Distribution Agreement

Signature:

Dmitry Lagun   Date

Modeling User Attention and Interaction on the Web

By

Dmitry Lagun
Doctor of Philosophy

Eugene Agichtein
Advisor

Stuart Zola
Committee Member

Phillip Wolff
Committee Member

Alexander Smola, Carnegie Mellon University
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Modeling User Attention and Interaction on the Web

By

Dmitry Lagun
M.S., Emory University, 2013

Advisor: Eugene Agichtein

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
Computer Science and Informatics
2015

Abstract

Modeling User Attention and Interaction on the Web
By Dmitry Lagun

Analysis of user attention and Web page examination behavior, collected with specialized eye tracking equipment, has offered numerous insights about how users examine content online. Unfortunately, eye tracking technology is currently available for relatively small scale user studies, due to its high costs and the effort associated with participant recruitment. This thesis develops several alternatives to eye tracking for studying user attention and behavior. We start by introducing ViewSer - a method based on idea of restricted focus viewing, that allows measuring attention for thousands of participants. Then, we develop a probabilistic model that infers most likely position of user's gaze on the screen from user interactions and Web page content. Our model outperforms current state of the art for gaze position prediction that only uses behavioral signals or information about Web page visual content. In addition to the methodological contributions, this thesis develops several important applications in Web search and medical domain. First, we describe a scalable approach for extracting frequent mouse cursor movement patterns from large scale cursor data. These patterns could be used to improve quality of search result relevance estimation and search result ranking. Second, we show that attention measured with cursor and viewport position could be used to improve automatic Web page summarization. Lastly, we demonstrate an important medical application of restricted focus viewing, for automated diagnostic of memory impairment that could be administered remotely over the Internet anywhere in the world. Together, the techniques developed and evaluated in this thesis substantially advance the state of the art of user attention modeling and enable novel practical applications.

Modeling User Attention and Interaction on the Web

By

Dmitry Lagun
M.S., Emory University, 2013

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
Computer Science and Informatics
2015

Acknowledgments

First and foremost, I must thank Professor Eugene Agichtein of Emory University, who was my academic advisor throughout PhD years, and provided me with tremendous support and guidance on research presented in this thesis. I started my work with Eugene from a project on automatic detection of cognitive impairment using the eye movement data, which at that time, I was quite unfamiliar with. Eugene guided my learnings on the topic, and showed me how the insights from this work can be applied to problems of attention modeling during Web search, and eventually benefit information retrieval research. He patiently listened to craziest of my ideas and provided some critical feedback that made several pieces of this thesis materialize into successful research. He highly encouraged me to attend conferences and pursue internships at top industry research labs, where I was able to connect with world known researchers. Eugene is an amazing mentor and I feel very fortunate to be able to work with him and benefit from his advice.

I would also like to thank my thesis committee members: Dr. Stuart Zola, Professor Alexander Smola, and Professor Phillip Wolf. Their thoughtful comments and suggestions allowed me to strengthen the technical content and see the broader context and potential implications of the attention modeling work. In addition to being my committee member, Alex was extremely kind in providing helpful feedback on my internship work, pointing out surprising connections between my research and important machine learning problems, and suggesting practical ways to approach them.

A significant portion of this thesis is a result of a fruitful collaboration with my colleagues. Work on cursor motif mining and behavior biased automatic document summarization described in Chapters 4 and 6 were possible through a collaboration with Mikhail Ageev and Qi Guo. Throughout two Mikhail's visits to Emory, he and I had many discussions, which thanks to Mikhail's critical thinking, experience and great passion, materialized in a work that was recognized with several prestigious awards. Qi and I, being office mates, had numerous enjoyable discussions on the lab's whiteboard, which sharpened my intuition and motivated some of the ideas in this thesis. Work on web based visual paired comparison task and automatic detection of cognitive impairment resulted from collaboration with esteemed researchers at the Yerkes National Primate Research Center. I was honored to receive feedback on this work from Dr. Stuart Zola, Dr. Beth Buffalo and Cecelia Manzanares.

I gained a lot of experience through several internship opportunities at industry labs. First, I would like thank Daniel Billsus and Dominic Hughes for hosting me during summer of 2011 at eBay Advertising. Their thoughtful comments and timely feedback on the intern project allowed me to iterate faster and be able to work on multiple projects during the internship. Second, I would like to thank Peter Bailey and Ryen White of Microsoft for hosting me during the summer of 2012.

With their thorough guidance I was able to broaden my research horizon. I am thankful to other members of the team - Georg Buscher and Avneesh Sud. I truly enjoyed lunch time discussions with Georg Buscher that motivated some of the work presented in this thesis. Lastly, I owe a big thanks to Vidhya Navalpakkam of Google for hosting me during the summer of 2013 and being a fantastic mentor. During the internship, Vidhya kindly allowed me to explore several research projects and spared no effort in making each of the projects truly enjoyable journey.

While successful PhD is mostly determined by the academic achievements, one can not underestimate challenges facing international students coming from a different country to study in the US. I am extremely grateful to the administrative staff at the Math and Computer Science Department, including Terry Ingram and Dr. Vaidy Sunderam, for their constant support and help throughout my time at Emory.

I would like to thank all my collaborators and co-authors in various projects: Noah Adler, Mikhail Ageev, Amr Ahmed, Peter Bailey, Daniel Billsus, Beth Buffalo, Georg Buscher, Mark Cramer, Qi Guo, Chih-Hung Hsieh, Dominic Hughes, Hajian Jin, JinYoung Kim, Ravi Kumar, Mounia Lalmas, Qiaoling Liu, Cecelia Manzanares, Vidhya Navalpakkam, Denis Savenkov, Ilya Shats, Avneesh Sud, Jaime Teevan, Sergei Vassilvitskii, Dale Webster, Josh Weinstock, Ryen White, Shuai Yuan, and Stuart Zola.

Special thanks go to the members of Emory Information Intelligent Lab: Noah Adler, Mikhail Ageev, Ablimit Aji, David Fink, Haojian Jin, Julia Kiseleva, Alexander Kotov, Qiaoling Liu, Denis Savenkov, Ilya Shats, JongHo Shin, Yu Wang, Josh Weinstock, and Nikita Zhiltsov.

Work presented in this thesis was funded by National Institute of Health under Grant R01-EB014266, National Science Foundation with Grant IIS-1018321, Yahoo! Key Scientific Challenges Award and Yahoo! Faculty Research and Engagement Program. I am truly thankful for their generous support and making this work possible.

Finally, I'm incredibly grateful to my wife Dina for reminding me to eat and sleep, for love and encouragement when things weren't going well, for always being ready to drop everything because of me, for patience and understanding all the late nights and weekends that went into this thesis, and during which I was not able to spend time with the family.

To my parents for all their love, support
and encouragement to pursue graduate studies.
To my wife Dina for sharing this journey with me.
To my children, Yakov and Milka, for inspiring me.

Contents

List of Figures

List of Tables

Chapter 1

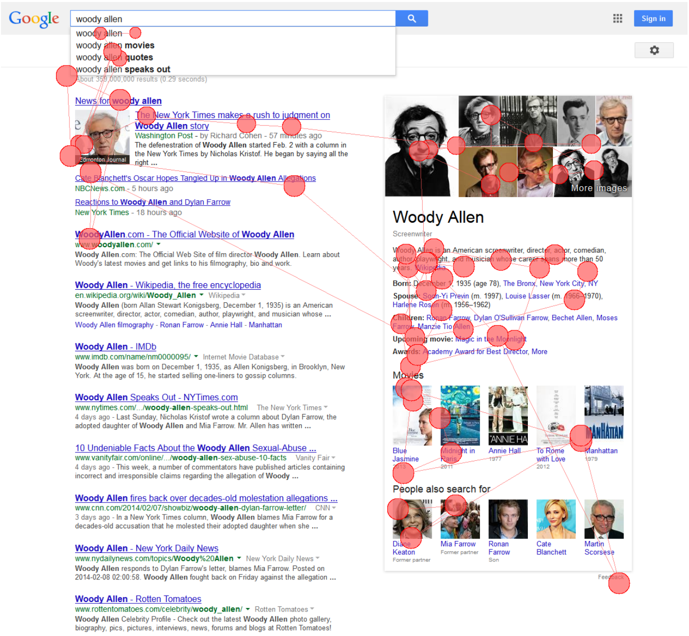
Introduction

1.1 Background and Motivation

Studies of online user behavior emerged as a powerful tool for improving constantly evolving and increasingly complex online systems, such as Web search engines. For this reason, every user action, however small is, being scrupulously recorded in order to provide information about typical system usages, and, in some cases, allow to infer user's preferences and satisfaction with the online service. By analyzing user actions recorded in the most natural settings, engineers and designers are able to gain comprehensive understanding of system's users at scale not possible in the laboratory settings.

Laboratory studies, despite their limitations, allow analyzing user behavior at significantly greater detail, e.g., with eye tracking equipment, that is unavailable in large scale log data. Eye tracking technology emerged as a powerful methodology for studying online user behavior and provided numerous insights into how users interact with Web pages, what page content they pay attention to and what information on the page is being ignored. It is difficult to overestimate the benefits from discovery of what is known in Web search as *position bias* [45] (or F-shaped examination pattern) in viewing of Web search results, as it enabled new research direction in modeling and interpreting user click behavior [26, 55, 74], which in turn, led to significant improvement of search quality [4, 48, 55]. However, despite the clear utility of eye tracking method, its limitations – relatively high cost of equipment and significant burden on participant recruitment and data collection, restrict potential pool of applications.

On the other hand, mouse cursor tracking has been proposed as an affordable and scalable alternative to the small scale eye tracking laboratory studies. Cursor tracking allows to track mouse cursor movements of Web site visitors which can help to identify key elements of a Web site that users interact with frequently and are of greater importance to the users. Recently, some promising results in user attention modeling were demonstrated using mouse cursor movements - they were used to approximate human attention and predict user's gaze position on the computer screen during Web search. This task still remains remarkably difficult due to the vast uncertainty in eye movement behavior on a Web page. However, if accomplished successfully, this could potentially impact various applications ranging from improvements of Web site usability[88, 94] to high throughput behavioral



woody allen

woody allen
woody allen **movies**
woody allen **quotes**
woody allen **speaks out**

About 359,000,000 results (0.29 seconds)

News for **woody allen**

Edmonton Journal

The New York Times makes a rush to judgment on **Woody Allen** story
Washington Post - by Richard Cohen - 57 minutes ago
The defenestration of **Woody Allen** started Feb. 2 with a column in the New York Times by Nicholas Kristof. He began by saying all the right ...

Cate Blanchett's Oscar Hopes Tangled Up in **Woody Allen** Allegations
NBCNews.com - 5 hours ago

Reactions to **Woody Allen** and Dylan Farrow
New York Times - 18 hours ago

WoodyAllen.com - The Official Website of **Woody Allen**
www.**woodyallen**.com/
Woody Allen.com: The Official Web Site of film director **Woody Allen**. Learn about Woody's latest movies and get links to his filmography, bio and work.

Woody Allen - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Woody_Allen** Wikipedia
Woody Allen (born Allan Stewart Konigsberg, December 1, 1935) is an American screenwriter, director, actor, comedian, author, playwright, and musician whose ...
Woody Allen filmography - Ronan Farrow - Annie Hall - Manhattan

Woody Allen - IMDb
www.imdb.com/name/nm0000095/ Internet Movie Database
Woody Allen was born on December 1, 1935, as Allen Konigsberg, in Brooklyn, New York. At the age of 15, he started selling one-liners to gossip columns.

Woody Allen Speaks Out - NYTimes.com
www.nytimes.com/.../**woody-allen**-speaks-out.html The New York Times
4 days ago - Last Sunday, Nicholas Kristof wrote a column about Dylan Farrow, the adopted daughter of **Woody Allen** and Mia Farrow. Mr. Allen has written ...

10 Undeniable Facts About the **Woody Allen** Sexual-Abuse ...
www.vanityfair.com/online/.../**woody-allen**-sex-abuse-10-facts Vanity Fair
4 days ago - This week, a number of commentators have published articles containing incorrect and irresponsible claims regarding the allegation of **Woody** ...

Woody Allen fires back over decades-old molestation allegations ...
www.cnn.com/2014/02/07/showbiz/**woody-allen**-dylan-farrow-letter/ CNN
3 days ago - In a New York Times column, **Woody Allen** blames Mia Farrow for a decades-old accusation that he molested their adopted daughter when she ...

Woody Allen - New York Daily News
www.nydailynews.com/topics/**Woody**%20**Allen** New York Daily News
Woody Allen responds to Dylan Farrow's letter, blames Mia Farrow. Posted on 2014-02-08 02:00:58. **Woody Allen** fought back on Friday against the allegation ...

Woody Allen - Rotten Tomatoes
www.rottentomatoes.com/celebrity/**woody_allen**/ Rotten Tomatoes
Woody Allen Celebrity Profile - Check out the latest **Woody Allen** photo gallery, biography, pics, pictures, interviews, news, forums and blogs at Rotten Tomatoes!

More images

Woody Allen
Screenwriter
Woody Allen is an American screenwriter, director, actor, comedian, author, playwright, and musician whose career spans more than 50 years. Wikipedia
Born: December 1, 1935 (age 78), The Bronx, New York City, NY
Spouse: Soon-Yi Previn (m. 1997), Louise Lasser (m. 1966–1970), Harlene Rosen (m. 1956–1962)
Children: Ronan Farrow, Dylan O'Sullivan Farrow, Bechet Allen, Moses Farrow, Manzie Tio Allen
Upcoming movie: Magic in the Moonlight
Awards: Academy Award for Best Director, More

Movies
Blue Jasmine 2013
Midnight in Paris 2011
Annie Hall 1977
To Rome with Love 2012
Manhattan 1979

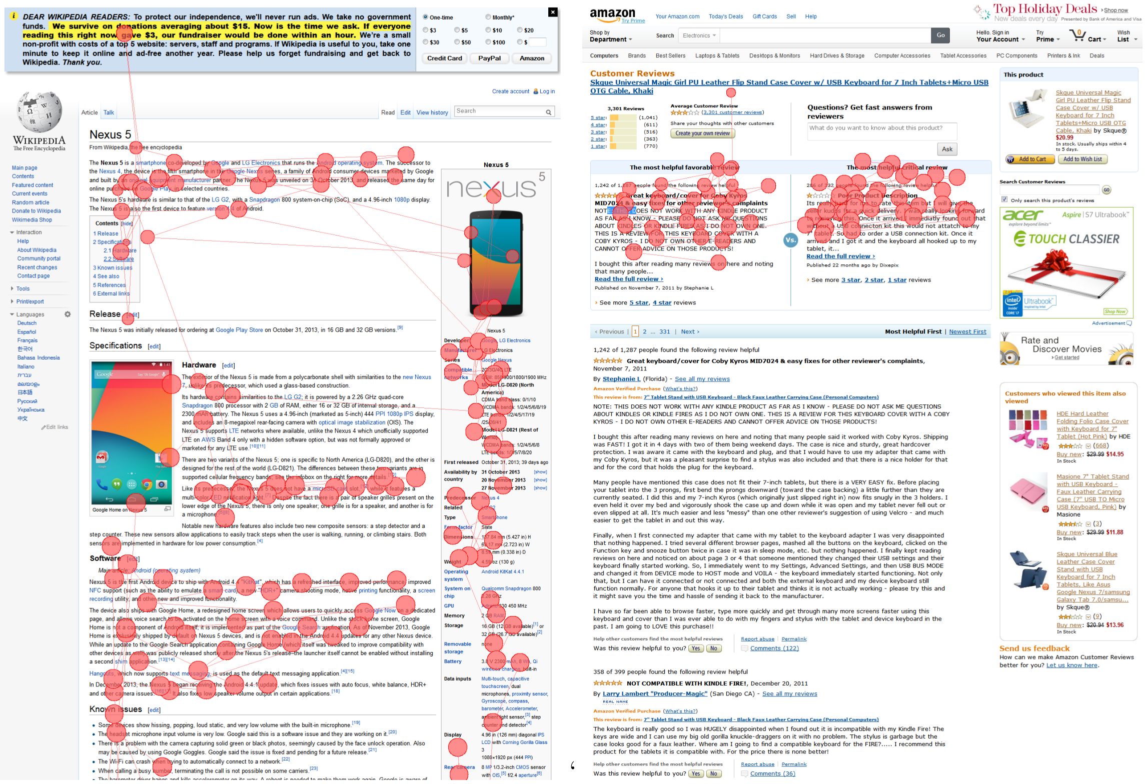
People also search for
Diane Keaton Former partner
Mia Farrow Former partner
Ronan Farrow Son
Cate Blanchett
Martin Scorsese

Feedback

Figure 1.1: Example of search result page examination. Eye movement fixations are shown with red circles. A perfect attention tracking approach should be able to detect user's interest in information panel on the right side of the Web page.

testing that can detect early onset of cognitive decline for elderly population or detect attention disorders.

