Beyond Actions: Exploring the Discovery of Tactics from User Logs

Jiyin He[a], Pernilla Qvarfordt[b], Martin Halvey[c], Gene Golovchinsky[b]

[a]*Centrum Wiskunde & Informatica, Science Park 123, Amsterdam, the Netherlands*
[b]*FX Palo Alto Laboratory, Inc., 3174 Porter Drive, Palo Alto, CA, USA*
[c]*Department of Computer and Information Sciences, University of Strathclyde, Glasgow, G1 1XQ, Scotland, UK*

Abstract

Search log analysis has become a common practice to gain insights into user search behaviour; it helps gain an understanding of user needs and preferences, as well as an insight into how well a system supports such needs. Currently, log analysis is typically focused on low-level user actions, i.e. logged events such as issued queries and clicked results, and often only a selection of such events are logged and analysed. However, types of logged events may differ widely from interface to interface, making comparison between systems difficult. Further, the interpretation of the meaning of and subsequent analysis of a selection of events may lead to conclusions out of context—e.g. the statistics of observed query reformulations may be influenced by the existence of a relevance feedback component. Alternatively, in lab studies user activities can be analysed at a higher level, such as search tactics and strategies, abstracted away from detailed interface implementation. Unfortunately, until now the required manual codings that map logged events to higher-level interpretations have prevented large-scale use of this type of analysis. In this paper, we propose a new method for analysing search logs by (semi-)automatically identifying user search tactics from logged events, allowing large-scale analysis that is comparable across search systems. In addition, as the resulting analysis is at a tactical level we reduce potential issues surrounding the need for interpretation of low-level user actions for log analysis. We validate the efficiency and effectiveness of the proposed tactic identification method using logs of two reference search systems of different natures: a product search system and a video search system. With the identified tactics, we perform a series of novel log analyses in terms of entropy rate of user search tactic sequences, demonstrating how this type of analysis allows comparisons of user search behaviours across systems of different nature and design. This analysis provides insights not achievable with traditional log analysis.

Keywords: Search behaviour, search tactics, log analysis

1. Introduction

Understanding how people use interactive search systems can be expensive in terms of time, money, resources etc. With this in mind, almost every search system, ranging from lab studies up to commercial search engines, keeps a usage log of the system. These logs provide a persistent record of interactions that users have had with these search systems. Analysis of these logs often yields important insights into user search behaviour that help to understand user needs and preferences, as well as providing insight into the performance of a system in supporting user needs. Logs can be used to study how users approach a particular type of search task and provide guidelines and suggestions for informed system design, e.g. [53, 52]. Logs can also be used to perform formative or summative evaluations of search user interfaces using A/B testing for example [35], where statistics of a specific logged event often serve as an online metric. For instance, click-through rate from major Web search engines are often considered the primary source for inferring user preference in search results [32].

Current log analyses, in particular analyses of large-scale logs from commercial search systems, are focused on low-level user actions, i.e. logged events such as keyword queries and result clicks. Further,

Email addresses: j.he@cwi.nl (Jiyin He), pernilla@fxpal.com (Pernilla Qvarfordt), martin.halvey@strath.ac.uk (Martin Halvey)

typically only a selection of such events are being analysed, as demonstrated by studies that are focused on query logs, e.g. [6, 27, 31, 5, 40]; click logs, e.g. [32, 17, 10]; dwell time, e.g. [39, 34], etc. Despite the benefits gained (in part because of the ease and scale) from current log analyses there are also some drawbacks, most prominently *interpretability* and *comparability*.

Interpretability. During a course of search, users interact with various components of a system, and these interactions together advance the progress of their search task. Analysis based on a selected single type of logged event may lead to an interpretation of the event out of context, and thus observations of biased statistics. This may not be obvious with logs from a system equipped with a standard 10-blue-links search interface, where interpretation of logged events is relatively clear: e.g. clicks as users perceive results being relevant [32] and mouse hovers as users inspect result summaries [30]. However, as modern search interfaces become increasingly complex, allowing more dynamic user interactions beyond the traditional query-ranked list interactions, the problem becomes more prominent. The reason for issues surrounding interpretability is two-fold. (1) The interpretation of a logged event can be ambiguous, and the same user action may have different intentions depending on the context. For instance, when hovering over a result summary, instead of reading this particular summary, the user may be just skimming over the result list. The context of this action provides indications of its actual intention: e.g. the latter is more likely if it is shortly followed by a hover over another result, while the former is more likely if it is followed by an action such as book-marking the result. (2) The observed statistics of particular types of logged events are not independent from interactions enabled by other components of the search interface, as it may be possible for users to achieve the same outcome via different operations. For instance, users often need to reformulate their search multiple times before reaching their search goal. However, with a system equipped with a relevance feedback component, the observation that users perform fewer query reformulations does not necessarily mean that they spent less effort as they may have been using the relevance feedback component as an alternative way to modify their searches.

Comparability. The types of events that can be observed from a search log depend on the design of the system and the interactions allowed by the search interface, which may vary widely from system to system. This makes a direct comparison of the observations between systems very difficult. It is also difficult to compare user interactions involving different content types— even with similar system designs. For instance, it would be difficult to compare user activities recorded in the log of a video search system [25] to those logged by the image search systems that inspired its development [8, 50, 43]—as the type of low-level user actions recorded as logged events would be inherently different, although the actual type of the intended interactions may be the same—e.g. to "examine" a result, users would "play" a video, but "look at" an image.

To overcome the drawbacks highlighted above, what is needed is a log analysis method that (i) makes observations from search logs in context; and (ii) enables comparisons across systems of different designs and implementations. One way to overcome these drawbacks is to abstract away from the low-level logged events and analyse user activities in terms of a higher-level representation such as search tactics e.g. [2, 42, 3]. Bates [1] originally defined search tactics as "a move made to further a search". Typically, tactics describe various stages of the information seeking process, for example in Bates' model a set of search formulation tactics describes various actions around designing or redesigning the search formulation. Similarly, Marchionini [42] has a "Formulate a query" tactic in his representation of search tactics. A more in-depth discussion of various models of search tactics is provided in Section 2.2. In practice, however, such abstractions can be time- and labour-consuming to create, making them too expensive to scale well. To bring log file and tactic representations closer, attempts have been made to make tactic representations from log files through hand coding [55, 19]; while such approaches do not scale to large and dynamic log files, they are well suited to smaller lab studies. Automatic methods for tactic identification from log files using unsupervised HMM have also been proposed [26, 57] where tactics are modelled as hidden states and low-level actions are modelled as observables; however, the interpretation of the resulting hidden states remains unclear—a drawback inherent to unsupervised machine learning methods. Thus, many proposed solutions still suffer from the problems of *interpretability* and *comparability* which plague action-level log analysis. For these reasons, automatically or semi-automatically detecting tactics from log files remains a

challenging and open research problem, with a solution opening a range of new possibilities for analysis of search interaction at large scale.

In this paper, we take a step beyond the above attempts [55, 19, 26] towards a solution for analysing user search behaviour at a tactical level, specifically, we seek answers to the following research questions:

Q1 How do we identify search tactics from search logs in a *scalable* and *interpretable* manner?

To answer this research question we use log files from two interactive search systems, namely ViGOR (an exploratory video search system [25]) and Querium (an asynchronous collaborative exploratory search tool [18]), more details on both systems can be found in Section 3. Using the logs from both search systems we present methods for efficient construction of a human annotated training set for tactic identification, which reduces effort involved in determining tactics making the approach more scalable from a human effort perspective. From this training set statistical models are then learnt to automatically identify tactics that correspond to human interpretations from unseen action sequences (See Section 4). With an automatic method, our approach enables applications to scale to large logs; and by using a supervised learning method, it links logged events to states associated with human interpretations. We empirically validate the effectiveness of our tactic identification method using logs from two reference systems that are of very different designs and content types (See Section 5). One key aspect of this analysis is that we illustrate how human intervention, via training set construction, can be minimised. Meaning that this method can be applied with little extra effort above what would be normal for log file analysis, thus making our approach scalable.

Having been able to identify search tactics from sequences of logged events, we further investigate:

Q2 How do we use the identified tactics to compare user search activities across different systems?

To this end, we perform a series of illustrative analyses that compare the patterns of search tactics employed by users of our two different reference systems. In particular, we demonstrate how we can compare patterns of user search tactics quantitatively using standard statistical tools (see Section 6).

The remainder of this paper is organised as follows. In Section 2 we discuss related work in search log analysis. In particular, we discuss a range of different approaches that have been proposed to manually, semi-automatically and automatically determine search tactics from search logs. Each method has its advantages and disadvantage; we differentiate these approaches from our method, which aims to address some of the common disadvantages of the approaches outlined. In Section 3, we introduce the two search systems whose logs are used as reference data in our study. We present our search tactic identification method in Section 4. We take a supervised approach using human-annotated data, where the hidden states correspond to tactics that have explicit associations with human interpretations of the data. In Section 5, we evaluate the effectiveness and efficiency of our approach and demonstrate that powerful interpretations can be made with minimal manual effort. Further, in previous studies, analysis of user tactics have mostly focused on comparing frequencies of tactics or sequences of tactics employed. In Section 6 we provide an alternative analysis method that summarises patterns of user tactics into single statistics and allows statistical testing for comparing user behaviours across different search systems and user groups. Finally, in Section 7, we reflect on the proposed tactic identification method and tactic-based log analyses as well as providing a conclusion for the paper.

2. Related work

Search logs contain a wealth of information about search system operations and user interactions. To date, there has been a great deal of work on methodologies as well as applications of search log analysis. One common type of log analysis is focused on query and click logs, i.e. logs containing a selection of low-level actions of users. We start by providing a brief overview of research that falls into this category. We then move on to studies that analyse user search behaviour beyond the action level. In particular, we focus on research that models and analyses user behaviour in terms of search tactics.

2.1. Action level log analysis

Analyses of (Web) search logs are typically focused on a small number of logged events (i.e. user actions) such as queries, result clicks, mouse movements, as well as dwell time. These logged events are seen as indicators reflecting underlying user needs, preferences, or search approaches. A wide range of models have been developed to identify patterns from logged events and to translate these indicators to an interpretation of user search behaviour.

Queries. Individual queries have been used for classifying users' search intent. Broder [6] classified Web searches as informational, navigational or transactional based on an analysis of logged queries and user surveys. Rose and Levinson [47] expanded on Broder's classification by adding a resource-seeking category. In their expert-based log analysis, they found that navigational searches are less prevalent than generally believed, while resource-seeking queries may account for a large fraction of Web searches. Classifying intent is problematic, however, as recent research has shown [51].

Sequences of queries have been studied with the aim of abstracting away from individual queries to a higher level representation of users' search intent, tasks or missions [40, 41, 23]. In addition, patterns of query reformulations have been studied to understand users' strategies for formulating their search needs [31, 7, 5, 15, 27, 29]. He et al. [28] proposed translating various user operations on search results (e.g. filter, find similar, etc.) as direct and indirect query reformulations, creating a single representation for user interactions with results across systems with different interface designs.

Click Analysis. Using large-scale click logs from commercial search engines, researchers have developed a variety of stochastic models to capture patterns of, and to predict, user interactions, such as selection of search results or navigating between pages of search results. Examples are models for identifying patterns of typical user browsing behaviour [16], predicting user selections [17, 20, 21, 58, 10] or deriving system performance [13].

Dwell time and mouse moves. Some studies have focused on dwell time [39, 34] and mouse movements [38] as alternatives to queries and clicks. For instance, to model user behaviour with respect to clicks on relevant and non-relevant documents, Kim et al. [34] studied post-click dwell time and Guo and Agichtein [22] studied post-click mouse movement and scrolling. Lagun et al. [38] proposed to identify salient sub-sequences of user mouse movements and use these sub-sequences as a feature for improving estimation of document relevance.

While it is feasible to infer a user's short term intent or perception of relevant information from queries, mouse clicks or other user interaction described above, it is difficult to infer high-level search tactics and strategies that a user employs to advance his/her search. A common assumption shared by the above research is that users are interacting with a standard "10 blue links" search interface, with which limited user interactions are allowed. However, in a different user interface model the same observations, conclusions and models may not generalise due to the interpretability and comparability issues highlighted in Section 1.

In this paper, we present a search log analysis method that abstracts user interactions from specific actions to a higher-level representation, i.e. search tactics. This allows not only analysis of user search behaviours in context, but also comparison between user behaviour recorded in logs from different search interfaces, whether the standard "10 blue links" interface or advanced interfaces employing richer interaction models. It should be noted that while many benefits can be gained with our proposed method, it is not a replacement for the traditional log analyses. Rather, our method offers a different lens to view logs, which we believe allows exploitation of the rich resources available in log files to provide a deep understanding of user search behaviour.

2.2. Beyond actions

Search tactics

While some researchers have used log analysis to gain insight into user search behaviour, others have taken a different approach. The information seeking and information retrieval communities have long aimed to understand searchers' behaviours in terms of search tactics employed during the search process. Bates [1] originally defined search tactics as "a move made to further a search." In her seminal work on search tactics, Bates defined 29 tactics for searching library systems. These tactics ranged from "Monitoring tactics" focused on monitoring the progress of the search process to "Term tactics" focused on selection

and revision of specific terms used. The tactics Bates formulated include some that are only cognitively manifested, while others are manifested in the interaction with a search system (hence possible to infer based on information in log files).

