presented at the Association for the American Society of Information Science and Technology, Charlotte, NC. Retrieved March 17, 2009, from http://eprints.rclis.org/5263/1/Ozmutlu_Cross.pdf

Park, S., Bae, H., & Lee, J. (2005). End user searching: A Web log analysis of NAVER, a Korean Web search engine. Library & Information Science Research, 27(2), 203–221.

Pazzani, M., Muramatsu, J., & Billsus, D. (1996). Syskill & Webert: Identifying interesting web sites. Proceedings of the 13th National Conference on Artificial Intelligence and 8th Innovative Applications of Artificial Intelligence Conference, Volume 1(AAAI 96). Menlo Park, CA: AAAI Press.

Penniman, W.D. (1975, October). A stochastic process analysis of online user behavior. In C.W. Husbands & R.L. Tighe (Eds.), Proceedings of the 38th Annual Meeting of the American Society for Information Science (ASIS) (pp. 147–148). Washington, DC: American Society for Information Science.

Penniman, W.D. (1982). Modeling and analysis of online user behavior. In A.E. Petrarca, C.I. Taylor, & R.S. Kohns (Eds.), Proceedings of the 45th Annual Meeting of the American Society for Information Science (pp. 231–235). White Plains, NY: Knowledge Industry Publications.

Qiu, L. (1993). Markov models of search state patterns in a hypertext information retrieval system. Journal of the American Society for Information Science, 44(7), 413–427.

Rieh, S.Y., & Xie, H. (2006). Analysis of multiple query reformulations on the web: The interactive information retrieval context. Information Processing & Management, 42(3), 751–768.

Shen, X., Dumais, S., & Horvitz, E. (2005). Analysis of topic dynamics in Web search. In A. Ellis & T. Hagino (Chairs), Proceedings of the 14th International Conference on World Wide Web (pp. 1102–1103). New York: ACM Press.

Shi, X., & Yang, C.C. (2007). Mining related queries from web search engine query logs using an improved associate rule mining model. Journal of the American Society for Information Science and Technology, 58(12), 1871–1883.

Siegfried, S., Bates, M., & Wilde, D. (1993). A profile of end-user searching behavior by humanities scholars: The Getty Online Searching Project Report No. 2. Journal of the American Society for Information Science, 44(5), 273–291.

Silverstein, C., Henzinger, M., Marais, H., & Moricz, M. (1999). Analysis of a very large Web search engine query log. SIGIR Forum, 33(1), 6–12.

Spink, A. (1997). A study of interactive feedback during mediated information retrieval. Journal of the American Society for Information Science, 48(5), 382–394.

Su, Z., Yang, Q., & Zhang, H.-J. (2000). A prediction system for multimedia pre-fetching in Internet. In S. Ghandeharizadeh, S.-F. Chang, S. Fischer, J.A. Konstan, & K. Nahrstedt (Chairs), Proceedings of the Eighth ACM International Conference on Multimedia (pp. 3–11). New York: ACM Press.

Tolle, J.E. (1984). Monitoring and evaluation of information systems via transaction log analysis. In A.S. Pollitt (Chair), Proceedings of the 7th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 247–258). New York: ACM Press.

Tolle, J.E., & Hah, S. (1985). Online search patterns: NLM CATLINE database. Journal of the American Society for Information Science, 36, 82–93.

Wang, P., Berry, M., & Yang, Y. (2003). Mining longitudinal Web queries: Trends and patterns. Journal of the American Society for Information Science and Technology, 54(8), 743–758.

White, R.W., Kules, B., Drucker, S.M., & Schraefel, M.C. (2005). Workshop and conference reports: Exploratory search interfaces: Categorization, clustering and beyond: Report on the XSI 2005 Workshop at the Human–Computer Interaction Laboratory, University of Maryland. ACM SIGIR Forum, 39(2), 52–56.

Wolfram, D. (1999). Term co-occurrence in Internet search engine queries: An analysis of the Excite data set. Canadian Journal of Information and Library Science, 24(2/3), 12–33.

Xie, I., & Cool, C. (2009). Understanding help-seeking within the context of searching digital libraries. Journal of the American Society for Information Science and Technology, 60(3), 477–494.

Zhang, Z., & Nasraoui, O. (2007). Efficient hybrid Web recommendations based on Markov clickstream models and implicit search. In T.Y. Lin, L. Haas, J. Kacprzyk, R. Motwani, A. Broder, & H. Ho (Eds.), Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (pp. 621–627). Piscataway, NJ: Institute of Electrical and Electronics Engineers.

DOI: 10.1002/asi