Nonetheless, the accuracy of attention tracking, using current mouse cursor tracking approaches, remains unsatisfying for some important applications. Relatively low accuracy of gaze prediction models, in part, stems from inadequate fitness metrics used for model training. Consider, for instance, eye movement trace recorded on a search result page shown on Figure 1.1. Due to lack of Web page content information most of the current state of the art gaze prediction models would predict gaze position near top search results effectively ignoring highly attractive information panel shown on the right side of the page. The "ideal" system should account for the eye catching content within the information panel on the right side of the page. Figure 1.2 shows additional examples of why incorporating page content information into the attention model is important. Figure 1.2a shows example eye movements recorded during browsing of the Wikipedia Web page and Figure 1.2b shows eye movement trace for the Amazon Web page. On both Figures eye movements are attracted by the task relevant content, suggesting that incorporating Web page content information should improve the prediction accuracy. Moreover, since the extent of eye-cursor coordination may vary depending on the user's task (i.e. eye and cursor are well coordinated when user performs an action, such as click, and much less coordinated during reading, when cursor is inactive), Web page

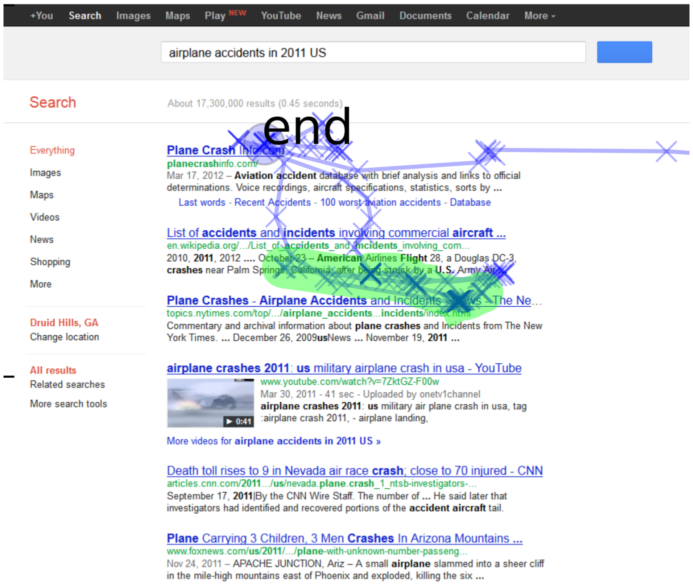


(a) (b)

Figure 1.2: Example eye movement traces recorded during browsing of Wikipedia (a) and Amazon (b) Web pages. The figure shows that eye movement patterns are determined by position of the task relevant content.

content information becomes particularly useful to identify most probable position of user's gaze. Practically speaking, such "ideal" system should incorporate evidence from both sources: content attractiveness and behavioral signals (e.g. cursor movements and clicks) revealing user attention during actionable interactions.

While not necessary accurate from attention measurement viewpoint, cursor tracking is certainly of great utility to the Web search. Only until recently researchers and practitioners have been using search result click data as a sole source of information about user behavior on the search result page. Although click information is quite important for the search engines, it only reveals a final decision - which result user decided to click, and conceals intermediate considerations taken by a user. For example, consider Figure 1.3 that shows a search result page for a search query "airplane accidents in 2011 US" overlaid with traces of user activity. User's cursor movements are shown with blue crosses (x). The clicked search results is annotated with "end" marker, indicating that first result was clicked. As we can see user examines first three results and decides to click on first search result. Interestingly enough, third search result seems quite relevant as it has several keyword matches with the search query (more than first search result), however, user decides to click on the first result for whatever reason. Nonetheless, we can find that third result attracted user attention, as indicated by prolonged cursor movement, hovering near the third result. Thus, if cursor movements are recorded by a search engine, information about what other search results attracted user attention (but received no clicks) can be further utilized by the search engine to improve its ranking algorithm.



airplane accidents in 2011 US

About 17,300,000 results (0.45 seconds)

Plane Crash Info.com
planecrashinfo.com/
Mar 17, 2012 – **Aviation accident** database with brief analysis and links to official determinations. Voice recordings, aircraft specifications, statistics, sorts by ...
Last words - Recent Accidents - 100 worst aviation accidents - Database

List of **accidents** and **incidents** involving commercial **aircraft** ...
en.wikipedia.org/.../List_of_**accidents**_and_**incidents**_involving_com...
2010, **2011**, 2012 .... October 23 – **American** Airlines **Flight** 28, a Douglas DC-3, **crashes** near Palm Springs, California, after being struck by a **U.S.** Army Air ...

Plane Crashes - **Airplane Accidents** and Incidents - News - The Ne...
topics.nytimes.com/top/.../**airplane_accidents**...**incidents**/index.html
Commentary and archival information about **plane crashes** and Incidents from The New York Times. ... December 26, 2009**us**News ... November 19, **2011** ...

airplane crashes 2011: **us** military airplane crash in usa - YouTube
www.youtube.com/watch?v=7ZktGZ-F00w
Mar 30, 2011 - 41 sec - Uploaded by onetv1channel
airplane crashes 2011: **us** military air plane crash in usa, tag :airplane crash 2011, - airplane landing,

More videos for **airplane accidents in 2011 US** »

Death toll rises to 9 in Nevada air race **crash**; close to 70 injured - CNN
articles.cnn.com/**2011**.../**us**/nevada.**plane**.**crash**_1_ntsb-investigators-...
September 17, **2011**|By the CNN Wire Staff. The number of ... He said later that investigators had identified and recovered portions of the **accident aircraft** tail.

Plane Carrying 3 Children, 3 Men **Crashes** In Arizona Mountains ...
www.foxnews.com/**us**/**2011**/.../**plane**-with-unknown-number-passeng...
Nov 24, 2011 – APACHE JUNCTION, Ariz – A small **airplane** slammed into a sheer cliff in the mile-high mountains east of Phoenix and exploded, killing the six ...

Figure 1.3: Example search result page with mouse cursor movements shown with blur crosses (x). Click position is marked with “end” marker, indicating that first search result was clicked.

In addition to usability studies and Web search, human attention tracking has become a fertile ground for research, thanks to recent advancements in the field of neuroscience. There is mounting evidence that eye movement collected in visual paired comparison task can predict future onset of Alzheimer Disease in up to three years in advance [129, 92, 33]. Even more interesting, abnormalities in attention allocation were shown to indicate attention deficit hyperactivity disorder, fetal alcohol spectrum disorder and Parkinsons disease [118]. Deployment of such methods for screening of large populations has potential to drastically improve the public health and enable early diagnosis of attention and memory disorders. While such techniques currently require expensive eye tracking equipment with all its limitations, we demonstrate that in some cases it is possible to adapt the behavioral test and produce an equivalent Web based version of the test that relies on mouse cursor tracking and can be administered remotely to large population [5, 91].

In this thesis, we build upon these ideas and develop scalable and more accurate models of attention tracking as well as its applications in Web search and behavioral testing. First, in Chapter 3 we describe a technique, called ViewSer, that enables remote studies of Web page examination at scale not feasible for the traditional laboratory eye tracking studies. In Chapter 4 we describe our approach to understanding vast amounts of mouse cursor data by finding frequent mouse cursor movement patterns (called *motifs*). To fully implement our vision in part of developing scalable attention tracking techniques we borrow insights from computer vision and computational neuroscience. Namely, we seek to model allocation of human attention on the Web pages using information about what is currently displayed user's computer screen and user's action, including mouse cursor movements, scrolling and clicking activity. To this end, in Chapter 5 we propose a model for human attention that naturally integrates information about user interactions (e.g. mouse cursor movements) and a Web page content user is currently viewing. In Chapter 6 we demonstrate practical utility of mouse cursor tracking in tasks of document relevance prediction and attention biased document summarization. Finally, in Chapter 7 we apply idea of restricted focus viewing to image stimuli viewing, and demonstrate how it can be used for high throughput behavioral testing of memory impairment.

1.2 Contributions

This thesis makes both methodological and empirical contributions:

- Methods
  - **A scalable approach for remote studies of Web page examination** [88]. Using the idea of restricted focus viewing we develop a framework for conducing remote studies of Web page examination allowing to rapidly evaluate usability of user interfaces and collect Web page examination data. We show that data collected with our approach closely approximates the data collected with unrestricted viewing during an in-situ laboratory study. In the application part of the thesis we demonstrate utility of this technique for Web search and medical applications.
  - **An effective approach for mining representative mouse cursor movement patterns** [87]. We develop a viable way to automatically extract frequent patterns of mouse cursor movements that could be used to understand the common behavior of the Web site visitors. In addition, the extracted mouse cursor patterns (also called *motifs*) can be used to relate user behavior to subjective or objective variables of the Web page or the user. This approach eliminates laborious manual feature engineering used by previous techniques and, at the same time, builds task-free compact representation of mouse cursor behavior that could be used for variety of tasks beyond relevance prediction.

  - **A joint model of visual attention on Web pages integrating Web page content and user interactions.** We develop a model of user attention during information seeking and Web page browsing tasks. We build upon the prior work on user attention tracking from user's mouse cursor movements. Our approach incorporates two sources of evidence: user mouse cursor movements and Web page information to improve accuracy of gaze position inference.

- Applications

  - **A scalable approach to behavioral testing of memory function** [91]. Based on idea of restricted focus viewing [88] we develop Web based version of visual paired comparison task, that is able to detect memory decline, enabling more effective medical intervention and treatment. This work paves the way for wide availability of attention tracking for behavioral testing. It could also open new methods for rapid screening of large population for attention and memory disorders.
  - **Web search ranking with mouse cursor movement data** [87]. We demonstrate how mouse cursor data can be used to better estimate document relevance in web search settings and subsequently benefit web search ranking system.
  - **Attention biased document summarization** [3, 2]. We demonstrate how mouse cursor data could be utilized in automatic document summarization. We build a statistical model that is able to predict text fragments that are of most interest to a user from the mouse cursor movement recordings and position the text fragment appeared on the user's screen. Our results indicate that attention based model is able to significantly improve quality of automatic summaries [3] and benefit passage retrieval in context of automatic question answering [2].

To summarize, in the first part of the thesis we develop methods for accurate attention measurement and analysis of cursor movement data collected from large user populations. In the second part, we describe various applications of cursor movement data for improving Web search ranking, automatic generation of attention biased document summaries and online tools for attention based screening of memory impairment.

Chapter 2

Related Work

Four lines of prior research relate to our work. First line of research focuses on controlled studies investigating various dimensions of user interface efficiency. These studies employ *eye tracking methodology* mainly as exploratory and evaluative tool. This includes studies of web page examination strategies, user and content factors and their effect on user behavior. Second line of research relates to models of *information processing* and patterns of information consumption by humans. This includes models of visual saliency and oculomotor control during viewing of still images and video content; behavioral patterns in reading and abnormalities of eye movements under certain psychological or cognitive impairments. Third line of related work focuses on *user attention tracking* from interaction with a user interface, such as search results pages. It includes studies of coordination between user's gaze position and mouse cursor position on the screen, as well models directly predicting user attention from mouse cursor movement features. Lastly, the fourth line of prior research relates to ideas of using user interactions such as eye gaze movements, mouse cursor clicks, cursor movements, internet browser scrolling and other observable user actions to infer some latent state related to a particular user or viewed object (e.g., a web page). Examples of such approaches in web search include models of document relevance, user's intent inference and prediction of search satisfaction; examples in medical domain include models predicting patient's cognitive status based on the eye gaze or mouse cursor movements recording in a behavioral test.

2.1 Eye Movement Data and User Interfaces

Eye tracking technology has become an extremely valuable tool for analysis of user's visual search strategies in various layouts and arrangements of presented content.

In web search domain, one of the early studies was conduced by Goldberg et al. [44] where they explored eye movement transition patterns on the search result page. In the controlled user study they experimented with a two-column result presentation layout. Their analysis showed that horizontal direction dominated in the visual search, i.e. users switched between columns frequently, as opposed to reading information within the single column and only then switching to the second. Later, eye tracking technology has been extensively used in studies of web search result examination behavior. Granka et al.[45] studied how users browse search result list and select links. Their results suggested that users spend most of the time inspecting the first and the second result before their initial click. Based on the insights gained from eye tracking data Joachims et al. [75] formulated most common examination strategies and demonstrated one way such strategies can be used to infer relevance of result ranking as perceived by user from the result click information. Pan et al. [107] found that gender and web page complexity have a significant impact on some eye tracking metrics, such average fixation duration and degree of scan-path variation for different subjects on the same page.

Lorigo et al. [99] used eye tracking to study eye gaze trajectories on a search result page in more detail. They found that only 25% of users examine search results in the order they are presented by a search engine. A similar study was conducted by Guan and Cutrell [47], where they studied effect of target result position on searcher's examination behavior.

More recently, Buscher et al.[24] investigated effect of advertisement quality on searcher's receptiveness (or blindness) to ads. They found that when ad quality varied randomly, users paid very little attention to the advertisement. Navalpakkam et al.[105] conducted controlled study where they varied presence and relevance of a rich informational panel placed to the right from the organic search results. They found that relevant knowledge panel attracts more user attention, than irrelevant knowledge panel modules.

Aula et al.[8] reported two types of search result examination patterns – *economic* and *exhaustive*. Economic users inspect results sequentially from the top to bottom and click on the first relevant link they notice. In contrast, exhaustive searchers thoroughly examine search result page and consider every result before choosing a result they want to click. Dumais et al. [36] extended this work by clustering users based on their examination behavior of whole search page. In addition to user examination pattern on organic search results they considered user attention on advertisement modules.

2.2 Eye Movements and Information Processing

Significant amount of research focused on studying relationship between characteristics of the eye movements and underlying cognitive processes happening inside of the human brain. Pioneering study of Just and Carpenter [77] presented the "eye-mind link" hypothesis and demonstrated that to some degree eye movements can indicate the thought currently the on "top of the stack" of cognitive processes. Thus, by examining eye movement recordings we could trace person's attention on a visual display at any given point of time. Measuring other aspects of eye movements, such as fixations (moments when the eyes are relatively stationary, taking in or encoding information), can also reveal the amount of processing being applied to objects at the point-of-regard. Interestingly, interpretation of fixations may depend on the task. That is, in a web page browsing task, higher fixation frequency on a particular area might be indicative of greater interest in the target or it might indicate that target is complex in some way and is difficult to encode [77]. However, these interpretations may be reversed in a search task - a higher number of fixations may indicate a greater uncertainty in recognizing the search target. Another common type of eye movement, saccades are often defined as rapid eye movement jumps (up to 900deg/s). It was shown that no information processing is performed during the saccades[38]. However, saccades metrics such as number of saccades or saccade amplitude are very informative about directionality of visual searching and the effort spent on visual searching as opposed to information encoding[108].