Later, Marchionini [42] proposed an alternative model of search tactics which consists of eight stages and focuses on describing possible transitions between them. The eight stages are "recognise accept," "define problem," "select source," "formulate query," "execute query," "examine result," "extract information," and "stop reflect." In our opinion, Marchionini's model provides a better match (than Bates') for modern information retrieval, using, for instance, a Web-based search system, in comparison to Bates. Further, this model is attractive since it captures the search process on a level independent of search system or user interface.

Belkin et al. [3] applied models of information-seeking dialogues to model search in information retrieval. Here, interaction is modelled at different levels: dialogue structures, cases, and scripts. Since this model builds on a dialogue model, both user's and system's responses are modelled. Wilson et al. [54] combined search tactic models developed by Bates [1] and Belkin et al. [3] into a unified framework. This framework aimed to predict how well a system meets user needs on a tactical level. Wilson et al. applied their framework for heuristic evaluation of three faceted search interfaces to estimate how well the user interface supports different search tactics.

Tactics offer an abstraction from the implementation details of a particular search system and the ability to capture the information seeking process. That being said, it can be costly to encode user interactions into tactics, especially compared to the cheaper but less deep log analysis option. Further, the various information seeking behaviour models presented in the literature provide abstraction of user search behaviour at different levels and from different perspectives, leaving many options when choosing a target model to encode user search tactics. Similar to Wilson et al. [54], in our approach, we encode user tactics more associated with the functionality of a system, and less with the cognitive activity of the users. An important difference, however, is that Wilson et al. used experts to project their hypothesised tactics to the functionality of the evaluated systems, while we project search tactics to actual user activities recorded in log files. In practice, our work could be used to validate Wilson's work given the availability of search logs, but we reserve this for future research.

From actions to tactics

Research on users' search tactics typically relies on manual coding to map observed user actions to search tactics. More recently, a number of studies have attempted to automate this process.

Downey et al. [16] aimed to model search actions to predict what the user will do next in a standard "10-blue link" search system. For this purpose, they developed the Search Activity Model to describe the temporal dependency between user search actions and predict a user's next action. While this model does not aim to identify search tactics from usage logs, it provides a general language for describing all kinds of user browsing and searching activities with a standard Web search engine that considers user actions in context. However, as it directly models the action level patterns, the issue of comparability remains, as with other action-based user behaviour models.

Han et al. [26] proposed an automatic tactic identification approach using unsupervised Hidden Markov Models, where tactics are modelled as hidden states and user actions as recorded in usage logs are modelled as observables. Bayesian Information Criterion (BIC) was used to determine the number of hidden states that best fit the data. Specifically, in [26], 5 actions (submit query, click on search result, save a result summary, comments on a saved item, and comment on topic statement) were mapped to 5 HMM states. After the fact, the HMM states were mapped as tactics using Marchionini's ISP model [42]. Using the same approach, Yue et al. [57] modelled individual and collaborative search using the same search system. The difference to the logs in Han et al. [26] was that a chat channel was turned on during collaborative search. Two models were created: an individual search model with 4 hidden-states and a collaborative search model with 6 hidden-states. It is interesting to note that the same search system with the difference of a chat channel creates very different models using automatic statistical modelling.

Although little human effort in data processing is required, the HMM based approach suffers from issues of interpretability. The mapping between hidden states and the ISP model is based on the observation that the number of states and the transition patterns between the two models (HMM and ISP) are similar, rather than a matching between the interpretation of the HMM states and the ISP states. Also it assumes

that there is a one-to-one mapping between actions and hidden-states, i.e. one action can only be indicative of one tactic. With this model, actions that are ambiguous or context dependent would need to be filtered out before the model is built. This line of research is closely related to the work presented in this paper. However, instead of using an unsupervised approach, we take a supervised approach and look into methods for efficient manual annotation, thus the learnt mapping between user actions and tactics are associated explicitly with human interpretations.

Analysis of tactics

Multiple studies have used tactics as a means to characterise user search patterns. Of particular interest are studies focused on modelling transition patterns between tactics as a model of how users advance their search via sequences of tactics.

Chen and Cooper [12] examined the transaction logs from a university library search system. They defined search states corresponding to Web-pages in the structured search system and fitted a continuous-time semi-Markov model to these states. To group users, clustering was applied on usage characteristics of search sessions; and these clusters, or user groups, were compared in terms of the transition patterns between search states. Chen and Cooper found that the group exhibiting high interactivity with good search results had a fourth-order sequential dependency, meaning that the current state depended on the previous four states, while other groups had a third-order dependency. They also compared different user groups using the same system with a chi-square analysis on the state transition probabilities.

Using image search, Goodrum et al. [19] examined and categorised transitions between search states. In their study, the steps participants took during the search process were recorded, and these search scripts were manually coded into one of eighteen search states. To examine transition patterns between states, Maximal Repeating Pattern analysis was used, where strings of states are analysed to identify the longest repeating pattern. The identified patterns and transitions between patterns were reported with frequencies. However, no attempt was made to align these patterns with established search tactics.

Xie and Joo [55] studied transition patterns of user search tactics derived from a lab study where 31 participants each created two Web search tasks based on their own search needs using a variety of search engines and tools. User intentions were gathered using a think-aloud protocol during the search session. All search sessions were recorded using a usability testing tool and the recorded search sessions were manually coded by experts for search tactics and transitions. This method gives the researchers access to both the user's reasoning and their interactions during the search sessions. This material is much more detailed than can be expected from a typical search log and is also much more expensive to gather in terms of time and effort than a typical search log. To study transition patterns of user tactics, the authors defined a set of search tactics for Web search by extending Bates [1]'s and Marchionini [42]'s models. A fifth-order Markov chain was then applied to calculate the most common search strategies represented by patterns of tactic transition that occurred at the beginning, middle and end of each search session. They found that participants used different search tactics depending on the phase of the search session.

Similar to Chen and Cooper [12] and Xie and Joo [55], our proposed method identifies search tactics from logs, and we analyse user search tactics in terms of their transition patterns based on a Markov chain model. However, instead of a heuristic comparison of pattern frequencies, we summarise tactic transition patterns into a single statistics in terms of entropy rate of Markov chains, with which analysis can be performed with standard statistical tools.

3. Reference systems

In this section, we provide an overview of the systems we used for exemplifying our proposed method. The log data we use as an example and to validate our method are generated from Querium [18, 45] and ViGOR [25]. Both logs contain usage data collected during two different experimental studies. Each study had two conditions, one control and one experimental, where the control condition was a baseline system and the experimental condition included novel search features beyond the baseline system.

Querium. Querium is an asynchronous collaborative exploratory search tool [18]. Each search activity is organised into and shared as tasks; each task contains its own queries, retrieved documents, comments and assessments of relevance. Within each task, a searcher can run multiple queries, examine results,

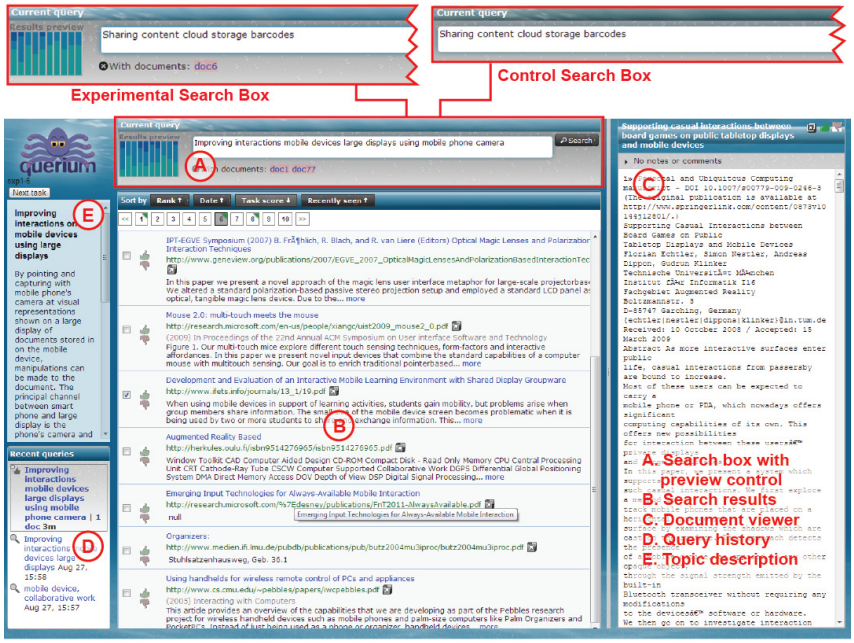


Figure 1: Screenshot of Querium interface with parts of the user interface marked. In the experiment two versions of the search box were used (top of figure).

save documents, perform relevance feedback (RF), etc. Querium makes it possible to perform relevance feedback by checking one or more check boxes next to document snippets in the results list, and re-running the search. Terms drawn from selected documents are used to expand the query.

In the present work, we used logs collected with a version of Querium particularly instrumented for studying the effect of a preview widget [45]. Querium was connected to a snapshot of the CiteSeer database of academic papers, containing about two million documents. The search UI (Figure 1) organises the display into several regions: the query area (A), the search results (B), a query history (D) and the document display area (C). PDF documents were replaced with their extracted text because the browser used in the study could not display PDFs. Finally, the task the participants were working on was shown in topic description (E). To study the effect of the preview widget, an experimental condition was compared with a control condition. In the control condition, all components except the preview widget in the query area (A) were present. The preview widget visualises the number of new documents a new query would retrieve if the user would run it, as well as the number of previously retrieved and viewed documents within the current search session.

The log of Querium contains data from 13 subjects collected while performing searches on 6 different tasks, resulting in a total of 78 search sessions. Each task lasted for about 15 minutes. The experiment used a within subject design. For more details about the experiments see [45].

ViGOR. ViGOR [25] is a video retrieval system that allows users to group video shots in order to facilitate video retrieval tasks. The aim of the system is to allow users to visualise and conceptualise many aspects of their search tasks in one workspace and to allow users carry out a localised search using groups in order to solve an overarching search problem.

Figure 2 shows ViGOR's UI which comprises of a search panel (A), a results display area (B) and a workspace (C). These facilities enable the user to both search and organise results. The user enters a text based query in the search panel to begin a search session. The result panel is where users view search results (a). Additional information about each video shot can be retrieved by placing the mouse cursor over a video key-frame for longer than 1.5 seconds, this results in any text associated with that video being displayed as a tool-tip. If a user clicks on the play button the highlighted video shot plays in a pop out window. Users can play, pause, stop and navigate through the video as they would on a normal media player. The main novel component of ViGOR is the provision of a workspace (C) for grouping and organising videos.

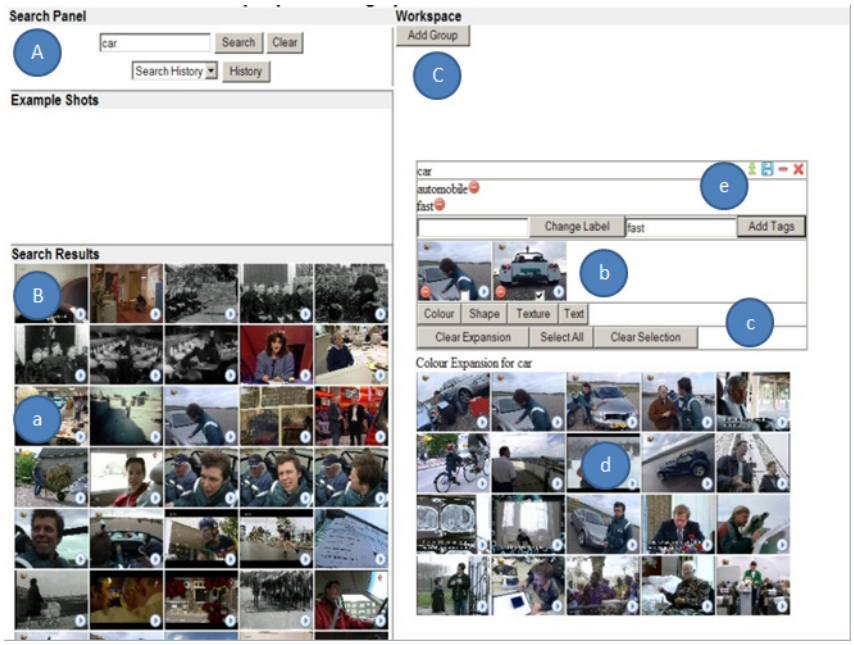


Figure 2: Screenshot of ViGOR interface with user interface parts (A: Search Panel, B: Search Results, C: Workspace) and user action functions marked (a: inspect search results, b: group videos, c & d: formulate query expansions, e: manipulating groups.)

Groups are created by clicking on the create group button. Users must then select a textual label for the group and can add any number of annotations to describe a group. Each group must have at least one annotation. Drag-and-drop techniques allow the user to drag videos into a group or reposition the group within the workspace (b). Groups can be deleted, minimised and moved around the workspace using a number of buttons (e). The workspace was designed to accommodate a large number of groups. Groups can be used as a starting point for further search queries. Users can choose to view an expansion of the group that contains similar videos based on a number of different features (c, d). ViGOR offers three expansion options for each group 1) similar colour; 2) similar shapes, which retrieves videos using edge histograms 3) and similar homogenous texture.

The log of ViGOR contains data from 24 experimental sessions with 16 participants, 8 novices and 8 experts. Expertise is based on experience of video search using TRECVID data [48]. Each novice completed 2 experimental sessions and each expert 1 experimental session. Each experimental session consisted of 4 TRECVID 2007 interactive video retrieval tasks, resulting in a total of 96 search sessions. All participants used ViGOR in two conditions. In the control condition the participants used a system resembling YouTube (i.e. ViGOR without the workspace) and in the experimental condition they used ViGOR as outlined above. The main difference between the two conditions was the availability of the workspace. For more details about the experiments see Halvey and Jose [24].

4. ESTI: Method for Efficient Search Tactic Identification

In this section, we describe a four step method for efficient development of a model for analysing search logs, with the purpose of identifying tactics. This method attempts to address our first research question, namely, *How do we identify search tactics from search logs in a scalable and interpretable manner?*

Often a search session consists of hundreds of actions and it is far from trivial for a human annotator to code tactics from actions recorded in log files with a reasonable amount of effort. With our Method for Efficient Search Tactic Identification (ESTI), we aim to reduce an annotators' effort by speeding up the process of analysing log files. Our method consists of multiple approaches at each step, each explored and designed to reduce the human effort. In Section 5, we evaluate our method and compare different approaches (manual, rule based and statistical) and give recommendations for practical use. Our intention

UserId	TopicId	Condition	Action	Timestamp
			...	...
exp1-16	1	exp	query_run	1347303423332
exp1-16	1	exp	snippet_viewed	1347303434963
exp1-16	1	exp	query_modify	1347303462150
exp1-16	1	exp	query_run	1347303465759
exp1-16	1	exp	document_assessment	1347303526516
exp1-16	1	exp	rf_query	1347303528737
exp1-16	1	exp	query_run	1347303533234
exp1-16	1	exp	rf_query	1347303533257
			...	...

Table 1: An extract from the parsed action sequence of the Querium usage log with Action and Timestamps of each action. Each entry have user (UserId), topic (TopicId) and experimental condition (Condition) noted.

is that while there are various steps in our method which will result in comparable tactic representations for different search logs, there is some flexibility in how that representation is derived depending on data, resources etc. Some of the steps in our method are prescriptive and some are achievable through different approaches. The process of identifying search tactics from log data can be summarised into the following sequential steps:

S1. *Action parsing from log data*: parse the log and prepare the data to provide necessary information for tactic identification.

S2. *Identify target search tactics*: decide on a set of tactics that are supported by the system and can be observed from system usage.

S3. *Action segmentation*: segment action sequences in a log into meaningful units e.g. that can be interpreted as search tactics.

S4. *Tactic classification*: classify the action segments into target search tactics.

The major challenges for analysing log files to infer tactics is (1) to segment the stream of actions into meaningful units and (2) to classify these action segments as a tactic. These challenges are captured in the steps S3 and S4. It is also in these steps, that we seek automatic methods to support human annotation. Below, we explain the purpose and challenges within each step. Further, using the data from the reference systems as examples, we detail our proposed tactic identification method, including the tools and algorithms we have developed and explored to facilitate each step.

4.1. Action parsing from log data

Log data comes in many shapes, for ESTI we expect that one or more people have interacted with a system to perform some search tasks, and that interactions are recorded in a log file as a sequence of discrete events (actions) of a finite number of types. The records are ordered temporally. For instance, with a typical Web search interface the types of log events could be *issue_query, click_result, click_result*. The log data is parsed into a sequence of actions to prepare it for search tactic identification. During this parsing process, any information that can provide additional information for the tactic identification is recorded. In our case, we recorded dwell time duration between actions. The sequence of actions should preserve the order of events.

Table 1 shows an extract from action sequence of the Querium log. In this extract, the user (userId) worked on topic 1 (TopicId) under the experimental condition (Condition). Within this extract, we see that the user performed the following actions: ran a query, viewed a snippet, then did a query reformulation and ran the query again, followed by a document assessment, and then he/she continued to reformulate the query by relevance feedback. Each of the events is accompanied by a timestamp, listed in temporal order, from which we derive the dwell time between events.

4.2. Identifying target search tactics

To map action sequences to search tactics, a set of *target tactics* needs to be pre-defined. This set of pre-defined tactics are used as candidate labels for tactic classification (see Section 4.4). In the classification stage, our annotators or classification algorithms will select tactics from the target tactic set as labels and assign them to action segments in the log.

Various models of information seeking behaviours have been proposed in the literature [1, 42, 3]. One may approach the selection of a target tactic model in a number of ways. For instance, from a system point of view, one can select tactics that are supported by the system features; or from a search task point of view, one can select tactics that are typically employed for a specific search task as discovered in the literature.

For this work, the target tactic model was chosen based on our system features and Marchionini's model. We expanded Marchionini's model since exploratory search systems generally include functions for organising search results. Also the tactics are at an operational level rather than at a cognitive level. Han et al. [26] adopted a similar approach of using a subset of tactics for their method which attempts to map tactics to HMM hidden states. Of course, this does not preclude any further studies from using models that include cognitive search tactics, e.g. Bates' tactics [1], with our proposed method—if information about users' cognitive activities is available and can be represented for quantitative analysis.

Further, a meaningful set of target tactics should satisfy two conditions: (1) the pre-defined target tactics should be supported by the underlying system; and (2) it should allow the identified action segments to be mapped to the target tactics. For instance, using a log with only queries recorded, we would not be able to identify actions and derive tactics relating to users interactions with the search results. Therefore, when deciding if a tactic should be included in the target tactic set, we consider whether this tactic can be associated with the observable user actions recorded in the search log. We present the target tactics we used in Section 5.2.

4.3. Action segmentation

In this step, we group the parsed actions into action segments that likely correspond to a search tactic. Critical for the segmentation is that actions are grouped meaningfully and can easily be interpreted by human annotators. If the segments are only slightly different from the raw actions, it would be hard to gain any insights into a user's tactical decisions beyond the actions employed by the user. On the other hand, action segments that are too long can be hard to interpret as representing a single tactic.

Manually going through each user action recorded in a search session and deciding when and where a segment should be formed is far from trivial. To minimise the effort in segmenting the actions, we explored alternative approaches to manual segmentation, namely heuristic rule based automatic segmentation and statistical model-based segmentation.

Manual segmentation

During manual segmentation an annotator goes through a sequence of recorded action events, and for each action decides whether or not this action is the start of a new tactic based on his/her interpretation. If there is more than one annotator, this process is typically followed by a stage where multiple annotations are compared and merged through discussion or other means to achieve agreement.

Manual segmentation is time consuming. We developed an online annotation tool (Figure 3) to assist manual segmentation. This tool allows actions to be separated based on dwell time between actions on a global and local level. The slider can be used for setting different thresholds on the dwell time for a global split. A human annotator can then look over the resulting segments and combine or further split specific segments by clicking on the white segments representing dwell times in the user interface. The actions are colour-coded, where semantically related actions are assigned similar colours to provide annotators with a better view of related actions.

Our experience was that this manual segmentation tool speeds up the segmentation process to some extent, since correcting errors was faster and easier than identifying all segments. Further, this process allows the annotators to analyse each individual action or combination of actions in detail and in context. The drawbacks of this approach are, however: (1) similar to previous methods [12, 55, 19], it is labour intensive, although a labelling tool helps to better visualise and manipulate the sequences; (2) a strategy to resolve disagreement between annotators is needed. To address these shortcomings we explore alternative approaches, namely heuristic rule-based and statistical segmentation.

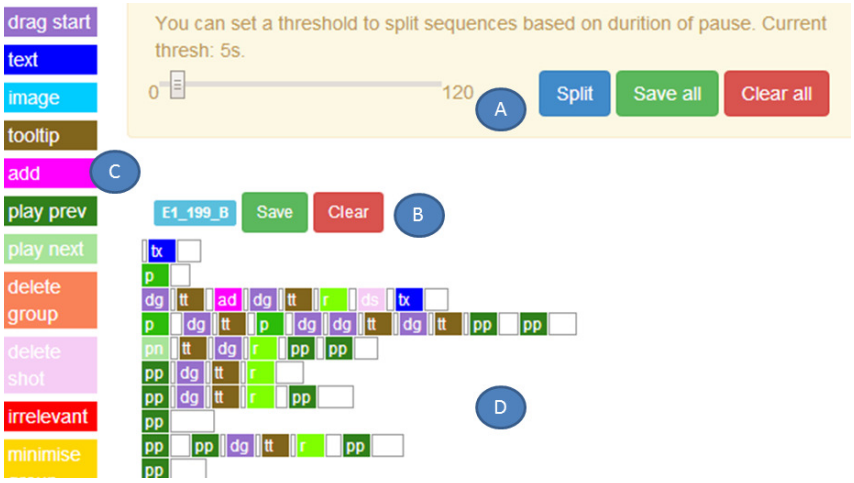


Figure 3: Screenshot of our segmentation tool. User can set a threshold in seconds for splitting segments (A). The user can then view segments for each log (D). If segments are not satisfactorily split, they can be combined or further split by clicking on white segments between or after actions. A legend describes the coding scheme for the actions to facilitate interpretation of segments (C). A user can save or clear segmentation if they wish (B).

Heuristic rule based segmentation

Segmentation can be determined by a number of factors, such as the dwell time duration before or after an action, combination patterns of actions, etc. One simple segmentation rule is to use a threshold on dwell time, e.g. one may assume that a user would start doing something different after a long period of inactivity. Similar heuristics have been used for identifying query sessions in other query log analyses [9, 49]. However, when looking closely into the resulting action segments using this approach, we found that thresholding durations was fairly error prone. Sometimes even a quite short dwell time could indicate a shift in tactic, and sometimes a long dwell time did not indicate a shift in tactic. Further, each action in our two reference logs has a different distribution of dwell times, e.g. it generally takes longer for a user to play a video than to do a pagination. Therefore we found that a global time thresholding was not likely to work.

Hence we explored heuristic rules that are more context dependent than a simple dwell time threshold. The annotators discussed and agreed on strategies to segment the action sequences based on their observations and interpretations and formalised them into as set of guidelines (L1 — L4 below) for automatic segmentation. To begin with, each action was assumed to formulate an individual tactic. We then derived a guideline, identifying four situations when consecutive actions can be merged into a segment. Although this guideline was developed with our two example systems in mind, we believe it applies to other search systems as well.

L1. *Repeated actions.* Users sometimes repeatedly perform an action, e.g. continuous book-marking of relevant documents or continuous pagination for a quick scan through the results. In these cases users do not switch tactics. However, caution needs to be taken as actions can sometimes be ambiguous—with the same recorded action event the user may be doing different things given different context, e.g. the dwell time after/before the action. See also discussion in Section 4.4.

L2. *Semantically close actions.* Users may perform actions that serve similar purposes. For instance, users may attempt to reformulate their queries by performing relevance feedback or by directly modifying their query in the query box. While a user may switch between these two semantically close actions (i.e. relevance feedback and revising query), the user does not change tactic.

L3. *Fixed combination of actions.* Some systems may require users to perform a series of actions in a particular order to achieve a certain goal. For example, with the ViGOR system, users can drag and drop a video shot to a relevance feedback panel for the purpose of performing relevance feedback. Hence, the combination of actions “drag-drop-a-shot” and “add-shot-to-panel” is frequently occurring.

L4. *Unintentional pre- and -post actions.* Some recorded action events can be triggered unintentionally. This is often a result of a specific design of the interface and the logging system. For instance mouse hover over a snippet is a commonly recorded event in our logs. Our reference system Querium (Figure 1), has a "thumb-up/thumb-down" button in front of each snippet. When users attempt to select this button, it is very likely that they would move the mouse over the snippet, which may unintentionally trigger a mouse-hover-over-snippet event. Unintentionally triggered actions should be merged as pre- or post-fix of the intended action. However, a mouse-hover event with a longish pause is likely an intentionally triggered event.

With the above guidelines, heuristic segmentation rules were generated by analysing merging conditions between each of the action events as identified from the action parsing step. The resulting rules are specific to each of the systems being analysed. We specify the rules discovered from our reference logs in Section 5.

Compared to manual annotation, a heuristic rule-based approach saves the effort of manual marking every segment boundary. However, the derived rules' completeness depends on how many action sequences the annotators have seen in a log, as well as how much the action sequences vary within a log. For a small log, the annotators can scan through all sequences and derive a relatively complete set of rules. For a large log, e.g. a log of Web scale, it would be difficult to derive a complete set of rules, unless users are provided with very limited interaction modes, and behave more or less the same. In addition, these rules can be combined with manual correction to annotate a set of training data, which can then be used to learn more sophisticated models which are more likely to generate to unseen cases. Below, we discuss one type of such models based on statistical learning.

Statistical model-based segmentation

An alternative method to automate the action segmentation process is to train a statistical model on action segments. While this is possible, classifying action segments identified with manual or heuristic methods may not be the most efficient approach. Such an approach would require yet another statistical model for classifying tactics from action segments. Instead, we define a statistical model that performs the segmentation and search tactic classification in one step. In Section 4.4 where we discuss tactic classification we describe the statistical model we used to do this.

4.4. Tactic classification

After obtaining the action segments we can assign them to a predefined target tactic. In practice we identified the following three situations:

one-to-one mapping. In this case, one action corresponds to one particular search tactic. However, if all action segments have a one-to-one relation with a target tactic, then the task of classifying search tactics is trivial and analysis based on tactics would not provide insights beyond the action level.

many-to-one mapping. In this case, a tactic can correspond to multiple unique action segments. This means, a user can achieve the same goal via different types of interactions with the system. For example, users can reformulate a query by applying relevance feedback or directly modifying the query text. In these cases the different types of actions are *unified* to the same tactic.

one-to-many mapping. In this case a single action can be interpreted as different tactics given different context. For example, mouse hovering over a result summary may indicate the user is inspecting the summary; it can also be that the user is skimming through the result list and the mouse quickly moves across the summary; or it may be unintentionally triggered because of the design of the interface and logging set-up as discussed in L4. In cases like this, actions need to be *disambiguated* by taking into account context for a better interpretation in terms of search tactics.