In reading research eye tracking has been extensively used for studying cognitive processes during reading tasks [112]. Early research on eye movement control during reading task dates back to the work of McConkie et al.[100] where they investigated fixation landing position within word boundaries. Remarkable progress in modeling eye movement control during reading task was done in past ten years. Reichle [113, 114] presented the E-Z Reader model that is able to simulate eye movement behavior in terms of word skipping and fixation duration in the reading task.

In computational neuroscience and vision research eye tracking has been used to study low level mechanisms of human attention. The basis of many attention models dates back to Treisman and Gelades [123] “Feature Integration Theory,” where they showed which visual features are important and how they are combined to direct human attention. Koch and Ullman [86] then proposed a feed-forward model to combine these features and introduced the concept of a saliency map which is a topographic map that represents conspicuousness of scene locations. Almost a decade later Itti et al. [68] proposed a computational model of human attention in the image viewing task. Itti’s model mimics early visual processes thought to be performed by our brains. It computes various feature maps of the input stimuli such as color, intensity, orientation and motion activation maps. Then it combines all of these maps to produce a single conspicuousness (saliency) map highlighting the regions of visual stimuli where person is likely to look at. His model was shown to produce remarkably good results and predict eye movement fixations during first several seconds of viewing the image. Harel et al. [60] introduced graph based visual saliency model, where model takes into account self resemblance of the scene. Recently, Borji and Itti [17] provided a comprehensive benchmark for attention models comparing performance of more than 65 models on the same data.

2.3 Modeling Attention from Cursor Interactions

Modeling searcher attention and interest has wide-ranging applications in web search ranking, evaluation, and interface design. Traditionally, most of the experimental work on user attention relied on infrared eye tracking which allows tracking eye movement on computer screen at great detail, however, recently mouse cursor tracking emerged as more accessible and scalable proxy for user’s attention (e.g., [53] and [62]).

Tracking user attention from mouse cursor movements has long history in the human computer interaction community. Chen et al. [27] was one of the first to study coordination patterns between mouse cursor and gaze. They classified mouse cursor movement in to five classes: Stay Nowhere, Go

Nowhere, Stay the Same Region and Go to New Region. The found that distance between mouse cursor and gaze position was as low as 90 px for the region of the page where user attended. Chen and Liu [28] explored cursor gaze coordination in context of online education websites. They proposed a methodology to detect page elements that attracted user attention. Their approach relied on partitioning the screen area (for a given resolution) into a number of "zones" and grouping all mouse movements that occur within a particular zone. They used Longest Common Subsequence algorithm to find repetitive pattern in cursor movements and detect page areas that interested user.

Rodden et al. [115] studied cursor gaze alignment on the Web search pages and discovered the coordination between a user's eye movements and mouse movements when scanning a web search results page. They identified three patterns of active mouse usage: *following the eye vertically with the mouse*, *following the eye horizontally with the mouse*, and *using the mouse to mark a promising result.* Guo and Agichtein [53] extended this work to predict eye-mouse coordination (i.e., whether the mouse cursor is in close proximity to eye gaze at any given point in time) by modeling mouse movements. This work was further extended by Huang et al. [62] to directly predict the gaze position from mouse cursor movement. Huang et al. hypothesized that extent of cursor-eye alignment is different for different types of cursor movements. They showed that cursor and eye are best aligned when user is performing click action, and have largest average distance in periods of cursor inactivity. More recently, Ageev et al. [3] demonstrated that cursor data collected on search result landing pages can be used to extract text fragments that attract user attention and subsequently improves quality of search result summaries (snippets). These efforts solidify the evidence that a user's attention in web search can be approximated by using mouse cursor movement, scrolling, and other interaction data.

2.4 User Interactions in Web Search

In addition to studying user attention, mouse cursor data have been used for more practical tasks. Goecks and Shavlik [43], modeled user actions such as mouse activity to infer user's interest in web pages. Shapira [119] studied several mouse cursor-based implicit interest indicators and found that the ratio of mouse movement to reading time was a good indicator of the explicit page rating. Guo and Agichtein [49] modeled mouse cursor movement and other interactions for inferring general search intent such as *navigational* vs. *informational*, as well as other intent categories, allowing for more accurate future ad clickthrough prediction [51]. Huang et al. [65] found that hovering over a search result provides indication of relevance in addition to result clickthrough. Huang et al. [62] also developed models to predict result clickthrough by incorporating mouse hovering and scrolling information. White and Buscher [126] proposed a method that uses text selections as implicit feedback. Most recently, Guo and Agichtein [55] proposed a *Post Click Behavior (PCB)* model to estimate the "intrinsic" relevance by engineering a wide array of features to capture post-click behavior such as mouse cursor movements and scrolling, resulting in substantial improvements in estimating personalized search relevance and re-ranking search results.

Chapter 3

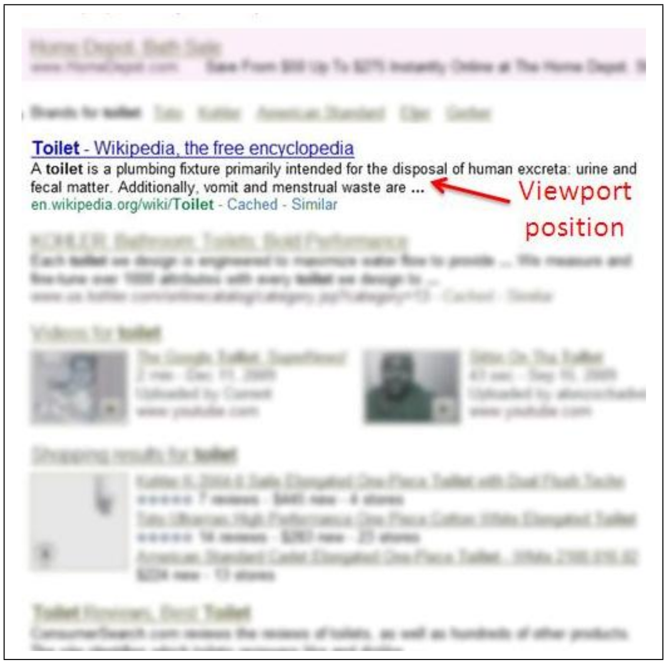
Restricted Focus Viewing

3.1 Background and Motivation

Web search engines serve billions of searches a day, providing information for a diverse range of information needs. Understanding and analyzing how users interact with search has emerged as an important area of research. In particular, eye tracking has proven to be an invaluable technology for studying search behavior, providing important insights into search interface design. Yet, despite these advantages, eye tracking studies remain relatively small scale, as they require in-lab participation and supervision, and thus are "too expensive" for day-to-day search evaluation.

This work proposes a new methodology for *performing large-scale behavioral studies of web search*, while maintaining many of the benefits of the controlled in-lab eye tracking studies of search. For this, we present a specially designed search engine result interface, which we call ViewSer (for Viewport Examination of Web Search Results). ViewSer aims to induce result examination behavior similar to unrestricted viewing, yet allowing us to track precisely the viewed portion of the search result page. For this, ViewSer blurs most of the search result page, except for the search result currently examined (pointed to) by the cursor, which creates a clear "viewport", illustrated in Figure 3.1. This viewport follows the cursor position, allowing a subject to examine the search results, while the viewport position is tracked.

The kinds of web search evaluation for which ViewSer is designed, focus on evaluating search results individually. Examples of such tasks are: measuring the rates of result examination and estimating snippet attractiveness – valuable for accurate clickthrough interpretation [125] and for learning to rank from click data; and evaluating snippets (result abstracts), e.g., by using the proportion of views to clicks on a result [128], as we demonstrate in this work. Indeed, in Section 6.1 of this thesis we explore multiple practical applications of ViewSer. As a first task, we show that ViewSer can serve as an effective method to measure and estimate snippet attractiveness - indicating that a snippet tends to "attract" clicks. This in turn can help better interpret clickthrough data for tasks such as learning to rank. Another crucial task is evaluating quality of search result snippets. To this end, we explore an application of ViewSer to detect bad (misleading) snippets which can serve as a valuable feedback to snippet generation algorithms.



Toilet - Wikipedia, the free encyclopedia
A toilet is a plumbing fixture primarily intended for the disposal of human excreta: urine and fecal matter. Additionally, vomit and menstrual waste are ...
en.wikipedia.org/wiki/Toilet - Cached - Similar

Figure 3.1: An example of the ViewSer interface displaying a blurred search engine result page (SERP) for the query “toilet”, with the viewport revealing the first result.

Recently, crowdsourcing methodology has emerged as a viable way to cheaply obtain human input for a wide range of tasks, including document relevance assessments. One of the most popular web sites providing a marketplace for hiring internet workers is Amazon Mechanical Turk (AMT). Previous efforts studied various aspects of document relevance rating crowd-sourced via AMT, including task completion time, worker’s responsiveness, locality and ratings quality in terms of accuracy and inter-rater agreement [6, 93, 29, 85]. Kelly et. al. reports an important study of searcher behavior for in-lab and remote participants [81]. In contrast to these studies, our focus is searcher behavior - specifically, search result examination. Somewhat related to our approach, [61] describes crowdsourcing user studies of graphical perception conducted via AMT. However, we are not aware of any published user study of web search examination behavior conducted for hundreds of users in crowdsourcing framework. Our work is inspired by the emergence of the large-scale, passive logging and analysis of search behavior as an alternative to in-lab studies: the log data has been used for search evaluation [6], for improving search engine ranking [4, 72] among other tasks. However, such log-based studies are a blunt instrument - they are more appropriate for overall search performance evaluation, whereas our proposed methodology enables precise tracking and characterization of searcher behavior, at the level of detail previously only possible with eye tracking studies of search. To enable this vision, our implementation of ViewSer builds upon the previous work on restricted focus viewing (RFV) described in references [16, 70, 12], where the

Query	***Description***
`mitchell college`	Find information about Mitchell College in New London, CT, such as a prospective student might find useful.
`cheap internet`	I'm looking for cheap (i.e. low-cost) internet service.
`espn sports`	I'm looking for various sports scores and information from the ESPN Sports site.
`euclid`	Find information on the Greek mathematician Euclid.

Table 3.1: Example queries and descriptions provided to the subjects.

authors explored the effect of restricted viewing in usability studies of user interfaces. However, our work substantially differs from prior applications of this idea, as our work is, as far as we know, the first to apply this idea to web search. Our approach is also more general, scalable, and efficient compared to previous work: our implementation is based on the Scalable Vector Graphics (SVG) technology natively supported by the Firefox browser, which in turn enables ViewSer to render and blur rich XHTML content such as text formatted with cascade style-sheets, images and videos, while [16] describes an application to image examination. As we show, ViewSer has many potential applications to web search. In particular, estimating the "attractiveness" (with respect to clickthrough) of search result summaries, or snippets can improve click interpretation, which is in turn helpful for more accurate ranking models. Previously, Clarke et al. [31] found statistically significant changes in clickthrough patterns due to caption features. The more recent work of [128] confirmed some of the findings of [31] in a different experiment using the concept of "fair pairs". Both of these approaches used methodology based on changes in clickthrough, whereas our work directly measures the searcher examination of the captions, and subsequent behavior. The bulk of this work was published in [**?**].

3.2 ViewSer Implementation for Web Pages

The ViewSer system is outlined in Figure 3.3. First, ViewSer retrieves and pre-processes the search engine result pages (SERPs), by inserting code into each SERP to modify the appearance and to enable tracking of the user interaction events. These SERP pages, and the landing pages of the results (the actual documents) are cached in a database for the subsequent studies, enabling fully reproducible and repeatable experiments. A participant opens one of the pre-processed SERPs, which causes her browser to blur all but one results on the SERP. As the participant moves the viewport around to view the rest of the results, the precise position of the viewport and other searcher interactions are sent to the server and logged for future analysis. The rest of this section describes these steps in more detail.

```
<svg:svg>
<svg:filter id=”make−blur”>
<svg:feGaussianBlur stdDeviation=”2.5”/>
<svg:feColorMatrix values=”
 0.3333  0.3333  0.3333  0      0
 0.3333  0.3333  0.3333  0      0
 0.3333  0.3333  0.3333  0      0
 0       0       0       1      0”/>
```

Figure 3.2: A fragment of the Support Vector Graphics (SVG) code used by ViewSer to blur web page elements.

SERP pre-processing: to emulate the “viewport” of the ViewSer interface, we automatically modified the SERPs by inserting the JavaScript/Scalable Vector Graphics (SVG) code, which is directly supported by the Firefox browser, without requiring any additional plugins or other downloads. The code leaves clear one result region at a time (identified using the HTML DOM tags), and blurs the rest of the SERP. Specifically, the SVG specification is used to describe the blurring effect and incorporate it into regular cascade style sheet class, which can be added to any HTML DOM element in a web page.

More precisely, the fragment of the code in Figure 3.2, *make-blur*, defines a Gaussian filter for blurring a search result, and can be referenced in a style specification of any HTML element to blur the element’s content accordingly. This specific filter is a Gaussian filter with the σ parameter set to 2.5. The second operation performed by make-blur is gray-scaling the element appearance. Conveniently, each search result on a SERP is described within list element $\langle LI \rangle$. Therefore we modified the style of $\langle LI \rangle$ elements on the SERP, thus blurring all of the results.

ViewSer Front-End

Initially, all out-of-focus elements are blurred and discolored to grayscale in order to imitate peripheral vision. Then, when a viewport moves over an element (e.g., a search result), the element’s style can be changed back to the original appearance by detecting the onMouseOut event. The SVG-based implementation makes ViewSer scalable for crowdsourcing, as it does not require any additional installation, and responsive by exploiting the optimized native browser support, while allowing precise tracking of the viewing of any HTML element (such as the result position on SERP). These are significant advantages over previously proposed implementations using browser plugins or Java applications [16, 70]. A limitation of ViewSer is that only the complete HTML DOM elements can be revealed, not allowing for partial or gradual occlusion.

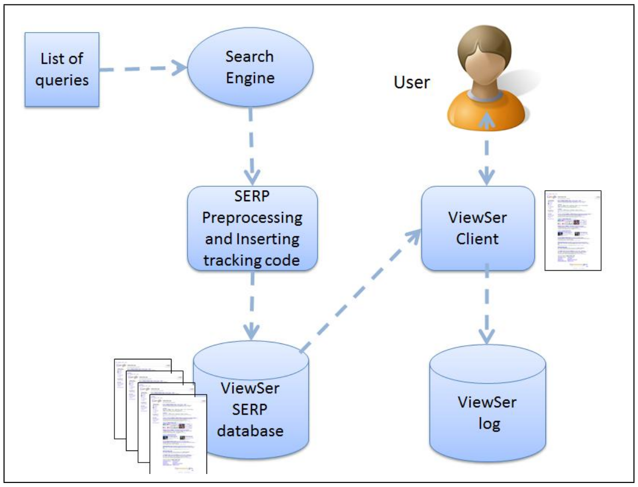


Figure 3.3: The ViewSer architecture for large-scale search result examination studies.

Logging the Searcher Behavior

To track the viewport movement and other user interaction events, we injected additional JavaScript code into the SERP shown on the client's machine. This code logged window events such as clicking, scrolling, mouse movements, and events indicating cursor hover over a search result lasting 200ms or longer (corresponding to the typical duration of a eye movement fixation [112]). These events were buffered and periodically sent to the server via asynchronous HTTP requests for the subsequent analysis.

3.3 Validating Viewser for Search Result Pages

To validate the ViewSer method we performed two main user studies: first, to collect "ground truth" eye tracking data; and second, to collect examination data using our ViewSer interface to compare to the eye-tracking behavior.