By considering these different mappings we provide a more realistic representation of user behaviour compared to approaches that assume a one-to-one mapping [26, 57].

Manual classification

Similar to the manual segmentation process, we refer to the manual classification process as individual annotators assigning each action segment to a predefined target tactic based on his/her interpretation. This shares the same pros and cons as the manual segmentation process. In fact, the annotators may perform this task in parallel with action segmentation: when deciding if a series of consecutive actions should form an action segment, he/she may have made an interpretation of *which* target tactic it corresponds to.

Heuristic rule-based tactic classification

Our approach to heuristic tactic classification consists of a general procedure and naturally, a set of specific rules. The general procedure has the following steps. (1) Classify single actions: each action of the users can be associated with an interpretation of the type of the tactic the user is applying; since some actions are ambiguous, context is needed to determine its label. (2) Classify action segments based on interpretations of single actions: Most likely, the meaning of a combination of actions would not be independent from that of the individual actions it contains. However, depending on specific combinations (e.g. cases L1 — L4 as listed in Section 4.3), different rules may apply. For situations L1 and L2, the segment label has the same label as its individual actions, as all actions in the segment would have the same tactic label. For L3, specific rules need to be formulated case-by-case. For L4, the segment label takes the label of its "main action", and similarly, rules to identify the "main action" need to be formulated on a case-by-case base.

To apply the above procedure, a set of specific rules need to be specified, including:

(a) the mapping between each individual action and the predefined target tactics;
(b) rules to disambiguate ambiguous actions;
(c) rules that handle L3 segments; and
(d) rules that determine the "main action" from an L4 segment.

These rules are specific to each system and its log. In Section 5, where we examine and validate our approach, we describe the specific rules derived for our reference logs.

Statistical model-based tactic classification

Given an *unsegmented* sequence of user actions, our goal is to assign each of the actions to one of the target tactics. This way the classifier simultaneously determines the boundary of the action segments and their corresponding tactic types. That is, an action segment contains consecutive actions that are assigned with the same target tactic. Labelled training data is used for identifying features of the action and its context, i.e. its precedent action, to create a model for assigning tactics to actions.

We start with a formal statement of the labelling problem. Let $A = a_1, \ldots, a_n$ be an unsegmented but observed action sequence of length n, and $\mathcal{T}$ be the set of target tactics identified in a log. Also, let $T = t_1, \ldots, t_n$ be an unobserved sequence of tactic labels, where $t_i \in \mathcal{T}$, and t_i corresponds to the tactic assignment to action a_i. To construct a model that predicts the mapping between A and T, we need a set of labelled training data D that consists of a set of A's, annotated with their corresponding T's.

Many approaches have been proposed to solve this kind of sequence labelling problem, e.g. hidden Markov models (HMMs), conditional random fields (CRFs), structural SVMs [44] etc. The performance of these different approaches depends on the specific task [44] as well as the implementation details of the methods [33]. An extensive comparison of different approaches and their implementation details is out of the scope of this paper. We choose a CRFs [37] approach for our labelling problem. It has shown state-of-the art performance in a wide range of similar applications [37, 33], and has the flexibility of allowing rich feature integration when compared to a generative model such as HMMs. Specifically, we define a conditional probability $P(T|A)$, i.e. the probability of observing an assignment T given action sequence A as follows:

$$p(T|A) = \frac{1}{Z(A)} \exp \sum_i^n \sum_k^m \lambda_k f_k(t_{i-1}, t_i, a_i, i). \quad (1)$$

Here, $f_k(\cdot)$ is a feature function, which describes a property of the action at position i, its corresponding tactic, and its previous tactic at position $i-1$. For instance, a feature describing the transition between t_i and t_{i-1} can be $f = 1$ if $t_i = x$ and $t_{i-1} = y$. Each feature has a weight λ. Using the same example, a

large positive weight means it is very likely that a tactic x is followed by a tactic y. The term $Z(A)$ is a normalisation factor that makes sure the likelihood is between 0 and 1.

Given D, the goal of the learning procedure is to find a set of λ's that maximise the log-likelihood of D (as summed over all sequences in D). With the learnt weights, Eq. 1 can be used to score and find the best labels for an (unseen) action sequence.

4.5. Summary

Our method for identifying search tactics from usage logs consists of four steps, namely, action parsing, target tactic identification, action segmentation and tactic classification. In terms of action segmentation and tactic classification, annotators have the choice between manual, heuristic rule based and statistical model-based labelling. These three options have their own advantages as well as disadvantages. Manual labelling is likely to be the most accurate, but requires large effort and post-processing for resolving disagreement and inconsistencies. Heuristic rule based labelling saves the manual effort of inputting labels. However, specific rules need to be formulated case-by-case, and its completeness depends on both the experience of the annotators as well as the variability of user search behaviour. A statistical model-based approach is expected to have better generalisability compared to a rule-based approach with regards to unseen cases. However, its performance depends on the quality of the training data, which is a result of manual or rule-based annotation, or the combination of the two. No matter which approach is taken, some manual annotation is necessary, as a starting point to derive heuristic rules, or to establish a training set for learning a statistical model. Next, we validate and examine our ESTI method by analysing the two search logs described in Section 3. In particular, we compare the different approaches for action segmentation and tactic classification.

5. Validation of ESTI with reference logs

In this section we demonstrate how we can apply the ESTI method to analyse search logs and validate its efficiency and effectiveness. We examine the trade-offs of the different options that are available within our proposed method. In particular, for action segmentation and tactic classification, we illustrate how these can be conducted in three different ways (manual, rule-based and statistical model-based approaches). We created a ground-truth by annotating the logs from our reference systems in order to evaluate the rule-based and statistical model-based approaches.

Three annotators were involved in the annotation process; one annotator had been involved in developing Querium, one involved in developing ViGOR, while the third annotator was independent from development of both systems. All annotators were in different geographical locations. The system expert annotators were responsible for explaining the system design and the set-up of the logging systems to the other annotators.

5.1. Action parsing of the reference systems' logs

The logs from the two reference systems were parsed as described in Section 4.1. Table 2 (Querium) and Table 3 (ViGOR) show the actions identified in the logs for each references system.

Of the nine actions in Querium, eight correspond directly to events in the log files. The remaining action, QM (modify query), was recorded in the log when a dwell time longer than two seconds was recorded when entering text in the query box. Querium includes a preview widget tied to the query box. A two second pause is used to trigger an update of the preview widget, hence this "composite" action.

In ViGOR, 17 actions are possible in both conditions, while five actions are only available in the experimental condition. The first three in Table 3 correspond to executing a search using combinations of textual and visual queries. ADD and DEL relate to adding and removing visual information to and from a query. REL and iREL denote actions for marking videos as relevant or removing videos from the list of relevant videos. The DRA action indicate start of a drag action to reposition a key frame in the user interface. The remaining actions (NEX, PL and PRE) relate to playing and navigating videos. The five additional actions in the experimental condition all relate to manipulating the workspace available in that condition.

Action	Description
QR	Excute a query
QS	Select query from history
MD	Mark document (as relevant or irrelevant)
VD	Open a document and view
VS	View a result summary (mouse hover summaries)
RF	Select a document for relevance feedback
PG	Pagination
SR	Resize the document or expand/minimize summaries
QM	Modify query. This is triggered by user typing in the query box. A QM is recorded if there is a pause over 2 seconds between keystrokes.

Table 2: Actions recorded in Querium log.

Action	Description
I	Execute a search using example video shots
T	Execute a search using text
B	Execute a search using both text and example video shots
ADD	Add a video shot to a relevance feedback panel to be used for relevance feedback
DEL	Delete a video shot from the relevance feedback panel, this is the opposite of ADD
DRA	Drag a video shot from somewhere in the interface
NEX	A video consists of a sequence of shots, this moves to the next shot in a playing video
PL	Play a video
PRE	Move to the previous shot in a playing video; this is the opposite of NEX
REL	Mark a video shot as relevant by adding it to a group or to the relevant panel
iREL	Delete a video shot that is marked as relevant from a group or the relevant panel, this is the opposite of relevant
tTIP	When a video shot is moused over, a box containing text describing the video pops up
Extra actions under experimental condition	
CreG	Create a group in the workspace
DelG	Delete a group from the workspace
MaxG	Maximise a group, like a window from a tool bar
MinG	Minimise a group, like a window to a tool bar
MovG	Move a group in the workspace

Table 3: Actions recorded in ViGOR log.

5.2. Target search tactics for the reference systems

Next, the target tactics need to be specified. We adopted the model by Marchionini [42]. This model served our purposes since it has tactics that can be manifested by actions recorded in the log files. Specifically, we used four of the eight stages; "formulate query (FQ)," "execute search (ES)," "examine result (ER)," and "extract information (EI)." The other stages are either not applicable for our data (but could easily be applied in other situations), or do not have physical representation as an action recorded in a log file.

Since modern search systems may allow user interactions beyond the tactics defined in the literature the tactics may need to be extended. Both Querium and ViGOR include one additional tactic not covered by Marchionini. For Querium, we added a tactic "Review History" (RV). Querium includes a UI feature that allows quick and easy access to previously run queries within a search session. For both systems, we also added a tactic "Organisation" (ORG), since ViGOR in the experimental version included features for organising relevant video clips within a workspace, and Querium allows users to resize the documents or result summaries for a better view. Table 4 lists the target tactics defined for our reference logs. In practice

Tactics	Description	Examples (Querium)	Examples (ViGOR)
FQ	Formulate/refine a query	RF+ QM*; QM+;	DRA ADD; DEL+
ES	Execute search	QR+	I+; T+; I+ B+;
ER	Examine results	VS+ (short dwell time); PG+	NEX+; PRE+; DRA+ tTip
EI	Extract information, e.g. relevance assessment, review results	VS+ (long dwell time); VS MD+	PL+; DRA iREL; DRG REL
Extra	tactics		
ORG	Organising results	SR+	[MinG MaxG MovG]+
RV	Review history	QS+	–

Table 4: Target tactics identified in the two logs. Example actions or action segments are written as regular expressions and separated by ";". For example, RF+ QM* means one or more RF, may or may not followed by one or more QM.

we also defined a category "O" meaning "out of vocabulary", in case some of the action segments cannot be interpreted as any search tactic.

As discussed in Section 4.4, there exist many-to-one as well as one-to-many mappings between actions and tactics. In Table 4, we see a typical one-to-many example from Querium: View Summary (VS) can either be Extract Information (EI) or Examine Results (ER). To correctly assign this action to a tactic, dwell time duration is used. If the dwell time is long, i.e. ≥ 5 seconds, it is more likely that a user is reading the search results summary (ER) rather than only skimming it (EI). The 5 second threshold was determined based on reading research using eye tracking [46] and the size of the summaries in Querium. A time span of less than 5 seconds is a too short period for being able to read a summary and extract information.

5.3. Validating action segmentation

The ground-truth for action segmentation was created using the segmentation tool as shown in Figure 3 and the set of heuristics developed (as discussed in Section 4.3). After the heuristic rules had been applied, the action segments were uploaded to the segmentation tool and the annotators manually reviewed and corrected the segments. With this ground truth data-set we can evaluate the effectiveness of both the heuristic rules and the statistical model-based approach. Note, our selected statistical method performs both segmentation and labelling, therefore validation of this method will be discussed in Section 5.4.

Heuristic rule based segmentation

Following the approach described in Section 4.3, our annotators formulated specific heuristic rules for each of the reference logs based on the general guidelines outlined in Section 4.3. The rules specified for Querium are listed in Table 5, and the rules for ViGOR are listed in Table 6. Guideline L1 is a straightforward rule that can be applied to any actions. However, note that certain actions may be ambiguous, e.g. VS in Table 5, in which case L1 cannot be applied without considering the context. Here we only mention guideline L1 as special cases as such, otherwise we only list rules corresponding to guidelines L2 — L4, which deal with merging conditions between *different* actions.

Performance of rule-based segmentation. The ground truth data contained 5852 action segments from the Querium logs and 12,437 action segments from the ViGOR logs. The identified segments from Querium and ViGOR contain on average 1.8 and 2.3 actions respectively.

To observe the effectiveness of the heuristic rules, we compare the result of rule-based segmentation to the ground truth, i.e. segments after manual correction. For the purpose of evaluation, we assign each action to one of the two classes: "SP" (splitting point) or "Non-SP" (non-splitting point). That is, if an action is labelled as "SP", then it is the start of a new segment. We report precision, recall, and F1 for each class over all actions in a log, as well as the micro and macro average over classes (Table 7).

Table 7 shows that the heuristic rules achieved very accurate results, i.e. > 0.95 precision and recall. However, our logs are relatively small, and our annotators had gone through almost all sessions in each log before formulating the rules, therefore these rules are rather complete and accurate. In other words, we can treat this result as an upper bound that could be achieved with a heuristic rule-based approach.