Search Tasks and Study Procedure

We used 25 benchmark search tasks from the WEB Track of the TREC 2009[1] competition. The goal for each task (the task description) was provided to the participants. For example, the goal of the query "toilet" was stated as: "Find information on buying, installing, and repairing toilets". For each task, the query keywords were submitted to the Google search engine, and the Search Engine Result Pages (SERPs), as well as all the result documents linked from each SERP, were cached. The original SERP layout was not modified (as shown on Figure 3.1), recreating a realistic

[1] `http://trec.nist.gov/data/web09.html`

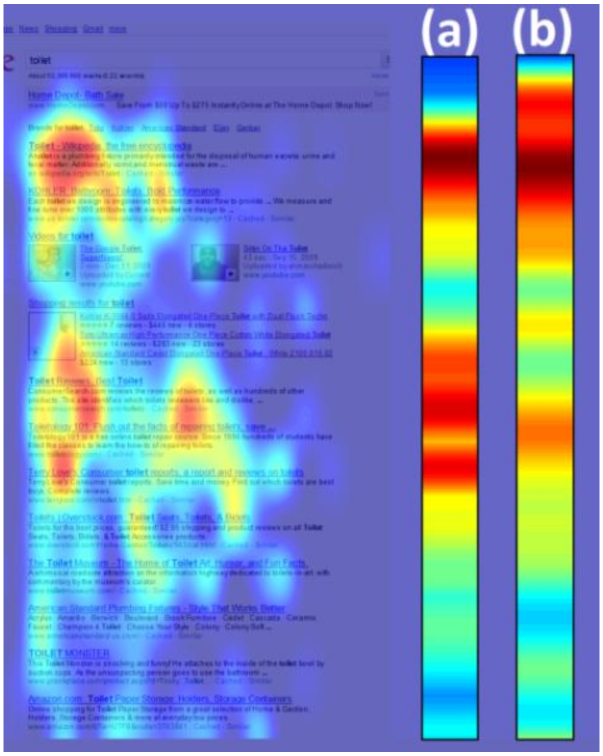


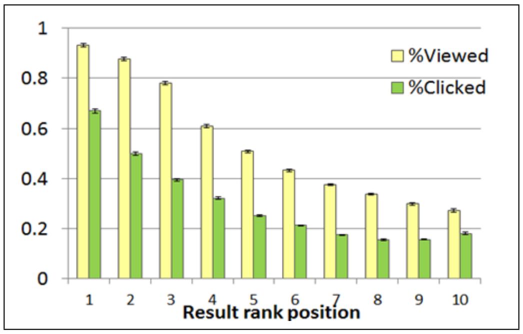
Figure 3.4: An example attention heatmap showing the relative viewing time over a SERP for the query “toilet” (Eye-tracking group), and the corresponding colorbar, showing the heatmap density projected onto the vertical axis (a). Overlaid as (b) is the colorbar for the viewing time for the same SERP but for the ViewSer group. This figure illustrates the similar distribution of attention between eye-tracking (a) and ViewSer (b).

search experience for the participants. The participants started with a provided SERP for each query, and were instructed to find the needed information with least effort that is, to click only on results that appear relevant. After a subject clicked on a result to examine the document and went back to the SERP, she was asked to rate the document relevance. To be considered a valid response, we required that participants attempt all search tasks, and click and rate at least one result for each task.

Eye-tracking group: for this “ground truth” group, ten participants (6 female, 4 male, ages 23.0±1.5, mostly graduate and undergraduate students and fluent English speakers) were recruited. The eye tracking was performed using a Tobii x60 eye tracker paired with a 17" LCD monitor set to 1280x1024 resolution. The subject’s gaze position was sampled at 60 Hz with accuracy of 0.5 degrees. For the two remote studies, participants were recruited through the Amazon Mechanical Turk website, using the standard mechanism of listing our study as an available Human Intelligence Task (HIT).

ViewSer group: the workers were required to use the popular Firefox web browser. They were instructed to view the search results using the ViewSer interface as described above. 203 MTurk workers attempted the remote study. As a first step, the data obtained from MTurk subjects were automatically filtered to discard careless or automated (robot) workers. While the instructions required providing relevance judgements, some workers did not provide relevance judgements, and/or spent less than 1 minute on the whole HIT of 25 queries (presumably, to obtain the payment and move on). After these cases were automatically filtered out, we had valid data from 106 workers (48%).

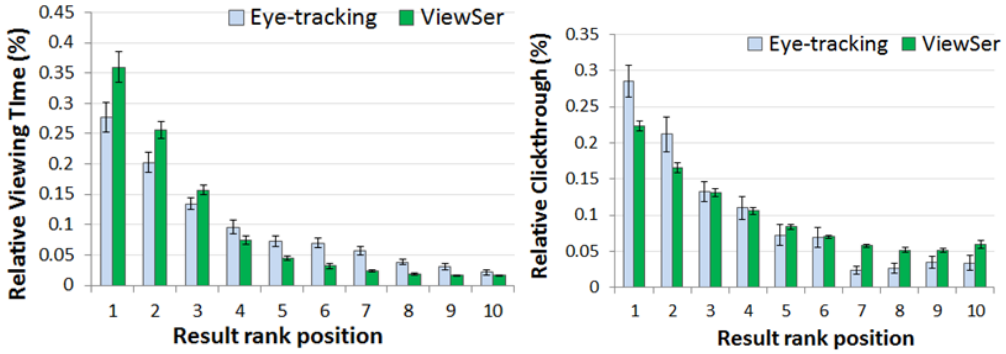
Unconstrained viewing group: to serve as "control" subjects, we recruited additional 25 MTurk workers. The task was identical to the ViewSer group, except that we removed blurring, allowing for unconstrained viewing of the SERP.



.

Figure 3.5: Viewing and clickthrough rates for each rank, aggregated for all queries and participants (ViewSer group).

Our goal was to investigate whether ViewSer indeed induces similar viewing and clickthrough behavior remotely in MTurk subjects, as in the unconstrained viewing setting for the in-lab eye tracking subjects and remote participants. Before presenting quantitative results we analyze examination behavior qualitatively using attention heatmap shown in Figure 3.4. The Figure shows an example heatmap of the relative time spent viewing the SERP for the query "toilet", aggregated for all subjects in Eye-tracking group. The first vertical colorbar (a) projects the relative viewing time onto the vertical axis of the SERP for the Eye-tracking group, and the second colorbar (b) for the ViewSer group, showing a noticeable similarity between the most intensely scrutinized search results. Overall, the ViewSer group required 1 minute and 37 seconds on average (SD=70 seconds) for each search task, compared to 55 seconds on average (SD=20 seconds) for the Eye-tracking group. While the subjects in the ViewSer group took more time for each task, this is to be expected due to more time required to move a mouse pointer. Interestingly, the resulting search behavior patterns of the two groups are remarkably similar otherwise.
]

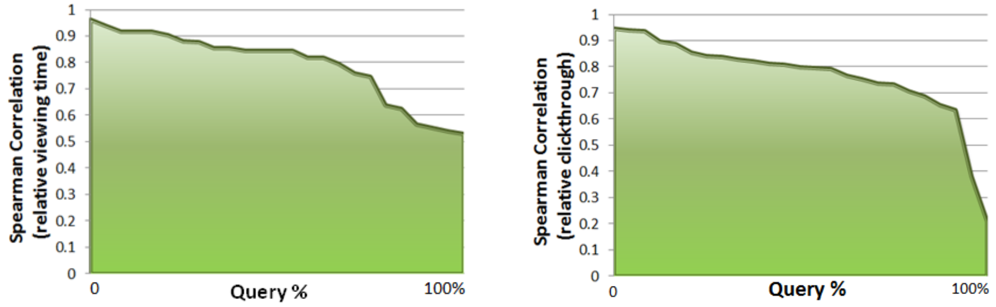


(a) (b)

Figure 3.6: Viewing time (a) and clickthrough rate (b) comparison for the ViewSer and the Eye-Tracking groups, aggregated across all queries and subjects.

ViewSer SERP Examination and Clickthrough

Figure 3.5 reports the viewing and clickthrough rates for each result rank for the ViewSer group. Each data point indicates the fraction of the result views at each rank position, and the corresponding fraction of the clicks landing on that position, for all searches and participants. The first 3 results were viewed for 93%, 87% , and 78% of the tune, respectively, dropping to 27% for the 10th result. The clickthrough values are correlated with the viewing, with the exception of the results in the last (10th) position, which is slightly more likely to be clicked than the 9th result. These viewing and clickthrough patterns correspond well to the previous studies of unconstrained search result examination behavior [99, 65].



(a) (b)

Figure 3.7: Spearman correlation of the relative viewing time (a) and clickthrough rates (b) for individual queries, for the Eye-tracking and ViewSer groups. The queries are sorted by the correlation coefficient. (mean viewing correlation: 0.79, mean clickthrough correlation: 0.76).

Comparative Analysis of ViewSer vs. Eye-Tracking

Figures 3.6a and 3.6b report the relative viewing times and clickthrough rates for the Eye-Tracking and ViewSer groups, respectively. The values in Figure 3.6a were computed for each subject and query (that is, the viewing time for a particular abstract by a subject was divided by the total viewing time of the corresponding SERP) for an individual query, and then averaged across all queries. The relative viewing time is important, as the longer a searcher's gaze stays on a particular area, more information is processed, and therefore this area receives more attention. Thus, comparing the relative amount of time (attention) spent on examining results through the ViewSer interface, vs. that of the Eye-Tracking group, quantifies the similarity of the viewing behavior of the two groups. Interestingly, ViewSer participants spent more time viewing first results is probably because of higher speed of eye movements compared to hand (mouse) movement, which may increase the likelihood of skipping over the results when viewing the page unconstrained (eye tracking group) compared to requiring to move the mouse to reveal the next result (ViewSer group).

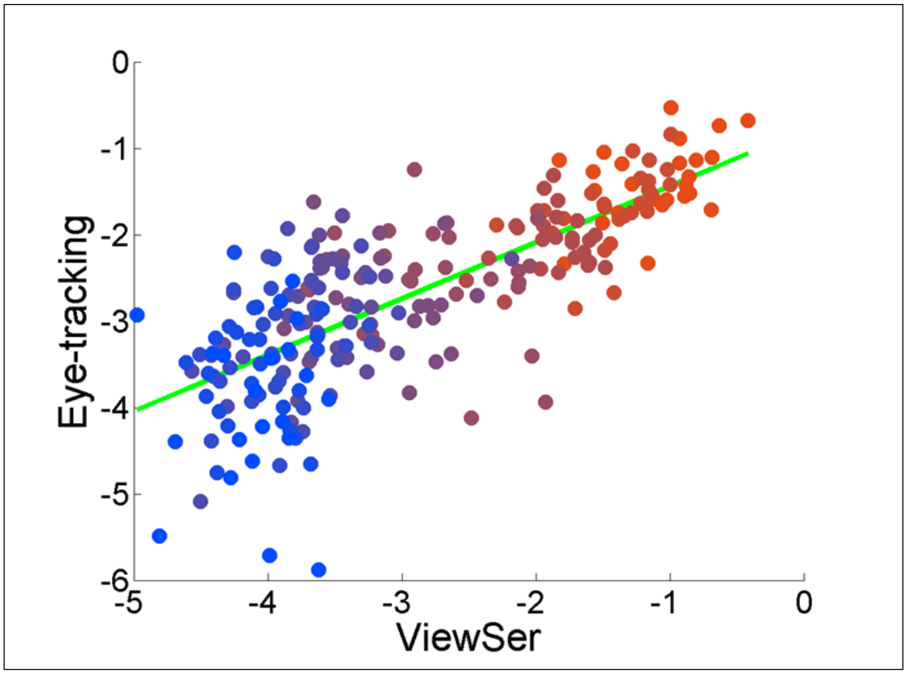


Figure 3.8: Logarithm of relative snippet viewing time for top 10 organic results measured with eye-tracking (y-axis) and ViewSer (x-axis). Color indicates position of the result in the list, i.e., red color corresponds to top results and blue color corresponds to results shown at lower positions (Pearson's $r = 0.74$).

Figure 3.6b reports the relative clickthrough rates for eye tracking and ViewSer participant groups. Each data point corresponds to the percent of all clicks for a query, landing on the corresponding result rank; these values are then averaged across all queries. In other words, the reported clickthrough rates are normalized for each query separately, and then averaged across all queries. We found that ViewSer group exhibits lower clickthrough rates on top results. We hypothesize that this is likely due to ViewSer interface encouraging more careful examination of the results in the top positions, resulting in lower rates of "indiscriminate" clicking frequently observed for top-ranked results [4].

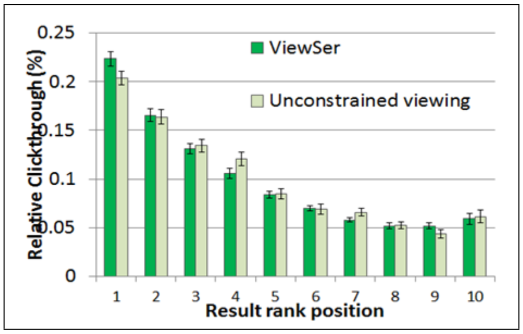


Figure 3.9: Clickthrough rates for ViewSer and Unconstrained viewing groups (Spearman rank correlation $r = 0.81$).

Comparative Analysis of Viewing and Clickthrough for Individual Queries

More detailed analysis of the Spearman correlation of viewing and clickthrough behavior for the ViewSer and Eye-Tracking groups for *individual queries* is reported in Figures 3.7a and 3.7b, respectively. For the vast majority of queries (over 80%), the correlation of the viewing and clickthrough behavior of the ViewSer and Eye-Tracking groups is over 0.8 and is never below 0.6, indicating that ViewSer provides a close approximation of eye tracking for over 80% of queries, and a moderate approximation for the remainder. To gain additional intuition of the relationship between Eye-tracking and ViewSer behavior for individual queries, we plot the relative viewing time measured using Eye tracking (Y axis) and ViewSer (X axis) for each result for all queries (Figure 3.8). The color shading indicates the results rank position, where the red color corresponds to rank 1, and the blue color to rank 10. The result viewing times, as measured by the two methods, correlate strongly (r = 0.74). Intuitively, results with higher ranks cluster in the top right quarter as both groups spend more time viewing higher-ranked results, as expected.

Further Analysis: ViewSer vs. Unconstrained Browsing

To validate ViewSer methodology further, namely, to determine whether ViewSer participants examine the SERP differently due to restricting of their peripheral vision, we performed a follow-up study with additional 25 MTurk workers. This final group enjoyed Unconstrained viewing of the SERP, without blurring of out-of-focus results. The clickthrough rates of this group are reported in Figure 3.9, together with the corresponding ViewSer clickthrough rates. Remarkably, the clickthrough behavior of these groups is similar, with Spearman correlation r = 0.81.

3.4 Summary

In this chapter we described and validated a scalable tool for conducing large scale studies of web page examination. Our tool, called ViewSer, accurately approximates time users spend viewing each search results as measured with conventional eye tracking method. In Section 6.1 we describe several practical application of ViewSer and demonstrate its utility detection of misleading search result snippets and web search result ranking.