Action	Rule	Guideline
QS	Alone or combined with following VS: viewing a result summary after selecting a historical query can be seen as a whole process of reviewing history.	L2
MD	Alone or combined with a precedent VS, regardless of the dwell time of VS: users need to move mouse over a result summary to mark the document, which most likely will trigger a VS.	L4
VD	Alone or combined with precedent VS: users need to click on a result summary to read the document, which will most likely trigger a VS.	L4
VS	1. Repeated VS's are grouped in long viewing and short viewings (i.e. depending on the dwell time after a VS as discussed above.)	L1
	2. Follow rules of combining with other actions.	L4
RF	1. It can be combined with QM, as RF can be seen as another form of QM;	L2
	2. VS with short dwell time may be intermixed, as VS may be triggered when users move the mouse over the summary to click on RF checkbox.	L4
PG	Alone or combined with VS with short dwell time: both actions indicate skimming of the results.	L2
SR	Alone or combined with following or precedent VD's: SR is performed to make the document window larger for easy reading.	L3
QM	Alone or combined with RF: semantically close actions.	L2

Table 5: Heuristic rules to segment Querium log with reference to general heuristic guidelines outlined in Section 4.3.

Actions	Rule	Guideline
I,T,B	Alone or combined with each other: there are three ways of issuing a query.	L2
ADD, DEL	Alone or combined with DRA, tTip, or their combinations: to add/delete or label relevance/irrelevance users need to drag-and-drop clips.	L3
DRA	Only meaningful in combination with other actions: users drag-and-drop a video clip or group to achieve a particular goal, e.g. to provide relevance feedback or manipulate a group.	L3
PRE, NEX	1. Alone or combined with each other: both are used for skimming a video clips.	L2
	2. Can intermixed with tTip, DRA, or their combination: tTip and DRA can be triggered when users click to play next/previous click.	L4
REL,iREL	The same rule as ADD/DEL, for the same reason.	L3
tTIP	Often combined with DRA: when drag-and-dropping a clip, users moves mouse over a clip, which can easily trigger the tTip.	L4
CreG, DelG, MaxG, MinG, MovG	1. Alone or combined with each other: they all relate to result organisation.	L2
	2. Can intermix with DRA: manipulating groups can trigger a "drag and drop" action.	L3

Table 6: Heuristic rules to segment ViGOR log with reference to general heuristic guidelines outlined in Section 4.3.

Segmentation rule coverage. To simulate the case when annotators have not or cannot look over all log entries we conducted the following experiment. The data was randomly split into 5 folds. 20%, ..., 100% of the data was used as an "observed set" to simulate the rule generation process. For this evaluation, we assume that if a situation described by a rule occurs in the observed set, then the annotator will identify this rule. By examining the coverage of the rules discovered in the observed set and comparing this to the "ground truth," we can see how many rules would have been discovered using a fraction of the logs. We performed the experiment as a 5-fold cross-validation, Table 8 reports the averaged number of discovered

	Querium			ViGOR		
Classes	Precision	Recall	F1	Precision	Recall	F1
SP	0.965	0.983	0.974	0.993	1.000	0.997
Non-SP	0.978	0.956	0.967	1.000	0.995	0.998
Avg (macro)	0.971	0.969	0.970	0.997	0.998	0.997
Avg (micro)	0.971	0.971	0.971	0.997	0.997	0.997

Table 7: Accuracy of rule-based segmentation after manual correction, in terms of precision, recall, and F1-measure. Each action is labelled as "SP" (split point) or "Non-SP" (non-split point).

Log	20%	40%	$\geq 60\%$	ground truth
Querium	7.6	7.75	8.0	8
ViGOR	12.0	12.0	12.0	12

Table 8: Average number of segmentation rules discovered from the "training sets" of different size.

rules over the 5 folds. For ViGOR, 20% of the data would discover all rules, while for Querium we may miss 1 rule when observing only 20% or 40% of the data, depending on the sample from which rules are discovered. This indicates that, at least for our reference logs, a small amount of data can cover the majority of the rules.

Thus, we have learnt that: (1) It is possible to formulate simple rules to achieve an accurate segmentation on action sequences. With a small set of sequences it is possible to derive rules with good coverage. This can greatly reduce the manual effort for annotation. (2) These rules are however not perfect. If necessary, the action segments can be manually corrected, e.g. to obtain a ground truth set for developing statistical learning models. This correction is, in our experience, faster than manually segmenting the logs.

5.4. Tactic classification for the reference systems

Having obtained action segments we can now map them to the target tactics. As with action segmentation we use the heuristic rule based approach combined with manual correction to create a ground truth set. With this set we then evaluate the effectiveness of the heuristic rules as well as the CRF-based automatic approach.

Heuristic rule based tactic classification

Following the procedure described in Section 4.4, four sets of specific rules need to be formulated in order to assign action segments to target tactics for our reference logs. These rules were summarised by annotators based on their observations and interpretations during the action segmentation stage.

Rules for Querium.

(a) Interpretation of individual actions. Table 9 lists the rules for mapping individual actions to target tactics. These rules can be overridden by rules in (b) — (d).

(b) Disambiguation rules. As already discussed in Section 5.2 we set a threshold of 5 seconds for the dwell time after SV to indicate short/long SV for the purpose of disambiguation. In some cases dwell time is not important, i.e. this rule can be overridden by rules (c) — (d).

(c) Rules for fixed combinations (L3 segments). Based on Table 5 we have the following L3 case for Querium: *VD combined with SR*. When SR follows VD it is most likely that SR is to make the window larger for easy reading, while the main purpose of the reader is to view a document (VD), which corresponds to target tactic EI.

(d) Rules for unintendedly triggered actions (L4 segments). As listed in Table 5 we have the following cases where L4 applies: *VS followed by MD, VD or RF*. Most likely VS is unintentionally triggered when users attempt to mark a document, open a document, or select relevance feedback. Therefore in these cases the segment takes the interpretation of MD, VD, or RF.

Action(s)	Tactic	Reasoning
QR	ES	QR runs a query, and therefore executes a search.
QS	RV	QS selects and executes a query from historical queries.
MD	EI	MD provides relevance assessment to documents, this is a type of information extraction.
VD	EI	Users extract information by viewing a document.
VS	EI/ER	For VS with short dwell time, users are likely to skim through a result summary, which would be "examine result" (ER), while for VS with long dwell time, users may be reading the result summary, and hence extract information (EI).
RF, QM	FQ	Both actions provide (re)formulation of a query, i.e. FQ
PG	ER	The purpose of pagination is to examine results (ER).
SR	ORG	Resizing the document window can be seen as organising results.

Table 9: Heuristic rules for mapping individual action to target tactics (Querium).

Action(s)	Tactic	Reasoning
I, T, B	ES	These actions are three different ways to execute a query.
ADD, DEL, REL, iREL	FQ	All four actions are for performing relevance feedback, which provides a way to (re)formulate a query.
DRA	—	This action is not meaningful alone, it has to be combined with other actions.
PL	EI	By playing a video, users extract information.
PRE, NEX	ER	These actions provides a means to skim through results, i.e. examine results.
tTIP	ER	This action alone can be interpreted as examining a result's text description.
CreG,DelG,MaxG, MinG, MovG	ORG	All these actions are for organising results in terms of groups.

Table 10: Heuristic rules for mapping individual action to target tactics (ViGOR).

Rules for ViGOR.

(a) Individual actions. Table 10 lists the rules that map individual actions to predefined target tactics. These rules can be overridden by rules in (b) — (d).

(b) Disambiguation rules. Not applicable for ViGOR. The only ambiguous action in ViGOR is "iTip", which may be unintentionally triggered when performing other actions. The interpretation of "iTip'" can be determined by its contextual actions, using rules (c) — (d).

(c) Rules for fixed combinations (L3 segments). From Table 6 we see the following L3 combinations of actions:

ADD, DEL, REL, iREL combined with DRA. Users need to drag-and-drop to perform ADD / DEL / REL / iREL to perform relevance feedback. Therefore these combinations are interpreted as QF. While drag-and-drop, tTIP may be triggered, and can be considered as an unintentional event and can be ignored.

CreG, DelG, MovG, MaxG, MinG combined with DRA. Users sometimes drag-and-drop groups in order to move them around while performing other operations on groups. Together they can be seen as organising tactic (ORG).

(d) Rules for unintendedly triggered actions (L4 segments). In terms of L4 rules, we have the following cases.

tTip combined with DRA. When a user drag-and-drops a clip, iTip is easily triggered as users need to position the mouse over the clip to perform DRA. In such cases iTip is unintentional and can be ignored.

tTip and DRA combined with PRE and NEX. When clicking to play the previous or next clip users sometimes unintentionally move the clip and trigger the drag-and-drop or tooltip event. In these cases, these actions are followed PRE or NEX and can be interpreted as ER.

Performance of rule-based tactic classification. To evaluate the effectiveness of the above rules, we created a ground truth set by manually correcting the labels after applying the rules. We then compare the output of the rule-based classification to the ground truth set. Table 11 and Table 12 list the results in terms of the precision and recall of each tactic category. The overall micro performance for precision, recall and F1 is high, about 0.98 for both systems. While macro performance takes the average of all classification classes, micro performance is measured over all instances whose values are influenced by the dominant classes. As Table 11 and Table 12 show the macro performance is lower than the micro performance. The performance difference can be traced to the low performance of the "O" cases, particularly with Querium. This is because with Querium no rules covered these cases, and only in manual correction did the annotators discover and decide to label these cases as "O". However, as the micro performance indicates, the "O" cases were quite infrequent and hence had very little impact on the micro performance.

Further, we observe that when an action can be interpreted as multiple tactics depending on how it is combined with other actions, rules tend to make errors. For example, we see that in Querium both RV and ORG have a relatively low precision compared to other tactics. A closer investigation reveals that the mistakes involve recognising ER as RV and EI as ORG, the problems being: QS can be interpreted as RV when standing alone, but also ER when combined with short SV; while SR can be interpreted as ORG when standing alone, but EI when combined with VD.

Classes	Precision	Recall	F1
FQ	0.965	0.999	0.982
RV	0.773	1.000	0.872
EI	1.000	0.983	0.991
O	0.000	0.000	0.000
ORG	0.811	1.000	0.896
ES	1.000	1.000	1.000
ER	0.985	0.973	0.979
Avg (macro)	0.791	0.851	0.820
Avg (micro)	0.983	0.983	0.983

Table 11: Accuracy of rule-based tactic labelling after manual correction in terms of precision, recall and F1-measure; Querium.

Classes	Precision	Recall	F1
FQ	1.000	0.992	0.996
O	0.941	1.000	0.970
EI	1.000	0.996	0.998
ORG	1.000	1.000	1.000
ES	1.000	1.000	1.000
ER	0.995	1.000	0.997
Avg (macro)	0.989	0.998	0.994
Avg (micro)	0.998	0.998	0.998

Table 12: Accuracy of rule-based tactic labelling after manual correction in terms of precision, recall and F1-measure; ViGOR.

Tactic identification rule coverage. We also evaluated the coverage of the rules discovered from partially observed data. Table 13 lists the results. Assuming the annotators can always identify a rule if the situation described by the rule occurs in the "observed set", we see that for Querium, all the rules can be discovered if only 20% of the data is inspected. For ViGOR, one rule may be missing if only 20% of the data is inspected depending on the sample. That is, similar to the segmentation rules, with our reference logs, a small portion of the data covers the majority of the rules discovered from the complete set.

CRFs-based tactic identification

Next, we move on to the application of the statistical model-based approach, following the CRF model specified in Section 4.4. We use the CRF implementation provided by crfsuite.[1] In terms of features, we apply the simplest features for a linear chain model: the transition relations between tactics and the emission relations between tactics and actions. These two types of feature can be directly observed in any logs. It is possible to consider more complex features if more information is contained in or can be derived

[1] http://www.chokkan.org/software/crfsuite/

Log	20%	40%	$\geq 60\%$	ground truth
Querium	10.0	10.0	10.0	10
ViGOR	17.8	18.0	18.0	18

Table 13: Average number of rules discovered from the "training sets" of different sizes.

from a log, but for this work we have concentrated mainly on simple text based logs as they are most prevalent. Further, with a linear-chain model we implicitly assume that the sequence of user search tactics constitute a first-order Markov chain. In Section 6 when using search tactics to analyse user behaviour, we will provide a validation of this property.

Here, we are interested in two aspects of the result of CRF-based tactic identification: (1) To which extent can the statistical model-based approach correctly segment and identify tactics for a log? (2) How well can a statistical model generalise when the training data does not contain all possible patterns?

To answer the first question we ran a 5-fold cross validation and used precision, recall and F1-measure as evaluation metrics to assess classification performance. We report the averaged values of the metrics over the cross validation folds.

To answer the second question, we experiment with different divisions between training and testing data by adjusting the amounts of holdout data during cross-validation. With the 5-fold cross validation, we experiment with training on k folds, and test on the rest, where $k \in [1, .., 4]$.

Classes	Precision	Recall	F1
FQ	1.000	0.982	0.991
RV	1.000	0.992	0.996
EI	0.942	0.909	0.925
O	1.000	0.714	0.833
ORG	0.843	0.983	0.908
ES	1.000	1.000	1.000
ER	0.895	0.938	0.916
Avg (macro)	0.954	0.931	0.943
Avg (micro)	0.939	0.939	0.939

Table 14: Accuracy of CRF-based tactic labelling in terms of precision, recall and F1-measure; Querium.