Chapter 4

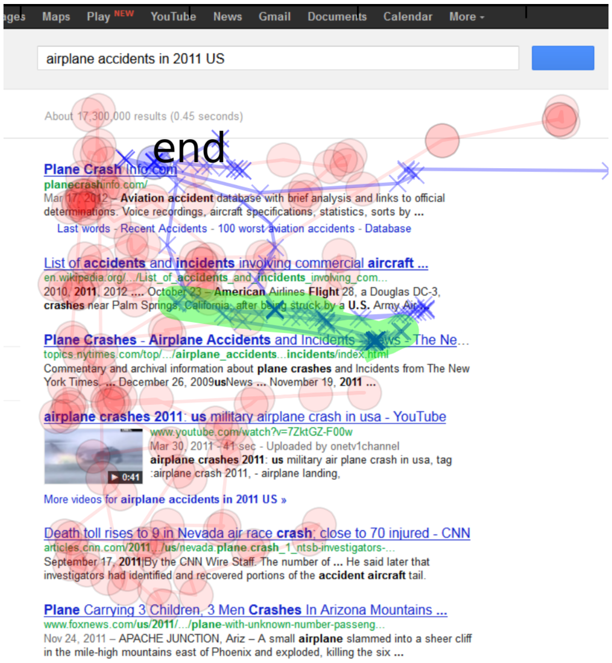
Discovery of Cursor Movement and Interaction Patterns

4.1 Background and Motivation

Millions of users interact with Web search engines daily. These interaction patterns contain valuable information, which could be useful for search engines to improve user experience, and for site designers to improve website layout and usability [7, 34, 23, 94]. Recently, studying fine-grained user behavior such as eye-gaze movements [46, 34, 22] and mouse cursor movements [116, 52, 66] have become an active area of research, as these interactions provide additional insights into searcher behavior compared to coarser models of clicks alone. In particular, recent work has demonstrated the coordination between eye gaze position and mouse cursor movements [116, 54, 63] and showed that both gaze and cursor interactions indicate user preferences [22, 66, 56]. Search engine companies also began investigating and modelling the cursor movement data to improve understanding of search result examination patterns [116], ranking of search results [64, 127], understanding of search result abandonment [66], and evaluation of content layout and noticeability [102].

While the importance of analyzing mouse cursor data for search is now evident, it often involves intensive manual effort [22, 56, 63] to gain insights about the data, and to make use of it for practical applications. For example, video recordings (or, similarly, a series of snapshots) from online analytic services, typically allow the replay of visitor interactions in great detail, but the process of viewing the replays is time-consuming – it is virtually impossible to view all the replays even for a relatively small site with thousands of daily visits, not to mention for larger web sites with millions of visitors. The alternative approach of visualizing areas of high cursor activity by using "heatmaps", that use different colors to indicate different levels of activity, provides a more complete view of the user behavior data in aggregate, but suffers from loss of detail about the sequences of interactions of individual users.

In this work, we propose a novel technique to *automatically* and *efficiently* extract common patterns from search result and landing page examination data, obtained via mouse cursor tracking.



airplane accidents in 2011 US

About 17,300,000 results (0.45 seconds)

Plane Crash Info.com
planecrashinfo.com/
Mar 17, 2012 – **Aviation accident** database with brief analysis and links to official determinations. Voice recordings, aircraft specifications, statistics, sorts by **...**
Last words - Recent Accidents - 100 worst aviation accidents - Database

List of **accidents** and **incidents** involving commercial **aircraft ...**
en.wikipedia.org/.../List_of_**accidents**_and_**incidents**_involving_com...
2010, **2011**, 2012 **....** October 23 – **American** Airlines **Flight** 28, a Douglas DC-3, **crashes** near Palm Springs, California, after being struck by a **U.S.** Army Air ...

Plane Crashes - **Airplane Accidents** and Incidents - News - The Ne...
topics.nytimes.com/top/.../**airplane_accidents**...**incidents**/index.html
Commentary and archival information about **plane crashes** and Incidents from The New York Times. **...** December 26, 2009**us**News **...** November 19, **2011 ...**

airplane crashes 2011: **us** military airplane crash in usa - YouTube
www.youtube.com/watch?v=7ZktGZ-F00w
Mar 30, 2011 - 41 sec - Uploaded by onetv1channel
airplane crashes 2011: **us** military air plane crash in usa, tag :airplane crash 2011, - airplane landing,
0:41
More videos for **airplane accidents in 2011 US** »

Death toll rises to 9 in Nevada air race **crash**; close to 70 injured - CNN
articles.cnn.com/**2011**/.../**us**/nevada.**plane.crash**_1_ntsb-investigators-...
September 17, **2011**|By the CNN Wire Staff. The number of **...** He said later that investigators had identified and recovered portions of the **accident aircraft** tail.

Plane Carrying 3 Children, 3 Men **Crashes** In Arizona Mountains **...**
www.foxnews.com/**us**/**2011**/.../**plane**-with-unknown-number-passeng...
Nov 24, 2011 – APACHE JUNCTION, Ariz – A small **airplane** slammed into a sheer cliff in the mile-high mountains east of Phoenix and exploded, killing the six **...**

Figure 4.1: An example of automatically discovered motif from mouse cursor data (shaded in green), corresponding to the common "follow" searcher behavior, where gaze (red circles) briefly follows the mouse cursor (blue crosses). The "end" label indicates the result click.

Our method, based on frequent subsequence mining, is able to capture common user- and location-invariant sequences from the mouse cursor data, some of which would be difficult to identify or describe by manual inspection or feature engineering. In data mining literature, such frequent sub-sequences have been called *motifs* [98] because of the analogy to their discrete counterparts in computational biology. For the rest of chapter we adapt the term *motif* to refer to a frequent pattern, representing a group of similar subsequences derived from the the mouse cursor movement data. We will define mouse cursor motifs more precisely in Section **??**.

An example of a common motif extracted from the mouse cursor data is shown in Figure 4.1. This motif corresponds to the common search behavior of "following" examination behavior (identified by Rodden et al. [116]). In this example, the user appears to examine the second and third search results, before returning to click on the first result. This behavior could be used to infer that a user has examined the second and third results and judged them to be non-relevant, providing valuable additional information to augment the click data.

As searchers examine results and pages at different rates, it is difficult to find exact matches between mouse cursor movements across different users. For this reason, we adapt a more robust distance measure, namely Dynamic Time Warping (DTW), that is capable to identify similar mouse cursor trajectories. However, discovering such motifs from large cursor movement datasets is computationally expensive, and is not feasible with existing motif mining techniques. To address this problem, we propose novel optimizations for motif discovery, specifically designed for cursor movement data, based on spatial indexing and learning-based similarity metrics. These optimizations enable an order of magnitude speed-up in motif discovery on realistic datasets. As a practical application (described in Section 6.2), the cursor movement motifs discovered by our approach can then be used as *features* for more accurate estimation of search result relevance and for significantly improving the quality of search result ranking.

Rest of the chapter is structured as follows. First, we define mouse cursor motifs more precisely, and formally state the problem. Then, we describe, in turn, the key components of our solution, that involves first generating and pre-processing many possible candidate subsequences, and then efficiently computing the similarity between them to find the groups of similar subsequences to discover the frequent motifs. Thus, the steps are, respectively, *Candidate Generation and Pre-Processing* (Section 4.3.1), *Similarity Computation* (Section 4.3.3) and *Efficient Computation of a Distance Measure* (Section 4.3.4), which together comprise our frequent motif discovery method.

4.2 Problem Statement

We start with introducing the necessary notation to define a motif and our problem. Intuitively, we first need to define what constitutes a non-trivial *match* between subsequences in cursor movement data, and then use this definition to define frequent subsequences (or motifs), and in turn state our motif discovery problem more precisely.

Match: Given a positive real number R as the range, or maximum distance, and a dataset of time series of cursor movements containing subsequences A and B, then B is called a matching subsequence of A, if $Dist(A, B) \leq R$, and A and B were recorded from *different* page visits. Here $Dist(\cdot, \cdot)$ a generic distance measure, such as Euclidian Distance or Dynamic Time Warping (DTW), defined below. The reason to insist on different page visits is to avoid "trivial" matches, most notably where A and B significantly overlap.

Motif: For a set of two dimensional time series $T = \{(x_i, y_i)\}_{i=1}^{N}$, and a subsequence length n, a *motif* M is defined as the subsequence M in T that has at least *min_count* matches, as defined above.

Cursor Motif Discovery Problem: Given a set of two-dimensional time series representing mouse cursor movements $T = \{(x_i, y_i)\}_{i=1}^{N}$, the range R, the subsequence length n, and a threshold *min_count*, *find all motifs* with match count higher than (*min_count*) and the distance $Dist(\cdot, \cdot)$ between each two of them is at least R.

Algorithm 1 FindMotifs

```
function FindMotifs($candidates, R, min\_count$)
    $motifs \leftarrow \{\emptyset\}$
    for $i = 1$ to length(candidates) do
        $similar \leftarrow FindSimilar(candidates, i, R)$
        $distinct \leftarrow DeDuplicate(similar, R)$
        if length $(distinct) \geq min\_count$ then
            $motifs \leftarrow motifs \cup \{i\}$
        end if
    end for
    return $motifs$
end function
```

4.3 Cursor Motif Discovery

4.3.1 Candidate generation and pre-processing

At this step our system creates all possible motif candidates that will be matched against each other in the similarity search. The candidate motifs are generated by maintaining a sliding window of a given length and shifting it for every example in the database. Every shift of the time window creates a motif candidate. In our experiments we used a sliding window of 5 seconds. We experimented with different values during development, and chose 5 seconds as long enough to capture interesting behavior patterns, and yet short enough to be able to capture short-term page visits. Other parameter values might be possible, to be explored further in future work.

After a candidate sequence has been generated, we normalize the values by subtracting the means of the x and y coordinates for that given candidate. This step is crucial as it allows us to match subsequences in different regions of the page, focusing on the their *shape similarity* instead of mining sub-sequences that occur in the same region area of screen. We do not otherwise rescale the values: in our development experiments with eye-gaze data, *z-score* normalization (suggested by Keogh et al. [111]) resulted in poor matches, as it leads to matching sub-sequences with large range to small oscillations of eye-gaze within the fixation.

4.3.2 Distance Measure

A distance measure defines the similarity between different motif candidates for grouping. As discussed above, we adapted a robust distance measure, namely *Dynamic Time Warping (DTW)* [82]. DTW method calculates the smallest possible distance between two time series by aligning one time series with another, such that, distance between them is minimized. The example shown in Figure 4.2 motivates the choice of DTW for mining mouse cursor data. Figure 4.2 plots the x coordinates of a discovered cursor motif (shown in blue) along with similar, but not identical, cursor movements (shown in different colors). While all movements exhibit similar periodic behavior, each individual movement peaks at different point in time, making it impossible for a simple distance measure, e.g., Euclidean distance, to identify the common similarity between them. In contrast, DTW allows to warp the series in time, such that they are best aligned. The flexibility of DTW comes with an expense – the time required to compute DTW is $O(n^2)$, as opposed to $O(n)$ for Euclidean distance. To reduce the computation time, constrained version of DTW [117] is often used in practice. Constrained DTW disallows warping (aligning) points that are farther than W time steps from each other. In our system we employ DTW constrained with Sakoe-Chiba band [117] with $W = n/2$, where n is time series length. Finally, the distance between two cursor movements is defined as the *sum* of DTW distances on the x and y dimensions.

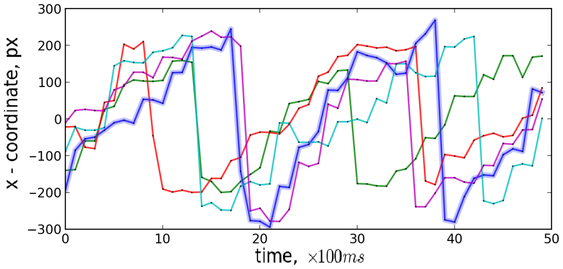


Figure 4.2: The x coordinate of an example motif (shown in blue) together with similar cursor movements recorded for different users (shown in different colors).

4.3.3 Candidate Similarity Computation

As the number of motif candidates can grow large (e.g., for even a small realistic dataset we use for experiments, there are several millions of candidate motifs and tens of millions of time series objects), we need an efficient way to search among the candidates to find similar objects. We employ *early abandonment* and *lower bounding* techniques, described below, that are commonly used in time series mining applications [111] which allow us to speed up the similarity search significantly. Algorithm 1 describes our FindMotifs algorithm more precisely. The algorithm starts with initially empty set of motifs and considers each motif candidate one-by-one. For each candidate *FindSimilar* function computes the raw number of matches, i.e. number of sub-sequences from the time series database that are similar (have distance smaller than a R) to the candidate. As motif candidates may match large number of overlapping sub-sequences, the match count computed by *FindSimilar* can be overestimated. To circumvent this problem, we only count matches that are distinct, i.e. are outside of R range from each other. Finally, if motif frequency defined as number of distinct matches exceeds *min_count* threshold the candidate is added to the set of discovered motifs. Note that exact implementation of *FindSimilar* depends on particular pruning strategy and is discussed below.

4.3.4 Scaling Up Motif Discovery

In order to scale up motif discover to realistic datasets, we adapt a number of optimization techniques, some well known, and some novel, combining them to speed up motif discovery by an order of magnitude.

Lower bounding: The idea of lower bounding has already become a standard technique to eliminate needless distance computations. In order to do that, one needs to compute a relatively cheap lower bound to see if DTW computation can be omitted. It is important to ensure that lower bound is exact, that is, it does not prune candidates in proximity of the distance threshold. Among several known lower bounds for DTW, the LB_Keogh lower bound is the commonly used solution due to its good pruning power and relatively fast computation time. LB_Keogh is calculated by computing Euclidian distance between "envelope" time series, hence it is $O(n)$. We implemented LB_Keogh as suggested and outlined in [111] and supporting web page of the *UCR-Suite*[1]. Other lower bounds are either more computationally expensive [95] or produce looser lower bounds, making pruning less efficient [83].

Early Abandoning: During the computation of either DTW or LB_Keogh, if our current value of lower bound measure exceeds a given distance threshold, we can safely abandon the computation for the remaining candidates at that point, since the resulting lower bound value will be higher than the distance threshold. Similar idea can be implemented for DTW. As this idea comes with no additional computation cost we employ it in multiple places in our code. For example, due to the nature of mouse tracking data we can early abandon after LB_Keogh or DTW values exceeded on either x or dimension.

Distance Measure Learning (Novel Optimization): As an alternate way to speed up similarity search directly, one can consider reducing dimensionality of time series data by either obtaining symbol representation such as *iSAX* [120] or Principal Component Analysis (PCA [76]). Neither of these approaches directly address the inaccuracy of time series measure calculated in the feature space caused by the reduction. Instead, we employ the idea of *learning* the distance metric, which was previously investigated by a number of authors [14, 30, 10]. Note that DTW is not a metric, and triangular inequality does not hold for it [95]. However, it is possible to approximate DTW by the Euclidian metric in the feature space. More specifically, feature space comprised of simple time series statistics such as standard deviation and range of x and y coordinates may well differentiate time series that are not similar and eliminate the need to compute the lower bound. In our experiments we use five features: standard deviation and ranges of x and y coordinates, respectively, and the mean squared speed of cursor movement. The intuition is that we can expect distinct mouse gestures to have different shape characteristics such as ranges, and speed, thus being sufficiently separated in the simplified feature space. To learn the feature weights directly from the data, we construct training and test data sets by sampling a large number of time series candidates, and computing the exact DTW for pairs of these time series. To obtain the weights of the features,

[1] UCR-Suite: http://www.cs.ucr.edu/ eamonn/UCRsuite.html

such that the error between the feature-based Euclidian metric and the true DTW of the original time series is minimized, we solve the following minimization problem:

$$\underset{w}{\text{minimize}} \qquad \sum_{i,j} \left(y_{ij}^2 - \sum_{k=1}^{d} w_k \left(x_i^{(k)} - x_j^{(k)} \right)^2 \right)^2 \tag{4.1}$$

where $x \in \mathbb{R}^d$ and d is the number of time series features; i and j are index i-th and j-th time series example in the training data, y_{ij} is the DTW for pair of these time series and indices i and j enumerate all the training examples. Since it is an unconstrained optimization problem, we can derive an efficient gradient descent-based method to find the feature weights w that minimize the error between the feature-based Euclidian metric and DTW on the original pair of time series. The resulting distance measure is referred as DM throughout the chapter. As the reduced feature space dimensionality is smaller than the typical length of time series we are interested in, such a measure allows us to index time series efficiently using any of the available spatial data structures, and subsequently pruning candidates that are unlikely to be similar at much smaller computational cost.