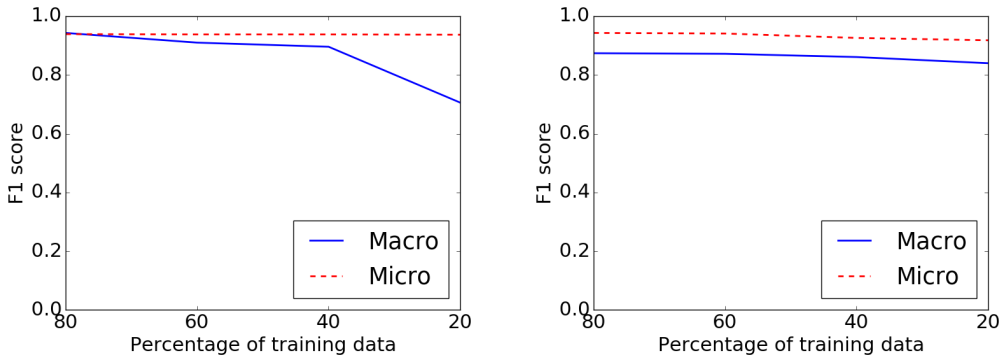
Classes	Precision	Recall	F1
FQ	0.903	0.921	0.912
EI	0.892	0.990	0.938
O	1.000	0.021	0.041
ORG	1.000	0.964	0.982
ES	1.000	1.000	1.000
ER	0.986	0.905	0.944
Avg (macro)	0.964	0.800	0.874
Avg (micro)	0.943	0.943	0.943

Table 15: Accuracy of CRF-based tactic labelling in terms of precision, recall and F1-measure; ViGOR.

Performance of CRFs-based tactic identification. Table 14 and Table 15 show the performance of the CRF based tactic identification approach in terms of precision, recall and F1 measure. From our results the following can be observed. First, the overall micro performance is quite promising (> 0.93). The macro recall and F1 is lower for ViGOR, which can be traced to the "O" class. In ViGOR, the "O" class consists of patterns that can be easily confused with EI and ER (e.g. the combination of "DRA"s and "tTip"s). Also, the number of examples for training of action segments belonging to the "O" class is much lower than for the other classes. Also for Querium, recall and F1 is lower than for the other classes due to fewer training examples. Second, if we compare the performance here to the performance of rule-based approach (Table 11 and 12), we see that it is hard to conclude which approach is better. In four out of seven cases the rule-based approach leads to better F1 scores for both logs. "ES" seems to be an easy tactic to identify, both approaches result in a F1-measure of 1.0. In the rest of the cases the CRF-based approach actually achieved better performance in terms of F1-score.

Here, the rules are formed based on extensive analyses of the logs, and the statistical models are trained on 80% of the data that have been manually annotated. That is, for these experiments extensive manual effort has been involved in both approaches. However, we have already seen that it is possible to identify the majority of the rules needed from a small proportion of the data. Next, we investigate if such effort can further be reduced by the statistical model-based approach.

Generalisability. Figure 4 shows the performance of the CRF-based tactic identification method trained



(a) Querium (b) ViGOR

Figure 4: Performance of automatic tactic identification with different amounts of training data.

on different amounts of data in terms of F1-measure. We see that when decreasing the amount of data used for training from 80% to 20% (and correspondingly increasing the amounts of data used for testing), the performance of the classifiers remains stable. The only exception is the relatively obvious drop in macro averaged F1 score for Querium when 20% of the training data is used, which is due to the absence of the very few "O" cases in the training data. This does not affect the micro averages where results are controlled by the dominant classes.

This suggests that the effort of manual annotation can be greatly reduced as the CRF-based tactic identification approach only requires a small portion of the action sequences labelled. For example 20% of the data only involves about 15 sequences for Querium and 19 sequences for ViGOR.

5.5. Summary and recommendation

In this section we have illustrated how the tactic identification method presented in the previous section can be applied in practice, using our reference logs as examples. The various steps of our method were evaluated and validated. Specifically for action segmentation and tactic classification we have empirically compared the performance of different approaches (heuristic-rule based and statistical model based).

Based on our evaluation, we have the following recommendations for choosing an analysis approach. If the log is small (like ours), perfect and efficient annotation can be achieved by using a heuristic rule-based approach combined with a manual correction process. Our manual segmentation tool can further facilitate this process by providing visual assistance and easy operation on sequences. If the log is large, a small training set can be created with the above procedure, and a statistical model can then be trained and used to predict the tactics on unseen new sequences. Our results show that even when annotating only a small number of action sequences that high accuracy output can be achieved.

Further, our evaluation of heuristic rule coverage shows that it is possible to derive the majority of the rules from a small portion of the data, which can further reduce the effort in creating a training set. It has to be noted, however, that this conclusion is based on the assumption that the annotators can always identify a rule when the situation it describes occurs. In practice, annotators may miss some situations and rules derived from the complete set may not be perfect.

In the two previous sections we have outlined and applied our method for efficient identification of tactics from search logs. This addresses our first research objective, namely, *How do we identify search tactics from search logs in a scalable and interpretable manner?* In the following section, we will use the resulting tactic representation to analyse user behaviour, and demonstrate how a focus on search tactics can provide a different lens with which to interpret results from a user study. This will address our second research objective, namely, *How do we use the identified tactics to compare user search activities across different systems?*

6. Using tactics for experimental analyses

Thus far in this paper we have established and validated our method that maps log events (i.e. user and system actions) to search tactics. In this section, we discuss how these identified tactics can be used to analyse user search behaviour across systems (i.e. RQ2).

6.1. Tactic sequences as Markov chains

Transitions between tactics have been widely used as a means of analysing patterns of user search behaviours [26, 55, 56, 11, 12]. Typically, Markov chains are used to model user search tactic sequences, and most frequent sequences of tactics are identified and compared from different user groups [12, 55]. We present an analysis method that models user search tactic sequences as Markov chains from which a measure of entropy can be computed. This entropy-based analysis method allows not only quantitative analysis of user behaviours, but also tests of statistical significance, which is often the key to the utility of a quantitative comparison between systems. With this method we investigate user search behaviour with new perspectives and insights that have not yet been explored in the literature.

The Markov property of tactic sequences

As in related literature [12, 55] we model sequences of user tactics as Markov chains. While construction of such a model is straightforward, one open question remains in relation to finding the appropriate order of the chain. For example, Xie and Joo [55] fit tactic sequences with a 5th-order Markov chain and use it to identify most frequent patterns of length 5, where the choice of length 5 is arbitrary. Alternatively Chen and Cooper [12] use a goodness-of-fit test to determine the order of semi-Markov chains for their analysis.

In our case, the order of the chain we construct for the tactic sequences has a direct influence on how the aforementioned entropy-based measure will be computed. Therefore, before moving on to discuss the entropy-based analysis method, we perform a test for the order of our tactic sequences. Besag and Mondal [4] discussed different procedures to perform a goodness-of-fit test for this purpose. Following their work, we describe the procedure we apply to analyse the tactic sequences.

Recall that the linear-chain CRFs model we used to annotate search tactics from log data is based on the assumption that the transitions between the hidden states (tactics) has a Markov property (i.e. first-order). We have observed that the resulting model can effectively predict the tactic labels for an action sequence. Given this observation, our hypothesis is that a first-order Markov chain is sufficiently good to model the transition patterns of tactics in our data.

Using the same notation as defined in Section 4, a sequence T contains individual tactics $t_1, ..., t_K$, where $t_k \in \mathcal{T}$, i.e. the predefined target tactics which define the state space. For an observed sequence T, our goal is to assess its compatibility with a first-order Markov chain with a transition matrix P whose entries p_{ij} are the probability that user switch from tactic i to tactic j, where p_{ij} are unspecified and $p_{i+} = 1$ ($i+$ denotes the summation over all transitions starting from i). Take a second-order chain as its alternative, the log-likelihood ratio statistic between the two models is

$$u = 2 \sum_{i,j,k} n_{ijk} \log \frac{n_{ijk}/n_{ij+}}{n_{+jk}/n_{+j+}}, \tag{2}$$

where n_{ijk} is the frequency of the observed transition triples (i, j, k) in the sequence. Here, the numerator is the likelihood of fitting the data to a first-order chain, and the denominator is the likelihood of the data given the alternative second-order chain. The higher the value of u, the better the higher-order model fits the data compared to the first-order model. If the first-order chain is correct, then u has a asymptotic distribution that is chi-squared with $s(s-1)^2$ degree of freedom where s is the size of the state space. Given this, an asymptotic p-value can be computed which suggests whether the hypothesis that the simpler formation for T should be rejected. Note that a more complex model can always fit the data better. The logic of this test is that if there is no evidence that there is a conflict between the data and the simpler model, then a more complex model is unlikely to be useful.

We applied the above tests to the tactic sequences obtained from our reference logs. For first-order formation, we found that 100% of the Querium, and 98% of the ViGOR tactic sequences have a p-value

larger than 0.05. That is, for majority of the sequences, the first-order Markov chain fits the data sufficiently well. The results of order testing provide a partial explanation for the promising performance of using a linear chain CRF model for automatic tactic identification.

Entropy as a single measure of tactic transition patterns

Having obtained a Markov model of user search tactics, the models can be used to analyse or compare user behaviours. A typical analysis looks at the most frequent tactic and tactic patterns and observes, often quantitatively, whether and how different user groups employ different tactics or tactic patterns [12, 55, 19]. For instance Chen and Cooper [12] used a Chi-square test to compare the distribution of transitions in search tactic transition matrices.

We propose to compute the entropy rate of the Markov chains derived from the tactic sequence as a single measure of user tactic usage patterns. This method is inspired by the work of Krejtz et al. [36] who analysed eye movement transition patterns between different regions of a stimuli. Using this approach we are able to test not only *whether* there is a difference between user tactic transition patterns, but also *how* they differ. Entropy measures how predictable the users' choice of search tactics are. Further, unlike in previous studies where aggregated transition patterns have been analysed, we construct Markov chains for each search session. This allows us to perform standard statistical tests such as an ANOVA to compare user groups and search systems. Analysis of entropy should not be seen as a replacement for other kinds of analysis, for instance comparing frequencies of tactics used, but as a complement in that it can provide a different perspective on the users' interaction with the search system compared to other methods.

Entropy can be calculated as the following. Given a sequence $T = t_1, ..., t_n$ modelled as a Markov chain with a transition probability p_{ij} and stationary probability π_i where $i, j \in \mathcal{T}$ are states of tactics, the entropy rate of this chain is computed as [14]:

$$H_t = -\sum_{i=1}^{s} \pi_i \sum_{i,j=1}^{s} p_{ij} \log p_{ij}. \tag{3}$$

Further, an entropy for the stationary probability can also be computed:

$$H_s = -\sum_{i=1}^{s} \pi_i \log \pi_i, \tag{4}$$

where p_{ij} and π_i are computed empirically from the observed sequence, i.e. $p_{ij} = n_{ij}/n$ and $\pi_i = n_i/n$ where n_{ij} is the observed frequency of transition from i to j, and n_i is the observed frequency of transitions ending in i.

In the context of search tactics transitions, the highest entropy can be reached when there is an equal probability of transitions between each of the search tactics. The minimal entropy (0) is achieved in a fully deterministic Markov chain where all transitions are either 1 or 0. This means that with a higher entropy there is more randomness in the searchers' transitions between different search tactics. This randomness is an indication that the searchers do not have a clear progression from one search tactic to another. On the other hand, low entropy indicates that the searcher's transition between tactics are highly predictable.

Stationary entropy is calculated from the distribution of search tactics. A higher stationary entropy value indicates that the search tactics are used uniformly, while a lower stationary entropy indicates that some search tactics are preferred over others. Values are expected to vary between 0 and a theoretical maximum depending on the number of states in the Markov model.

It is important to note that although the theoretical maximum of the entropy is dependent on the number of states, entropy is an absolute measurement of how much randomness exists in the transitions. This makes it possible to compare very different systems on the basis of how predictable transitions from search tactics are.

6.2. Comparing systems using entropy of search tactic sequences

To illustrate how two very different search systems, or user groups, can be compared, we used two sets of example hypothesis. The first set (H1) compares two different search systems where each system

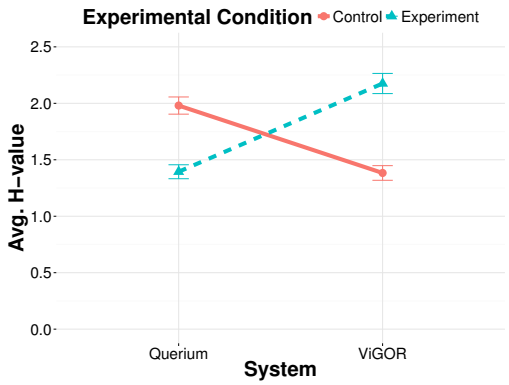


Figure 5: The interaction effect of system and conditions on user search tactic entropy (H).

is run in two conditions. The H1 hypotheses illustrate how our approach can be used to directly compare two very different search systems. The second set (H2) compares two different user groups, experts and novices, using one of the search systems in two different conditions. The H2 hypotheses illustrate how a focus on search tactics provides a different lens to view search logs.

Hypotheses.

H1.1: When introducing a new feature in a search system, users are likely to display a higher level of randomness in choice of search tactics compared to traditional search systems independent of type of search (document or video search).

H1.2: When introducing a new feature in a search system, users are likely to display a less uniform distribution of search tactics since the new features are targeted towards particular tactics.

H2.1: Search experts are likely to be more predictable in their choice of search tactics compared to novices independent of search user interface.

H2.2: Search experts have developed a set of search tactics they prefer over others, while novices use search tactics more uniformly.

H2.3: While working with a search system novices will find a preferred method of transitioning from one search tactic to another. In other words, their search tactics transitions will become more predictable over time.

H2.4: While working with a search systems novices will find preferred search tactics to use. In other words, their distribution of search tactics will become less uniform over time.

For our example analysis we computed the transitional and stationary entropy for each search session in the two data sets. For simplicity, let's assume that the data conformed to a two-way ANOVA, with the two factors: system (Querium and ViGOR) and condition (Experiment and Control), where each subject performed two search tasks[2] using each system.

To test our hypothetical hypothesis H1.1, that new features always increase the randomness of transitioning from one search tactic to another, we started evaluating the transitional entropy using a two-way ANOVA. Contrary to our hypothesis, we did not find a main effects for system, Querium vs. ViGOR, ($F(1, 24)=0.198$) or condition, Control vs. Experiment ($F(1, 24)=0.452$). Users of both system exhibited about the same entropy in their choice of search tactics, and the overall effect of new features did not effect the entropy. However, a significant interaction was found ($F(1, 24)=20.226$, $p<.0001$). Figure 5 illustrates this

[2]To simplify the analysis, the data set was balanced so that each data set included the same number of participants and the same number of tasks per participant. Random subsets of participants from the ViGOR data set and random subset of tasks from the Querium data set were selected.

interaction. When participants used Querium in the control condition they had a significantly higher transitional entropy (v = 0.5862, p<.01) than when using the experimental version of the system. For ViGOR the opposite was true. In the experimental condition the participants had a significantly higher transitional entropy than in the control condition (Experimental version, M=2.2, SD=0.91, Control version, M=1.4, SD=0.66, v=-0.7923, p<.001).

These results indicate that the participants behaved differently when exposed to the experimental condition of the two systems. In Querium the experimental condition appears to streamline the participants' interaction pattern, resulting in lower transitional entropy. In the ViGOR experimental condition included additional functionality with more possible transitions between search tactics. The higher entropy in this condition indicates that the participants incorporated these added functionality into their search strategy at various points. These results show that new features in a search system can have very different effects on the users' interactions on a tactical level with the search system.

Moving on to the second hypothesis comparing systems, H1.2, concerning how new features affect the distribution of search tactics used. This hypothesis is tested using stationary entropy with 2-way repeated measurements ANOVA. Again we see that the participants behaved differently using the two systems. The average stationary entropy was lower for ViGOR (M=1.26, SD=0.117) than for Querium (M=1.42, SD=0.110). This difference was significant (F(1, 29)=22.956, p<.0001) and indicate that when using ViGOR searchers developed preferences for particular search tactics to a higher degree than users of Querium. However, we could not confirm the hypothesis that new features cause a less uniform distribution of tactics, since no effect on condition, Control vs. Experimental, was found (F(1, 27)=0.013). No significant interaction was found (F(1, 27)=3.410). Our results show that the type of system influences to a higher degree the development of preferred tactics than new features.

6.3. Comparing user groups and search sessions

The ViGOR data-set includes data from two different user groups, experts and novices. The experts had extensive experience of video search with similar topics, but were not experts in using ViGOR. In a previous analysis Halvey and Jose [24] found that experts perform better on the search tasks than novices, and that they are more likely to use the grouping functionality in the experimental condition. These differences are all on the action levels. Here, we wanted to understand if the difference in experience transfers to the tactical choices the users make.

To test our hypothesis H2.1, that experts make more predictable transitions between search tactics than novices, we performed a mixed-plot repeated measurements ANOVA with the between factor Expertise (Novice vs. Expert) and the within factor Condition (Experimental vs. Control). We again found that the experimental condition in ViGOR had significantly higher entropy than the control condition (F(1, 14)=12.813, p <.01), also see Table 16. However, we did not find any significant difference between the two expertise levels (F(1, 14)=0.036, ns.), nor any interaction (F(1, 14)= 0.095, ns.). Experts and novices displayed equal entropy in their transitions between search tactics.

Hypothesis, H2.2 concerned how novices and experts developed preferences for particular search tactics. Our analysis of the stationary entropy testing H2.2 did not show any differences between expertise or condition. No significant interaction was found. For ViGOR, the user groups displayed the same level of stationary entropy.

These results are interesting since the previous study [24] found that the added grouping functionality in ViGOR was more frequently used by the experts than by the novices. However, the overall preference for the different search tactics did not differ. Neither did the grouping functions affect the transitional entropy, indicating that randomness in tactics transitions was equal for both user groups. If the novice group, for instance, had a higher entropy, we could have concluded that the higher entropy for this group showed that they failed to incorporate the added functionality into an effective search strategy. A no-difference result, on the other hand, indicates that although the frequency of the different action was different, the randomness in selecting a search strategy from available tactics was not affected. Entropy reflects how predictable transitions between the search tactics are. Hence, the frequency of various actions may not have an impact on the entropy. Entropy analysis, as we have seen in this case, complements frequency analysis of actions and brings a different perspective to the results.

User group	Session	Experiment condition M	SD	Control condition M	SD
Novice	1	2.2	0.84	1.5	0.50
	2	1.8	1.09	1.3	0.49
Expert	1	2.2	0.83	1.4	0.73

Table 16: Average transitional entropy (H) for the two user groups, sessions and condition (experimental or control) for the ViGOR data-set

The novice participants returned for a second study session. The purpose of the second session was to investigate if novices learned and retained from their search experience in their first session, and to see how close their performance in second session would be to the expert's performance.

In our hypothesis H2.3, we wanted to test if novices change their use of search tactics over time. Table 16 shows the transitional entropy for novices in study session 1 and 2 as well as experts' entropy in study session 1. The numbers shows that the novices decreased their entropy in Session 2 compared to Session 1. A repeated measurements analysis comparing the novices' entropy over two study sessions showed that the decrease in entropy between Session 1 and 2 were not quite significant ($F(1, 7)=5.039$, $p=.059$), and a significant difference in entropy between conditions ($F(1, 7)=5.635$, $p<.05$). No interaction was found. These results indicate the novice participants developed a strategy, i.e. a sequence of search tactics, for solving the task in Session 1 and kept this strategy in Session 2. While developing a strategy for tackling a search task, users are likely to explore and switch between search tactics until they have discovered a (for them) optimal transition between search tactics. In the second session, the lower transitional entropy maybe a results of that the novices in Session 2 have already had explored the system's possibilities, resulting in more predictable transitions between search tactics. It is unlikely that the novices switched strategy in the second session to a more optimal search strategy. Any changes from an established pattern of transitions between search tactics are likely to increase the entropy until a new pattern is established and this new pattern has surpassed the old pattern in frequency.

Hypothesis H2.4 dealt with the case of developing preferences for particular search tactics over time. Our repeated measurement analysis comparing the novices' stationary entropy over the two study sessions and conditions showed no significant differences over study sessions or conditions. The novices' distribution of search tactics remains the same over the two sessions, although, the novice's transitions between search tactics became more predictable.

In the previous study [24], the performance results had shown that novices performed closer to experts in Session 2 than in Session 1. However, the entropy results of the search tactics show a different picture. The transitional entropy of Session 1 is basically the same for both user groups. In Session 2, the novice users appear to have found a search strategy that they ran with. Transitional entropy analysis gives a measurement of predictability of search tactics transitions, however performance may not necessarily be correlated with search tactic transitions since it includes the choice of search terms to use as well as the tactical choices made by the users. The experimental condition in ViGOR was a new search user interface for the experts and our results show that they were just as likely as the novices to try out different functionality, however they knew by experience what features and search terms were likely to give better results than others.

6.4. Summary

In this section, we have provided concrete examples that illustrate how the identified search tactics enable cross-system comparison of user search behaviours. By modelling user tactic sequences as a Markov process and using the entropy of Markov chains as a single measure of search tactic transition patterns we were able to compare user search behaviours using standard hypothesis testing statistical procedures.

By comparing the transitional and stationary entropy of the users of the two reference systems in the experimental condition, we see that Querium users employed their search tactics in a more streamlined fashion compared to ViGOR users. Querium's added functionality in the experimental condition led to a lower entropy in transition between search tactics, while the added functionality in ViGOR had the opposite effect. While this type of analysis may seem more relevant if we were comparing systems that are

somewhere related, e.g. as in Wilson et al. [54]'s study, what we demonstrate here is that comparing user behaviour with systems of dramatically different design is possible. Given the availability of usage logs, any systems can be compared in this manner. This is a powerful tool which allows direct comparability between search systems which otherwise would not be possible.

Additionally, it is interesting to observe the contrasting results in comparing user groups in the ViGOR study. The results provide an additional perspective on how tactics and actions deployed by the experts and novices change, and highlight interesting results that could not be found in the original log analysis. Our entropy analysis of search tactics transitions modelled as Markov chains illustrates the possibilities and new perspectives that this kind of analysis can provide and addresses our second research question, namely, *how can the identified tactics be used to compare user search activities across different systems?*

Our proposed entropy-based evaluation method is, of course, only one way to compare user groups or systems. Other methods, such as comparing frequencies of actions and tactics, are well known and utilised. Evaluating entropy is a complement to established methods and can provide, as we have shown, insights into how the use of tactics differ from merely looking at frequencies of particular tactics would.

7. Discussion and conclusion

7.1. Discussion

In this paper we have argued that log analysis, while a powerful tool which offers insights into user behaviour, has a number of drawbacks. The main issues are unclear *interpretability* of findings and a lack of *comparability* of different search systems. To overcome these drawbacks, we believe that what is needed is a method that (i) makes observations from search logs in context; and (ii) enables comparisons across systems of different designs and implementations.

In order to provide a solution, we have proposed a method to abstract away from the low-level logged events and to analyse user activities in terms of a higher-level representation, i.e. search tactics (Section 4). Various aspects of our method have been examined and validated (Section 5). Further, we have demonstrated how this method can be used to provide new insights from log data that would otherwise not be possible (Section 6).

With our research, we have sought to address two research questions;

1. How do we identify search tactics from search logs in a scalable and interpretable manner? Existing approaches to identify tactics from search logs use either hand coding [55, 19] or machine learning. The advantage of hand coded search tactics is that annotators have attempted to understand user search tactics from search logs case by case, which allows relatively accurate interpretations even at a cognitive level, especially by considering the extra contextual information of the users. However, hand coding does not generalise to unseen cases. In addition, exhaustive manual annotation is not practical for analysing large-scale log data. In an effort to create a more scalable approach for identifying search tactics from search logs others have sought to use machine learning. For example, Han et al. [26] proposed an automatic tactic identification method based on HMMs. However, as the result of unsupervised learning, it remains unclear how the resulting hidden states can be interpreted as search tactics in a principled way.

To address the issues with current approaches we have developed ESTI, a new method for discovering tactics in logs of user interactions with complex information retrieval interfaces. This method combines heuristic rules with manual correction enabling efficient human annotation of log data, and a CRF-based statistical method that learns to effectively predict user tactics from search logs. This approach allows us to scale tactic identification, which purely hand coded approaches are unable to do. Also due to the human intervention, we can interpret our results and place meaning on the findings.

It is however clear that no approach is perfect. Well-designed heuristic rules rely on annotators with good skills in abstracting and mapping action segments to search tactics, as well as having a log sample that covers all possibilities. Machine learning may generalise to unseen cases, whereas heuristic rules cannot. However, once discovered heuristic rules can be more accurate. Thus, one of our main motivations for exploring a range of options was to examine the pros and cons of each approach. This extensive investigation means that any adopters of our method can make an informed decision about which approach they can adopt with success to their context. We believe that this serves as a strength of the method and research presented in this paper.

In an effort to evaluate and validate our approach analysis of two logs from very different search systems was conducted. Empirical results on logs of two exploratory search systems have shown the effectiveness of the proposed approaches. In particular, we see that a relatively small amount of training data (20%) can be used to provide reasonable annotation performance, with very little benefit to adding additional training data. This finding coupled with heuristic rule-based annotation and the use of manual correction with our online annotation tool means that a small amount of data can be quickly annotated, which in turn can then be used to provide an accurate classifier which can accurately identify search tactics, thus demonstrating its scalability.

How well tactics can be mapped from actions parsed from a log file depends on the design of the system. Both Querium and ViGOR included many different user manipulations which are captured by the log files. More actions captured in the logs express the context in which a single action can be interpreted. ESTI relies on using context of actions for mapping action sequences to tactics. Although a system with fewer actions expressed in logs may be more difficult to map to tactics, we believe that context assists the interpretation. In this work, we tested ESTI with two very different search systems which gives us an indication that ESTI is generalisable to other systems as well.

In order to identify tactics from search logs, we restricted the definition of tactics to be tactics supported by the system and that the tactics should be possible to map to an action sequence. The assumption that tactics can be inferred from actions is a necessary limitation in our work. Analysing large log files would not be possible without this simplification. Although this limitation affects conclusions that can be drawn from an analysis that excludes the cognitive dimension of the search tactics, we believe that ESTI provides a useful framework for large-scale log analysis where, for instance, long-term user behaviour is analysed or when comparing two different search system of real-life deployments.

2. How do we use the identified tactics to compare user search activities across different systems? While user actions are highly dependent on the specific design and implementations of a particular system, search tactics can be defined independent of a particular search system. By mapping user actions to tactics, system specific representations of user behaviour are translated into a shared vocabulary, which allows us to compare user behaviour across different types of systems. Moreover, while actions are not always intentional due to system design or logging deployment (cf. L3/L4 cases), tactics are meaningful groups of user actions and the transition between tactics conveys information about how users move their search forward. Consequently the dependencies observed between actions are not always meaningful. Whereas at a tactical level these cases are marginalised and we can focus on the intentional activities of the users.