R-Tree Indexing (Novel Optimization): For efficient query processing and effective candidate pruning we employ spatial indexing using the R-Tree data structure. The R-Tree [59] is one of the most popular index structure for large multidimensional databases. Data in the R-tree is organized in a tree, where each node contains a bounding box of all entries in the corresponding subtree, and the leaf nodes store the data required for each child. In our case, the entries are points in five-dimensional space of similarity features described above.

It is known [11] that R-Tree performance degrades in high-dimensional spaces, where $d > 16$. In our case, the dimensionality is 5, allowing us to effectively reduce search space of candidates for the exact computation of DTW measures.

Combining Pruning Strategies: Clearly, combining several pruning techniques may speed up our algorithm. In this work we consider four system variations depending on the pruning utilized. Table 4.2 summarizes the pruning strategies enabled for each of the system variations we have tested. The combined system, which we call **DM-RTree**, is expected to scale well to large datasets.

We analyze the expected complexity of the approximate DM distance measure, using the notation in Table 4.1. The exact computation of DTW for a single query point requires $O(k^2)$ time, so the computation of all ϵ-neighbors for all instances will take $O(N^2k^2)$ time. The computation of LB_Keogh lower bound requires $O(k)$ time for a single point, so adoption of LB_Keogh pruning requires N^2 LB_Keogh computations, plus the time for computation of exact DTW for all selected points, a total of

$$O\left(N^2k + \tau_{\text{LB\_Keogh}}(\epsilon)N^2k^2\right)$$

time, which is better than exhaustive DTW computation, but also quadratic over the the number of sequences in the database. Using DM gives $\frac{d}{k}$ speedup at the pruning stage, and $\frac{\tau_{\text{LB\_Keogh}}(\epsilon)}{\tau_{\text{DM}}(\delta)}$

Notation	Description
N	the number of sequences in the database
k	average time sequence length
ϵ	distance threshold for DTW similarity
δ	Euclidean distance threshold for DM
$\tau_{\text{DM}}(\delta)$	pruning factor for DM, the average ratio of instances in δ-neighborhood of a query point
$\tau_{\text{LB\_Keogh}}(\epsilon)$	pruning factor for LB_Keogh
$\text{PP} = 1 - \tau_*$	pruning power of DM and LB_Keogh pruning
$d = 5$	dimension of approximate distance function

Table 4.1: Notations for Complexity Analysis

System Name	LB_Keogh	DM	DM-RTree
Exact			
LB_Keogh	✓		
DM	✓	✓	
DM-RTree	✓	✓	✓

Table 4.2: Pruning strategies considered for each system.

difference in DTW computation stage. Using R-Tree index allows us to eliminate N^2 multiplier in pruning stage. The height of R-Tree is $O(\log N)$, and for a sufficiently small δ, the R-Tree search time depends on output size as $O(\tau_{DM}(\delta) N \log N)$, so the complexity of our algorithm is:

$$O\left(\tau_{DM}(\delta) N^2 \left(d \log N + k^2\right)\right)$$

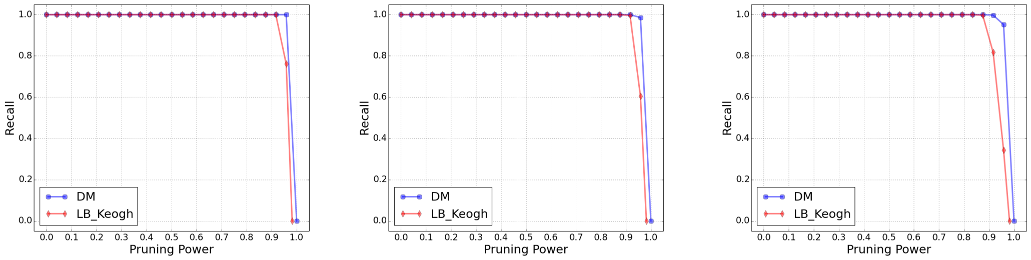
Thus, DM-RTree is expected to perform better than LB_Keogh pruning, if the pruning factor $\tau_{DM}(\delta) \ll 1$, as we verify experimentally in the next section.

4.4 Scalable Motif Discovery

This section demonstrates feasibility of our approach for large scale mouse cursor dataset, and evaluates its efficiency along with currently known techniques.

4.4.1 Experimental Setup and Dataset

We experiment with a dataset of mouse cursor movement collected “in the wild” using the EMU browser plugin [58] from over 5,000 real users of a university library. The EMU plugin recorded time-stamped events of user actions or changes in the web browser state, including url change, mouse cursor movements, clicks, page rendering and page content change events. The dataset contains 52,378 search engine result page (SERP) views, and 48,345 landing page views, resulting from over



(a) $\epsilon < 300$ (b) $\epsilon < 500$ (c) $\epsilon < 1000$

Figure 4.3: Recall of finding similar motifs vs. Pruning Power for LB_Keogh and DM methods, shown for representative values of ϵ.

31,860 queries. From this data, 100,723 cursor movement trajectory subsequences were extracted, comprising the sequence dataset for the experiments in the rest of this section. We use this dataset for empirical performance comparisons between LB_Keogh and our approach (Section 4.4.3), and to mine common motifs (Section 4.5) for subsequent relevance experiments.

4.4.2 Evaluating Distance Measure Learning

To verify the feasibility of effective distance measure learning we constructed the training and test data splits by randomly sampling a large number of time series pairs and calculating DTW for these pairs. Overall, our data set contains nearly 500,000 pairs and associated DTW distance values. The test and training datasets are split in equal proportions of 50%. We experimented with the five similarity features described above. We obtain the feature weights w by minimizing the objective function from Equation 4.1. Stochastic gradient descent is used to perform the optimization, converging in fewer than 10 iterations.

We now verify that we do not prune the truly similar candidates within ϵ-proximity of the query candidate. In other words, we investigate the "Recall" of our distance measure, vs. the corresponding "pruning power". Here, Recall is defined as the number of candidates found by our algorithm in ϵ-vicinity, divided by a total number of true positives according to exact DTW computation. *Pruning power* is defined as fraction of candidates pruned away early by our algorithm, and therefore eliminated from the (expensive) exact distance computation. By increasing the parameter δ, defined in Section 4.3.4, one can achieve better Recall, but lower Pruning Power, as more candidates are retained for exact distance computation. In our experiments we set the $\delta = 135$ resulting in desirable Recall of 95%. Figure 4.3 shows the Recall vs. Pruning Power curves for DM and LB_Keogh methods, with sub-figures corresponding to different values of ϵ. DM outperforms LB_Keogh exhibiting higher pruning power with 100% Recall, while significantly decreasing the computation cost.

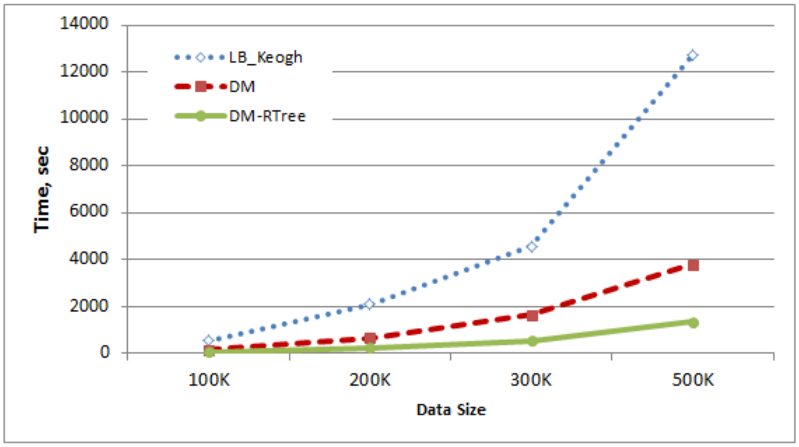


Figure 4.4: Comparison of running time for LB_Keogh, DM and DM-RTree systems.

4.4.3 Runtime Performance

In order to compare the performance of our final system DM-RTree (defined in Section 4.3.4), we performed benchmark tests for the three systems, using the dataset described above. Specifically, we compared: LB_Keogh, which uses pruning based on the lower bound; DM, our system based on distance measure with linear search among all candidates; and our DM-RTree system. We do not report the runtimes of the exact DTW computation, as it is 17 times slower than LB_Keogh, and more than 100 times slower than DM-RTree, and takes over several days to run for larger datasets. All experiments we performed on Intel Xeon CPU E5-2630 2.30GHz with 20 cores. Figure 4.4 reports the time performance for the systems as the size of the data is increased. Notably, DM-RTree system exhibits the lowest computation time, while growing at the slowest rate as the data size increases. The DM-RTree system outperforms the state-of-the art LB_Keogh by nearly an order of magnitude (8 - 9.5 times speedup), for different data sizes, without degrading Recall below 95%. Note that the speed-up factor increases with the data size, as the R-Tree index becomes more efficient in pruning candidates, compared to linear scanning performed by other systems. The benchmark test was repeated 10 times. For datasets greater than 200K, the differences in running times are significant with $p<0.01$.

4.5 Discovered Cursor Motifs

To demonstrate that our approach is not only efficient, but also effective in discovering meaningful motifs, we focus on the motifs extracted from the large dataset described above. At the same time, we investigate – whether common cursor motifs vary between search results pages and landing pages with possibly arbitrary layout. While applying motif extraction to a large dataset may result in

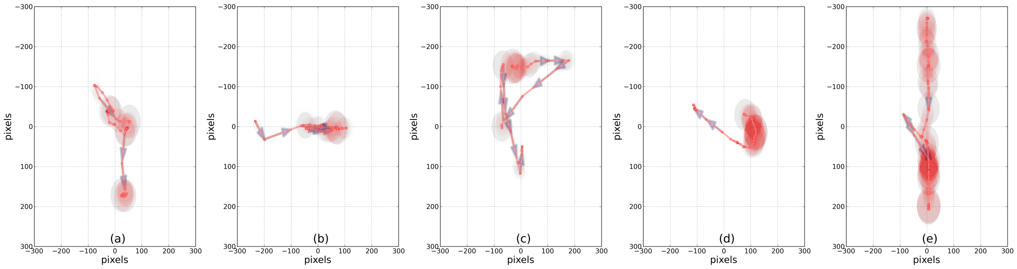


Figure 4.5: Top frequent motifs discovered from mouse traces recorded on search result pages (SERPs).

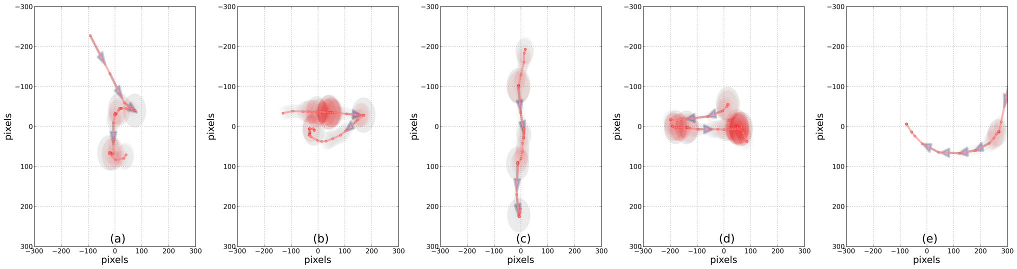


Figure 4.6: Top frequent motifs discovered from mouse traces recorded on landing pages (non-SERPs).

finding hundreds of motifs, we focus on the top 5 frequent motifs. Figure 4.5 reports the top 5 most frequent motifs (out of 127) discovered from the cursor movement data on the search result pages, or SERPs. Figure 4.6 reports top five most frequent motifs (out of 157), extracted from cursor data on the landing (clicked results) pages. In both figures, the mouse traces are annotated with arrows pointing in the direction of cursor movement, and shading of the circles indicates lower speed of cursor movement, or higher density of cursor positions. The common motifs extracted from SERPs correspond to previously (heuristically) identified patterns of cursor usage by Rodden et al. [116], such as marking of promising search results (a and d), and using mouse as a reading aid while following along a line of text (b), or interacting with the search query box (c), and following the attention vertically (e). Similarly, the discovered motifs on landing pages appear to indicate patterns of marking important information on page (a, b and d), and vertical movement (c) – potentially indicating the rapid shift of the user's attention downward, and directing the mouse cursor to click on a link (e), corresponding to heuristically identified patterns of cursor movement on landing pages in Guo et al. [56]. We emphasize that both SERP and landing page motifs above were discovered automatically, without changing the algorithm for the different page types.

4.6 Summary

In this Chapter we demonstrated that it is possible to automatically extract frequent mouse cursor movement patterns. The extracted patterns are helpful for understanding cursor movement behavior during web page browsing. As we demonstrate in Section 6.2, the motifs automatically discovered using our approach provide valuable information about the user's interactions with the result pages, and are effective for relevance estimation and ranking.

Chapter 5

Modeling Attention from Page Content and Interactions

5.1 Motivation

Attention data has successfully been used in various applications ranging from improvements of web site usability[88, 94] and automatic generation of attention biased summaries to high throughput behavioral tests that are able to detect memory impairment and attention deficit disorders [124]. While the information on how online users spend their attention during web page browsing appears to be useful for various search applications, it remains unclear how accurately one could predict actual eye gaze position on a web page using behavioral signals, such as cursor movements, clicks, etc., across various web page layouts. Moreover, it is not immediately clear, to what extent visual information about web page appearance is able to aid the prediction. In this chapter we review current approaches for modeling user attention from cursor and other interactions, and propose a probabilistic framework to incorporate information about web page content to allow more accurate prediction of the user's gaze position on a web page. To put our contribution in context, we compare our model to the previously published approaches.

The rest of the chapter is structured as follows. First, we review current techniques for attention modeling from mouse cursor movements. Then we survey computational models of visual saliency that aim to predict human attention on the visual stimuli based on local (pixel level) features, and discuss value of such models for attention modeling of web pages. Finally, we present a novel approach for attention modeling from user interactions and web page content, and compare it with previously proposed techniques that use only behavior or content information.

5.2 Attention Tracking from Cursor Movements

Tracking user attention from mouse cursor movements has long history within human computer interaction community. Chen et al. [27] was one of the first to study coordination patterns between mouse cursor and gaze. They classified mouse cursor movement into four distinct classes: “Stay Nowhere”, “Go Nowhere”, “Stay the Same Region” and “Go to New Region”. They found that distance between mouse cursor and gaze position was as low as 90 px for the region of the page where user attended. Chen and Liu [28] explored cursor-gaze coordination in context of online education websites. They proposed a methodology to detect page elements that attracted user attention. Their approach relied on partitioning the screen area (for a given resolution) into a number of “zones” and grouping all mouse movements that occur within a particular zone. They used Longest Common Subsequence (LCS) algorithm to find repetitive patterns in cursor movements on the screen and detect page areas that attracted user’s interest.

In context of Web search coordination between mouse cursor and eye movements was first studied by Rodden et al. [115]. They reported the alignment between user’s eye movements and mouse movements when scanning a web search results page, and identified three patterns of active mouse usage: *following the eye position vertically*, *following the eye position horizontally*, and *using the mouse to mark a relevant result.* Guo and Agichtein [53] proposed a natural extension Rodden’s work work - to predict eye-mouse coordination (i.e., whether the mouse cursor is in close proximity to eye gaze at any given point in time).

The works Rodden et al.[115] and Guo and Agichtein [53] were further extended by Huang et al. [62] to directly predict the gaze position from mouse cursor movement. As in prior work, Huang et al. reported that the extent of cursor-eye alignment is different for different types of cursor movements. They also found that cursor and eye are best aligned when user is performing click action, and have largest average distance in periods of cursor inactivity.

5.2.1 Notation and Data

Before discussing details user attention models we introduce notation related to eye and cursor movement data used throughout the chapter. We define each data point in our dataset $d_i = (\mathbf{x_i}, \mathbf{v_i})$, where $\mathbf{x_i} \in R^2$ is recorded position of the gaze on the screen, $\mathbf{v_i}$ is vector of features (e.g. mouse cursor features) and index $i \in [1, N]$ enumerates gaze data points. Our goal is to predict gaze position on the screen at a certain time given the feature values.