More concretely, beyond simply proposing a new methodology we have demonstrated how possible analyses can be performed with the identified search tactics, providing insights and perspectives into user search behaviours that have not been investigated or indeed possible previously. One of the main benefits of our approach is demonstrated by allowing us to compare Querium and ViGOR directly in terms of tactics although they have very different interfaces and implementations. For instance, we have observed that users have more diverse preferences in employing search tactics with Querium compared to ViGOR; the additional functionality provided by Querium in the experimental condition allows users to develop more predictable tactic transition patterns, while the opposite was observed for the additional functionality of ViGOR. This type of comparison is very powerful and has the potential to allow a comparison of many types of interfaces at a higher tactical level. Normally when need arises to compare two interfaces a new user study would have to be designed and conducted, which is costly for time and resources. With our method logs from different user studies or systems can be used to directly compare search systems.

Beyond the comparability offered by our method, it can also provide a new lens with which to view user behaviour encoded in log files. For example, when analysing ViGOR we saw that results from a tactical level investigation yield a different explanation of user behaviour in comparison to examining interaction at the action level. In the original experiment [24], we were unable to de-tangle expertise on a collection level and a system level. Insights gained through new analysis on a tactical level allow us to decouple these, which would not be possible otherwise.

7.2. Limitations

We have discussed the benefits of the new method that we have outlined in the paper, however this work also has some limitations that we should acknowledge. First, our proposed method assumes the existence

of a set of search tactics that captures all user interaction with a search system. Whilst the choice of search tactics does allow flexibility, there are multiple definitions of search tactics and multiple sets of search tactics [2, 42, 3]. The decision of which tactics to use is a choice to be made when applying our method. In our case it was informed by selecting a set of tactics that we felt matched the interactions captured in our logs. That being said we still had to add additional tactics into our set of search tactics to capture all possible interactions. For sets of search tactics like Bates [1] there are stages that involve primarily cognitive processes with few observable manifestations. These search tactics are much more difficult to capture by automatic methods such as in user logs, but with advances in EEG and eye tracking technologies it may be possible in the future. We believe that although it is possible to use different sets of search tactics, it may be possible to map between some of the states of these different representations. Further we strongly believe that a tactical level representation allows for greater comparability between different search systems than is currently afforded by current log level analysis.

Second, whilst we have demonstrated that the ESTI method can be applied with a small amount of manual effort, our method still requires some manual intervention. This involves some discussion of the log files and how they should be mapped to different tactics, and what the combination of tactics mean. However, we believe that this is necessary for standard log analysis. Whilst simply counting certain types of interactions and comparing them statistically can be straight forward, fully understanding the meaning of results of log analysis requires further and deeper thought beyond counting actions. Indeed this is part of the motivation for the development of this method. Through our analysis of different approaches we have demonstrated how this manual intervention can be reduced and in some cases automated.

7.3. Future work

A number of directions are left to be explored in the future. For instance, our human annotation process can be assisted further with unsupervised segmentation algorithms such as sequence clustering, or combined with the HMM approach proposed by Han et al. [26]. Further, while our study has focused on information seeking, we believe that the technique generalises to other kinds of interactive systems. Good candidates for this analysis should support cognitive tasks in which people have a large number of choices of what to do, and relative freedom about the order in which the low-level actions are performed. Examples are complex editors such as Photoshop, MS Word or even Emacs. This new methodology provides a new tool for evaluating user interfaces which provides the ability to make comparisons and provide insights that were not previously possible.

7.4. Conclusion

In this paper we have put forward a powerful new scalable and interpretable method which allows us to elicit search tactics from log files with very little additional effort beyond normal log analysis. The empirical result of our proposed methodology, combined with our illustrative examples suggest that identifying tactics in logs of user activities is both feasible and useful. This method (ESTI) extends the range of current evaluation techniques available to IR researchers and provides a new lens to help gain deeper insights about the rich information contained in log files.

Acknowledgements

This research was partially supported by the Dutch Technology Foundation (STW) under project nr. 13675 awarded to the first author, and the Scottish Informatics and Computer Science Alliance (SICSA) under the PECE scheme awarded to the third author. We thank the management of FX Palo Alto Laboratory for supporting this research and hosting Jiyin He and Martin Halvey's extended visit and Tony Dunnigan for creating the graphical abstract for this paper.

Our friend and colleague Gene Golovchinsky unexpectedly passed away in 2013. He was instrumental for this work by bringing all of us together during a few months at FXPAL, where the foundation of this research project was laid. Gene was generous with intense discussions, encouraging smiles and hands-on coding and labelling during this memorably time.

8. References

[1] M. J. Bates. Information search tactics. *J. Am. Soc. Inf. Sci.*, 30(4):205–214, 1979.

[2] M. J. Bates. Where should the person stop and the information search interface start? *Information Processing & Management*, 26(5):575–591, 1990.

[3] N. J. Belkin, C. Cool, A. Stein, and U. Thiel. Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert systems with applications*, 9(3): 379–395, 1995.

[4] J. Besag and D. Mondal. Exact goodness-of-fit tests for markov chains. *Biometrics*, 69(2):488–496, 2013.

[5] P. Boldi, F. Bonchi, C. Castillo, and S. Vigna. Query reformulation mining: models, patterns, and applications. *Information Retrieval*, 14(3):257–289, 2010.

[6] A. Broder. A taxonomy of web search. *ACM SIGIR Forum*, 36(2):3–10, 2002.

[7] P. Bruza and S. Dennis. Query reformulation on the internet: empirical data and the hyperindex search engine. In *RIAO'97*, pages 488–499, 1997.

[8] I. Campbell. Interactive evaluation of the ostensive model using a new test collection of images with multiple relevance assessments. *Information Retrieval*, 2(1):89–114, 2000.

[9] L. D. Catledge and J. E. Pitkow. Characterizing browsing strategies in the world-wide web. *Computer Networks and ISDN systems*, 27(6):1065–1073, 1995.

[10] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *WWW'09*, pages 1–10, 2009.

[11] J. Chapman. A state transition analysis of online information-seeking behavior. *JASIST*, 32(5):325–333, 1981.

[12] H. Chen and M. Cooper. Stochastic modeling of usage patterns in a web-based information system. *J. Am. Soc. Inf. Sci. (JASIST)*, 53(7):536–548, 2002.

[13] A. Chuklin, P. Serdyukov, and M. de Rijke. Click model-based information retrieval metrics. In *SIGIR '13*, 2013.

[14] G. Ciuperca and V. Girardin. On the estimation of the entropy rate of finite markov chains. In *Proceedings of the International Symposium on Applied Stochastic Models and Data Analysis*, 2005.

[15] R. P. Costa and N. Seco. Hyponymy extraction and web search behavior analysis based on query reformulation. In *IBERAMIA'08*, 2008.

[16] D. Downey, S. Dumais, and E. Horvitz. Models of searching and browsing: languages, studies, and applications. In *IJCAI'07*, pages 2740–2747, 2007.

[17] G. E. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *SIGIR'08*, pages 331–338, 2008.

[18] G. Golovchinsky, A. Diriye, and T. Dunnigan. The future is in the past: designing for exploratory search. In *Proceedings of the 4th Information Interaction in Context Symposium*, pages 52–61. ACM, 2012.

[19] A. A. Goodrum, M. M. Bejune, and A. C. Siochi. A state transition analysis of image search patterns on the web. In *Image and Video Retrieval*, pages 281–290, 2003.

[20] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y.-M. Wang, and C. Faloutsos. Click chain model in web search. In *WWW'09*, pages 11–20, 2009.

[21] F. Guo, C. Liu, and Y. Wang. Efficient multiple-click models in web search. In *WSDM '09*, 2009.

[22] Q. Guo and E. Agichtein. Beyond dwell time: estimating document relevance from cursor movements and other post-click searcher behavior. In *Proceedings of the 21st international conference on World Wide Web*, pages 569–578. ACM, 2012.

[23] M. Hagen, J. Gomoll, A. Beyer, and B. Stein. From search session detection to search mission detection. In *OAIR'13*, pages 85–92, 2013.

[24] M. Halvey and J. M. Jose. The role of expertise in aiding video search. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, page 8. ACM, 2009.

[25] M. Halvey, D. Vallet, D. Hannah, and J. M. Jose. Vigor: a grouping oriented interface for search and retrieval in video libraries. In *JCDL'09*, pages 87–96. ACM, 2009.

[26] S. Han, Z. Yue, and D. He. Automatic detection of search tactic in individual information seeking: A hidden markov model approach. *arXiv preprint arXiv:1304.1924*, 2013.

[27] D. He, A. Göker, and D. J. Harper. Combining evidence for automatic web session identification. *Information Processing and Management*, 38(5):727–742, 2002.

[28] J. He, M. Bron, and A. P. de Vries. Characterizing stages of a multi-session complex search task through direct and indirect query modifications. In *SIGIR'13*, pages 897–900. ACM, 2013.

[29] V. Hollink, J. He, and A. P. de Vries. Explaining query modifications - an alternative interpretation of term addition and removal. In *ECIR'12*, pages 1–12, 2012.

[30] J. Huang, R. W. White, and S. Dumais. No clicks, no problem: Using cursor movements to understand and improve search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1225–1234, 2011.

[31] B. J. Jansen, D. L. Booth, and A. Spink. Patterns of query reformulation during web searching. *J. Am. Soc. Inf. Sci. (JASIST)*, 60(7):1358–1371, 2009.

[32] T. Joachims. Optimizing search engines using clickthrough data. In *SIGKDD*, pages 133–142. ACM, 2002.

[33] S. S. Keerthi and S. Sundararajan. Crf versus svm-struct for sequence labeling. Technical report, Technical report, Yahoo Research, 2007.

[34] Y. Kim, A. Hassan, R. W. White, and I. Zitouni. Modeling dwell time to predict click-level satisfaction. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 193–202, 2014.

[35] R. Kohavi, R. Longbotham, D. Sommerfield, and R. M. Henne. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18(1):140–181, 2009.

[36] K. Krejtz, T. Szmidt, A. T. Duchowski, and I. Krejtz. Entropy-based statistical analysis of eye movement transitions. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 159–166. ACM, 2014.

[37] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML'01*, 2001.

[38] D. Lagun, M. Ageev, Q. Guo, and E. Agichtein. Discovering common motifs in cursor movement data for improving web search. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 183–192. ACM, 2014.

[39] C. Liu, R. W. White, and S. Dumais. Understanding web browsing behaviors through weibull analysis of dwell time. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 379–386, 2010.

[40] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei. Identifying task-based sessions in search engine query logs. In *WSDM'11*, pages 277–286. ACM, 2011.

[41] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei. Modeling and predicting the task-by-task behavior of search engine users. In *OAIR'13*, pages 77–84, 2013.

[42] G. Marchionini. *Information seeking in electronic environments*. Cambridge University Press, 1995.

[43] M. Nakazato, L. Manola, and T. S. Huang. Imagegrouper: a group-oriented user interface for content-based image retrieval and digital image arrangement. *Journal of Visual Languages & Computing*, 14 (4):363–386, 2003.

[44] N. Nguyen and Y. Guo. Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 681–688, New York, NY, USA, 2007. ACM. doi: 10.1145/1273496.1273582.

[45] P. Qvarfordt, G. Golovchinsky, T. Dunnigan, and E. Agapie. Looking ahead: query preview in exploratory search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 243–252. ACM, 2013.

[46] K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372, 1998.

[47] D. E. Rose and D. Levinson. Understanding user goals in web search. In *Proc. WWW*, pages 13–19, 2004.

[48] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, MIR '06, pages 321–330, New York, NY, USA, 2006. ACM. ISBN 1-59593-495-2. doi: 10.1145/1178677.1178722. URL http://doi.acm.org/10.1145/1178677.1178722.

[49] M. Spiliopoulou, B. Mobasher, B. Berendt, and M. Nakagawa. A framework for the evaluation of session reconstruction heuristics in web-usage analysis. *Informs journal on computing*, 15(2):171–190, 2003.

[50] J. Urban and J. M. Jose. Ego: a personalized multimedia management and retrieval tool. *International Journal of Intelligent Systems*, 21(7):725–745, 2006.

[51] S. Verberne, M. van der Heijden, M. Hinne, M. Sappelli, S. Koldijk, E. Hoenkamp, and W. Kraaij. Reliability and validity of query intent assessments. *Journal of the American Society for Information Science and Technology*, 64(11):2224–2237, 2013.

[52] R. W. White and S. M. Drucker. Investigating behavioral variability in web search. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 21–30, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. doi: 10.1145/1242572.1242576. URL http://doi.acm.org/10.1145/1242572.1242576.

[53] R. W. White, M. Bilenko, and S. Cucerzan. Studying the use of popular destinations to enhance web search interaction. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 159–166, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. doi: 10.1145/1277741.1277771. URL http://doi.acm.org/10.1145/1277741.1277771.

[54] M. L. Wilson, m. schraefel, and R. W. White. Evaluating advanced search interfaces using established information-seeking models. *J. Am. Soc. Inf. Sci. Technol.*, 60(7):1407–1422, 2009.

[55] I. Xie and S. Joo. Transitions in search tactics during the web-based search process. *J. Am. Soc. Inf. Sci. Technol.*, 61(11):2188–2205, 2010.

[56] Z. Yue, S. Han, and D. He. A comparison of action transitions in individual and collaborative exploratory web search. In *Information Retrieval Technology*, volume 7675, pages 52–63, 2012.

[57] Z. Yue, S. Han, and D. He. Modeling search processes using hidden states in collaborative exploratory web search. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 820–830. ACM, 2014.

[58] Y. Zhang, W. Chen, D. Wang, and Q. Yang. User-click modeling for understanding and predicting search-behavior. In *KDD '11*, pages 1388–1396, 2011.