In order understand effect of web page content saliency on user attention and to compare effectiveness of different approaches for attention modeling we collected realistic dataset of aligned eye gaze, cursor interactions and annotated web page content data. Our data was collected in a cross-domain user study aiming to investigate effects of various factors on searcher attention. In this study we systematically varied *scope* of the search task and the search *domain.* This allowed us to collect gaze and mouse cursor data for a variety of information needs, search scopes and web pages. In the user study we experimented with two types of information need (*scope*) – *Focused* and *Broad* and five different search domains: *Web Search* (Google), *Shopping* (Amazon), *Social Network* (Twitter), *News* (CNN) and *Wikipedia*. *Focused* information need required users to find some specific information, e.g. “How many megapixels does Nexus 5 camera have?” in *Web Search* domain, while *Broad* information need had no specific answer and required users to read on or learn about a particular topic, e.g. "Learn what people on Twitter are saying about gay marriage" in the *Social Network* domain. In order to capture *natural* user behavior we designed the tasks to reflect typical information needs encountered in a particular domain. While during the user study we collected user interaction data on both search result and landing pages, in this work we analyze user behavior only on the search result pages (not the landing pages).

We recruited 20 users (11 males) using university bulletin board. Each user was asked to perform four practice tasks to become familiar with the study flow, followed by the 20 tasks that we use in our analysis. Among the 20 tasks, each user performed four tasks in each of the five domains, i.e. two tasks per each (*domain*, *scope*) condition. We randomized presentation order of the tasks to eliminate possible learning effects. To avoid possible confounding factors we balanced the study design by ensuring the same amount of data to be collected for each (*domain*, *scope*) pair.

To capture user's eye movements we used Tobii T60 eye tracker system built into a 17" monitor with 1280×1024 screen resolution and allowing us to record eye movement with frequency of 60 Hz. The eye movement data was processed using Tobii Studio software to obtain fixations and saccades. All user actions, including query input, page navigation, clicks and mouse cursor movements were recorded using custom extension to the Firefox internet browser. Overall, we collected eye movement and interactions data for 2890 page views, with 673 page views corresponding to the search pages.

5.2.2 Linear Regression

Huang et al. [62] proposed to directly predict the gaze position from mouse cursor movement. They used linear regression model (**LR**) to learn relationship between eye gaze and cursor movement features. Their model can be expressed in the following way:

$$f(\mathbf{v_i}) = \langle \mathbf{w}, \phi(\mathbf{v_i}) \rangle \tag{5.1}$$

where $\mathbf{w}$ - vector of feature weights, $\mathbf{v_i}$ is vector of features for i-th data point and $\phi(\mathbf{v_i}) = \mathbf{v_i}$. During the training we find optimal vector of parameters $\mathbf{w}$, such that discrepancy between model predictions and the actual gaze positions in the training data is minimized. In case of the **LR** model squared error between gaze and prediction positions is being minimized.

To better fit the data regression model often include a constant bias (intercept) term, which accounts for the average gaze position on the screen. Moreover, when **LR** is trained with gaze data recorded at a specific web page domain, e.g., Web Search, intercept term accounts for the average position of gaze on SERP which is determined by position of the content displayed to the user.

5.2.3 Non-Linear Regression

Recently, more sophisticated model of attention tracking was proposed by Navalpakkam et al. [105]. The linear regression model from Section 5.2.2 assumes the same linear relationship between gaze positions and the features, wheres non-linear hierarchical model allows non-linearities be captured as well model predictions are being personalized according to each individual user. Their “global” model can be expressed with following equation:

$$f(\mathbf{v_i}) = \langle \mathbf{w}, \phi(\mathbf{v_i}) \rangle \tag{5.2}$$

where $\mathbf{w}$ - vector of feature weights, $\mathbf{v_i}$ is vector of features for i-th datapoint and $\phi(\mathbf{v_i})$ is the Nystrom approximation of Gaussian radial basis function kernel matrix [121][1]. During model training find the optimal $\mathbf{w}$ so that it minimizes discrepancy between model predictions and gaze positions in the training data.

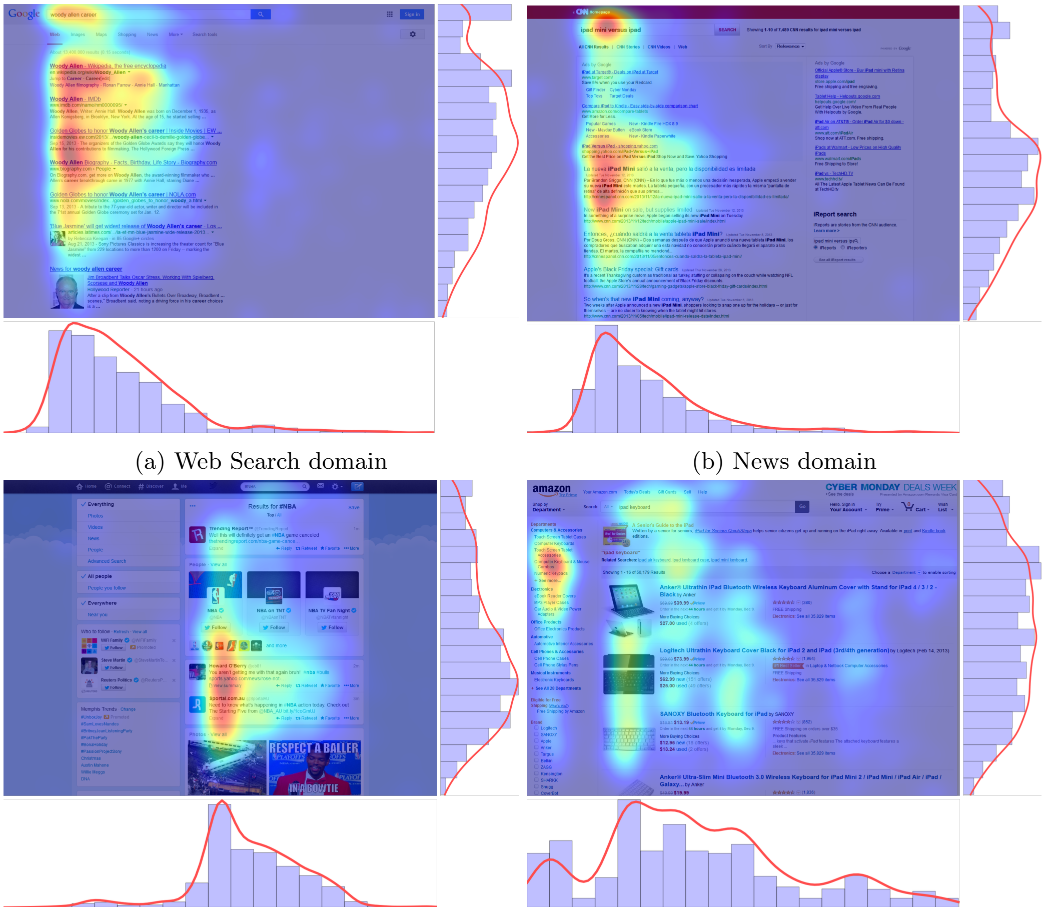
5.3 Effect of Web Page Content on Attention Distribution

Furthermore, to investigate to what extent average gaze on the screen for different web page types or domains we data from a controlled user study described in Section 5.2.1. If the average gaze position is similar across different web page domains, then the regression models trained a limited sample of the data (eye tracking data can only be collected at for a relatively small sample of web pages) would be able to generalize and provide accurate across different web page domains. In the subsequent section we will see to what extent this assumption holds.

Figure 5.1 shows heatmap (i.e. distribution) of attention computed from all page views in each of the web page domains. For purpose of discussion we only focus on web pages with search interfaces in four different domains: *Web Search*, *News*, *Shopping* and *Social Network*. In each heatmap the color indicates amount of the time user spent viewing a particular region of the screen (red color corresponds to large amount of time). As our goal is to analyze typical distribution of attention in web page each domain, we overlay heatmaps with a *typical* search page screenshot in the domain.

Figure 7.3c shows distribution of attention in *Web Search* domain. Similar pattern was previously reported by several studies (e.g., [99]) and was termed as “F-shape” or “golden triangle” due to the specific shape of attention distribution that decays on both x and y directions and resembles triangle. The decay in viewing along y direction is often referred to as position bias in result examination (higher results receive more attention because of the higher position and not necessarily higher relevance). While we clearly observe position bias on y direction, the rate of attention decay on x is arguably even more pronounced in our data, suggesting that users do not often read the entire line of text from left to right, but rather examine several words in the beginning of the result title. The latter fact was not emphasized in prior research.

[1] In this work we use N=500 basis vectors to approximate the kernel matrix.



(a) Web Search domain

(b) News domain

(c) Social Network domain

(d) Shopping domain

Figure 5.1: User attention heatmaps aggregated for all searches in each domain. Color indicates time spent on viewing a particualr region of the screen (red color indicates larger amount of time, blue color indicates smaller amount of time).

Figure 5.1b shows attention distribution for the *News* domain. Compared to the *Web Search*, the *News* domain has significantly more sponsored search results. One large block of sponsored search results is positioned on top of the organic results and another one is displayed on the right side of the page. Upon issuing the query, users not familiar with page layout in this domain seemed to be attracted by the sponsored search results, spending significant time on them (before realizing that the organic results are located closer to the center of the page). We observe the familiar triangular pattern of attention positioned next to the first organic search result. The attention decays more rapidly on x and y dimensions than in the *Web Search* domain. Due to absence of commercial intent in user study tasks participants spent very little viewing advertisements on the right side.

In the *Social Network* domain (Figure 5.1c) we find that user attention distribution is very different from *Web Search* and *News* domains. On the vertical dimension most of the attention is concentrated slightly below the center of the screen. The probability mass is also shifted along x direction – users mostly focus on Twitter posts appearing in the right half of the screen. We think this might be due to several reasons. The first is heterogeneity of Twitter search results, which are often comprised from most relevant *user* profiles (displayed at top of the page) and most relevant Twitter *posts* (displayed towards the bottom of the first screen). The second is that participants preferred to keep their sight position relatively constant, and scroll down the page to retrieve more results from the stream spending most of their time viewing posts in the bottom half of the screen.

Figure 5.1d shows the distribution of attention in the *Shopping* domain. In contrast to other domains where attention is concentrated on main page content on x axis, in Shopping domain we find more uniform distribution across the horizontal and vertical dimensions. Remarkably, participants spent significant amounts of time on the faceted search feature of the search interface (signified with hottest spot on the Figure 5.1d).

To summarize, we find that web page content (layout) has significant impact on distribution of user attention. These differences are caused by variation in web page layout and structure, which underlines a need for better understanding of the interplay between web page content and user attention. Most importantly, these results confirm our intuition that regression models are unlikely to produce accurate predictions of gaze position across different page domains (since they lack page content information). In rest of the chapter we attempt to design an efficient approach for incorporating web page content information into the gaze prediction model.

5.4 Computational Visual Saliency

Instead of relying on availability of mouse cursor data, computational visual saliency models take more principled approach and aim to model visual attention given an arbitrary visual stimuli (e.g., represented as pixel based image). In this section we review most basic concepts of computational visual saliency models relevant to our work.

Problems of understanding how human brain processes visual information captivated attention of the scientists for a long time. For instance, certain areas of theoretical neuroscience, such as those dealing with computation modeling of human attention on a visual stimulus, boomed for almost three decades and produced numerous models (e.g. [123, 86, 67]) to explain neurological mechanisms driving human attention. Such models take some information about visual stimulus (e.g. pixel based image) as an input and produce a single saliency map which assigns high values to locations where human's gaze is most likely to attend. Computational models of visual saliency can be roughly categorized into two groups: those, modeling attention from low level features of the stimulus (e.g. differences in color, contrast and orientation) are called *bottom-up* models, and those, modeling attention with additional information about viewing task (e.g. target position or information about objects present on the stimulus) - *top-down* models. Figure 5.2 illustrates a *bottom-up* computational framework for visual saliency introduced by Itti [67]. As many *bottom-up* approaches it models human attention as a feed-forward neural network. That is, it takes various features of the stimulus, such as color contrast, gradient and motion maps, as input, and produces a single saliency map that highlights locations where human gaze is mostly likely to attend on the input stimulus. According to some theories [67] representation similar to the saliency map may be used by the brain to control human's oculomotor system and direct gaze to move on the stimulus.

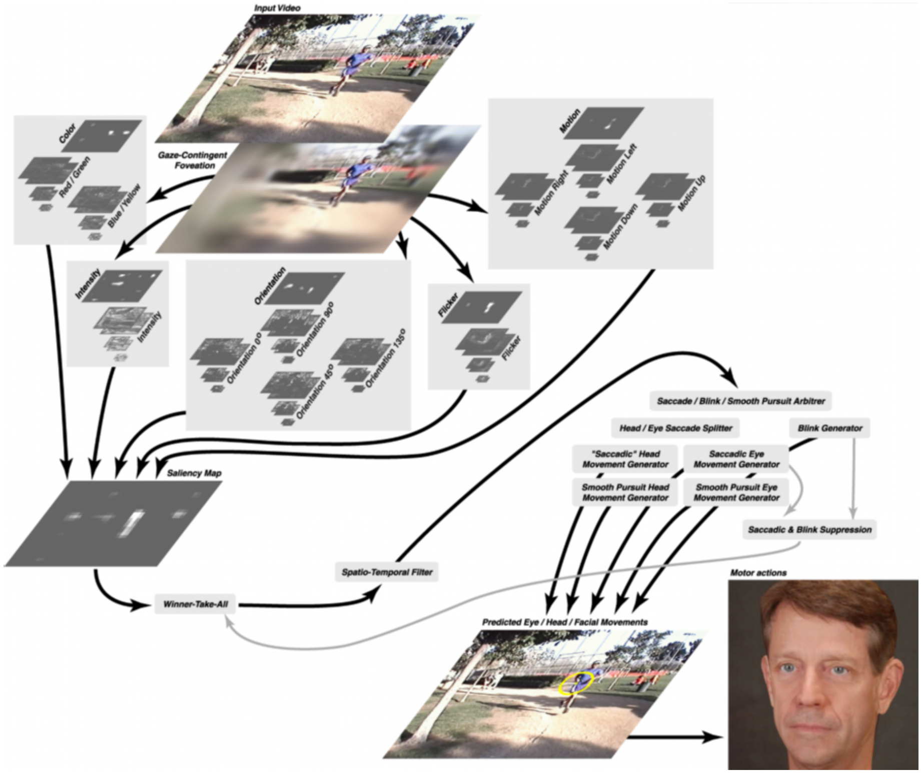

Input Video
Color
Red / Green
Blue / Yellow
Gaze-Contingent Foveation
Motion
Motion Right
Motion Left
Motion Down
Motion Up
Intensity
Intensity
Orientation
Orientation 0°
Orientation 90°
Orientation 45°
Orientation 135°
Flicker
Flicker
Saccade / Blink / Smooth Pursuit Arbitrer
Head / Eye Saccade Splitter
Blink Generator
"Saccadic" Head Movement Generator
Saccadic Eye Movement Generator
Smooth Pursuit Head Movement Generator
Smooth Pursuit Eye Movement Generator
Saccadic & Blink Suppression
Saliency Map
Spatio-Temporal Filter
Winner-Take-All
Motor actions
Predicted Eye / Head / Facial Movements

Figure 5.2: A simple framework for computing visual saliency of Itti [68].

Bottom-up models make almost no assumptions about the visual stimuli, which makes them very attractive for a variety of applications [68, 124, 17], however, accuracy of their predictions quickly deteriorates if relatively long (more than 5 seconds) prediction window is considered.

On the other hand, *top-down* models [103, 104] aim to account for longer tendencies in the gaze movements and incorporate information about a potential target during the viewing. While *top-down* approaches offer increased accuracy in a longer term, they require additional task or target information, that is rarely available. Hence, pool of applications of the *top-down* models remained limited[2]. Thus, for comparison with our approach we choose state of the art bottom-up model described in the Section 5.4.1.

5.4.1 Graph Based Visual Saliency

Among many *bottom-up* saliency models Graph Based Visual Saliency (GBVS) model achieved good performance in predicting of human fixations in the image viewing task [60]. The key idea behind GBVS model in its ability to model relative importance of each location on the saliency map. GBVS assumes that each location of the saliency map interacts with all $n-1$ locations of the the original stimulus' feature map, thus forming fully a connected network of the neuron interactions. The strength of such interactions is determined by the distance between two interacting locations and similarity of their feature values. Harel et al. [60] defined each connection weight as following:

$$w((i,j),(p,q)) \propto d((i,j)||(p,q)) \times F(i-p, j-q), \text{where} \tag{5.3}$$

$$F(a,b) \propto exp(-\frac{a^2+b^2}{2\sigma^2}) \tag{5.4}$$

where i, j, p, q are integers indexing two interacting locations on the feature map (i,j) and (p,q) and σ is a free parameter. Alternatively, GBVS model can be viewed as a Markovian process with transition probabilities proportional to the similarity weight w. The activation map is then defined as the stationary distribution of this Markovian chain. In our experiments (Section 5.5.1) we use GBVS visual saliency model[3] as a baseline to compare performance of different approaches for predicting human fixations during information seeking task.

To summarize, computational visual saliency models allow great extent of flexibility in modeling content and are capable of handling arbitrary visual stimuli. However, this flexibility comes with an added computational burden, e.g., to account for pixel-to-pixel interaction modeled by GBVS or pixel-level saliency maps required by Itti's model. In this work we try to strike a balance between model's expensiveness and the required computational cost.

5.5 MICS: Mixture of Interactions and Content Saliency

In this section we present our approach that allows us to more effectively infer user attention on Web pages by combining content and interaction signals. This is an even more challenging problem than predicting attention in images, as is done in computational visual salience research: Web pages contain extensive layout structure and multiple layers of meaning encoded in the text, layout, and metadata about a page.

[2] For more details on computation modeling of visual saliency reader is referred to the Related Work (Section 2.2) and a comprehensive review on the topic [60].

[3] We use publicly available implementation of GBVS from `http://www.klab.caltech.edu/~harel/share/gbvs.php`

To address this challenge, we exploit the observation that web pages, more so than image-only stimuli, can be effectively annotated with areas of interests (i.e., potential targets in the *top-down* models terminology), that can enable more accurate modeling of user gaze during web browsing or information seeking activities. Such annotations can be based on set of rules or rely on an automatic classifier to segment page elements that take part in the model. While the ultimate accuracy of the model is likely to depend on the quality of the page segmentation, for now let us assume that for popular types of Web pages (e.g., Search Engine Result pages or social media news feeds) such segmentation is available. The details of the particular page segmentation algorithm used in this work are provided in Section 5.6.2. Assuming a page segmentation is given to us, we can now define our Mixture of Interactions and Content Salience (MICS) model.

5.5.1 Definition

Our approach to modeling the allocation of user attention on a page is derived from the general idea of the *mixture of experts* model in machine learning [69]. As our goal is to estimate the task-specific (top-down) element importance on the page, and then *refine* the prediction based on how long each element on a page was displayed to the user, and where it was in the viewport[4] and what interactions user performed.

MICS operates by sub-dividing the visual space into regions - each corresponding to a particular Web page element. While the distribution of gaze positions within each element is determined only by the features of the element, the probability of attending a page element depends on relative attractiveness of *all the elements* displayed in the visible portion of the Web page. Intuitively, in our model each element “competes” for user attention against other visible elements on the page. Unlike previous approaches, which mainly use the visual stimuli information on pixel level (i.e., visual salience) to predict attention, our model takes advantage of the information about page element rendering (how elements are displayed by an Web browser to a user) and constructs compact, yet expressive, distribution of user attention on in the browser viewport.

More formally, our model defines a probability distribution of gaze position over the visual space (browser viewport) which is represented as a mixture of distributions - each corresponding to a particular web page element. This can be viewed as a particular type of a mixture of experts model (MICS, [69]), where each expert corresponds to a distribution representing the individual Web page elements. The reason MICS formulation is particularly well-suited to this setting is that it naturally manages uncertainty about the “attractiveness” of each element, which can be refined using additional features of the content element itself, or, later on, with interaction data.

MICS can also be viewed as a generative model. Figure 5.3 presents the *MICS* model diagram in plate notation. Table 5.1 defines the notation used in Figure 5.3. In the diagram, i stands for each data point, which consists of the set of elements and their locations on a page, visible at that time on the page, and the corresponding gaze position coordinates x_i. *MICS* states that the $i - th$

[4] We use the term *viewport* to denote the portion of a Web page visible to the user at given point of time.

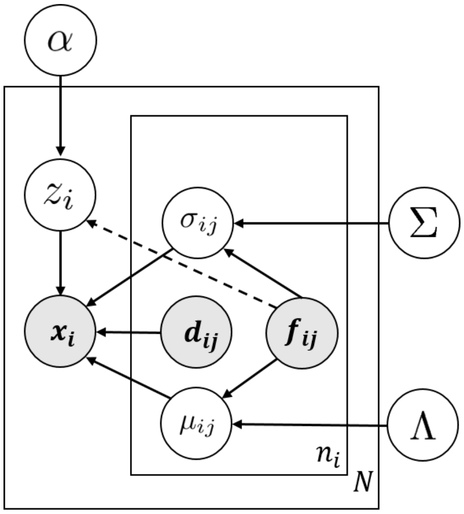


Figure 5.3: The *MICS* model for search attention modeling.

gaze position is generated from the observed element positions d_{ij} with their corresponding features $\mathbf{f_{ij}}$. The element's μ_{ij} parameters are defined as:

$$\mu^x_{ij} = d^{(x)}_{ij} + width_{ij} \cdot sigmoid(\Lambda \cdot \mathbf{f_{ij}})$$

where μ^x_{ij} is the horizontal component of element's Normal distribution mean parameter, $d^{(x)}_{ij}$ is the element's top left coordinate x, $width_{ij}$ is the width of the element, Λ is a free parameter estimated during training, and $\mathbf{f_{ij}}$ is vector of element's features. The element's variance parameter σ_{ij} is computed as:

$$\sigma_{ij} = exp(\Sigma \cdot \mathbf{f_{ij}})$$

where Σ is free parameter estimated during training. The probabilities of viewing an element are parametrized using the softmax function with the free parameter α:

$$P(z_i = j|\alpha, \mathbf{f}) = \frac{exp(\alpha \cdot \mathbf{f_{ij}})}{\sum_{j=1}^{n_i} exp(\alpha \cdot \mathbf{f_{ij}})}$$

5.5.2 Training

To make the model training more tractable, we make a simplifying assumption that all gaze positions are generated independently from each other. This allows us to derive an efficient inference and learning algorithm. Our algorithm learns the element importance weights α for the *MICS* model

Variable	Description
N	number of gaze data points
n_i	number of Web page elements at i-th viewport
$\mathbf{x_i} \in \mathbb{R}^2$	i-th gaze position
$z_i \in \{1, ..., n_i\}$	index of the Web page element being viewed at i-th viewport
$\mu_{ij} \in \mathbb{R}^2$	mean of the j-th element Normal distribution
$\sigma_{ij} \in \mathbb{R}^2$	variance of the j-th element Normal distribution
$d_{ij} \in \mathbb{R}^2$	position of the j-th element
p	dimensionality of element's feature space
$\alpha \in \mathbb{R}^p$	feature weights for the element importance distribution (z_i)
$\Lambda \in \mathbb{R}^{(p\times2)}$	feature weights for the element means μ_{ij}
$\Sigma \in \mathbb{R}^{(p\times2)}$	feature weights for the element variances (σ_{ij})
$\mathbf{f_j} \in \mathbb{R}^p$	feature vector of j-th page element.

Table 5.1: Summary of the notation used in the *MICS* model.

as follows. Let the dataset $D = \{\mathbf{x_i}\}_{i=1}^{N_k}$ collection of N_k gaze positions for k-th page view. Note that depending on the scroll position of the browser, there could be a different number of elements visible in the viewport, we denote this number as n_i. We assume that information about position of page elements (d_{ij}) and their features f_{ij} is available.

In order to find plausible values for model parameters $\Theta = \{\alpha, \Sigma, \Lambda\}$ we perform maximum likelihood estimation That is we optimize log-likelihood of gaze observations given the model parameters:

$$\mathcal{L}(\mathbf{x_i}|\Sigma, \Delta, \alpha) = \sum_{k=1}^{K} \sum_{i=1}^{N_k} \sum_{j=1}^{n_i} P(z_i = j) \log \mathcal{N}(\mathbf{x_i}|d_{ij} + \mu_{ij}, \sigma_{ij}^2)$$

In order to optimize the log-likelihood we use Stochastic Gradient Ascent (SGA) method with learning rate annealing. The model is implemented using symbolic differentiation tool Theano[13] that automatically generates code for gradient computation.

5.5.3 Inference

Once the *MICS* model is trained, gaze prediction distribution is computed as:

$$P(\mathbf{x_i}|\Sigma, \Delta, \alpha) = \sum_{j=1}^{n_i} P(z_i = j|\alpha, \mathbf{f_{ij}}) \mathcal{N}(d_{ij} + \mu_{ij}, \sigma_{ij}^2)$$

Note that $P(z_{ij} = j)$ gives us an importance weight (from 0 to 1) for each of page element d_{ij}. Thus, we could view it as a mixture distribution of n_i Normal distributions associated with the attractiveness and uncertainty predicted for each element, respectively. Computing the density of this distribution over a fixed grid of 2-dimensional points is tractable, as we demonstrate in the experiments section. Given the predicted density, the expected gaze position can be obtained by computing the maximum likelihood estimate of the x and y values under the predicted density distribution.

Since *MICS* is a generative model, for completeness we describe the generative process of how gaze positions could be generated by our trained model. The following generative process can be used to generate an i-th sample from our model:

1. Generate z_i with probability $P(z_i = j|\alpha, \mathbf{f_{ij}})$
2. Generate gaze position $\mathbf{x_i} \sim \mathcal{N}(d_{ij} + \mu_{ij}, \sigma^2_{ij})$, where d_{ij} is position of j-th element on the screen,

In practice, in order to avoid computationally costly sampling procedure, we could obtain estimates of gaze positions by numerically computing expectation over the predictive density:

$$\bar{\mathbf{x}}_\mathbf{i} = \sum_{l=1}^{L} \sum_{m=1}^{M} \mathbf{x_{lm}} P(\mathbf{x_{lm}}|\Sigma, \Delta, \alpha, \mathbf{f_{ij}}, \mathbf{d_{ij}})$$

where L and M are number if integration points in $\mathbf{x_{lm}}$ are nodes in two dimensional grid used for computing the expectation.

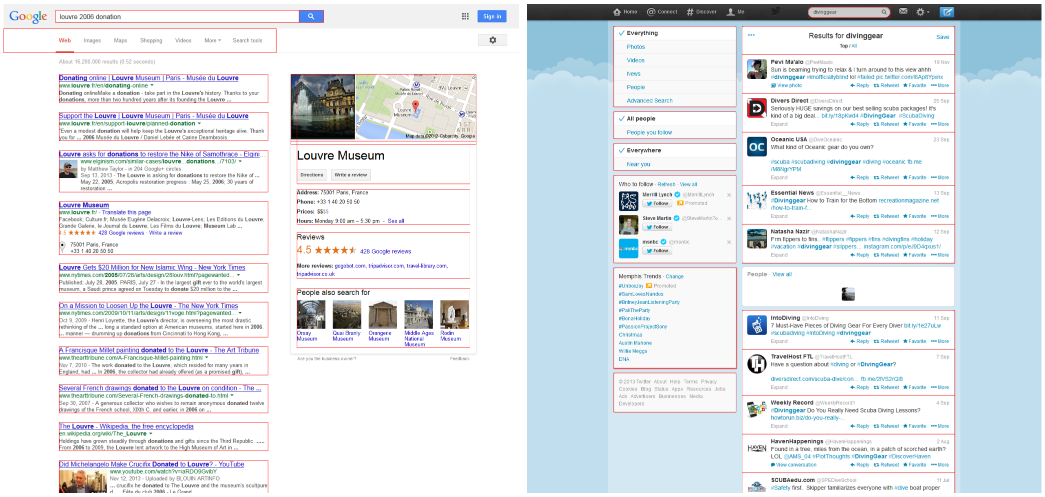
5.6 Experiments

5.6.1 Model Implementation

We now describe the specific implementation of the *MICS* model including Web page segmentation and content features that were used in this work.

5.6.2 Extracting Prominent Web Page Elements

Identifying most prominent Web page elements is not always a trivial task. Often, Web pages contain thousands of HTML elements, many of which are not even displayed to the user. As our goal is to model attention in presence of significant (visible and important) page elements, it is desirable to eliminate page elements that are unlikely to attract user attention, thus, considerably simplifying modeling complexity. To this end, our web page content analysis consists of first segmenting a web page into HTML DOM elements, then selecting a subset of the elements to consider, and finally extracting content features just from that subset. To take advantage of all Web pages in our dataset we employ both rule based segmentation, applied for frequent page types, and classifier based segmentation, applied for less frequent page types in our dataset. We would like to emphasize



(a)

(b)

Figure 5.4: Examples of page segmentation for the Google (a) and (b) Twitter pages. Segmented elements highlighted with red.

that this is just one of many ways to implement content element segmentation and other variations could be explored in future work to further improve performance.

For web pages that occur relatively frequently in our data, such Google search result pages or Twitter pages, we implement manually engineered segmentation. This is a common approach taken in previous work, and is applicable to a large and important subset of web pages which tend to share the same layout and page template.

For less frequent pages, we apply a supervised automatic classifier that for each web page *layout* element outputs a binary decision - whether this needs to be segmented or not. This makes our approach potentially applicable to a wider range of web pages. To perform this classification we use Gradient Boosting Decision Tree classifier (GBDT) [41]. The classifier uses page element's features to determine if element needs to be included or not. In order to train the this classifier we manually annotated page segmentation for 20 pages. Table 5.2 shows features used by our classifier. We utilized several types of information including the element's DOM Tree features (e.g. amount of links), the element's position information and size (e.g., width and height), as rendered by the browser at the time of page visit, and the element's style (e.g. visibility and text font size).

Figure 5.4 shows example of the page segmentation output for Google search result page and Twitter search. While the granularity of the segmented elements varies for different page types, we see that the elements carrying most important content information are captured. The fact that such segmentation only eliminates page elements that are not displayed in the browser or used only for layout or formatting, simplifies the salience modeling in a sense that we do not need to account for thousands of elements in our model.

5.6.3 Content and Interaction Features

We re-use content features employed by page segmentation algorithm (shown in Table 5.2, Content feature group). Our features encode information about element size, position on the page, style and font size, and simple information content measures such as number of words normalized by area. As discussed, additional more sophisticated content representation features could be invented, but in this reference implementation we opted for simplicity and generality. Despite the simplistic representation, the *MICS* model is able to use these features effectively, as shown in the experiments below.

MICS naturally allows to enrich the previously proposed regression models by allowing the features to be *element-specific*. For example, how *MICS* can exploit the information on how close is the mouse cursor to the particular element or whether a mouse cursor will hover over the element in the next few seconds. Such features allow the *MICS* model to learn cursor gaze coordination patterns not only on the overall behavior level, but on the element level as well. For example, if the mouse cursor hovers over the search box element, it is very likely that the user is going to reformulate the query terms, which implies the attention is focused on the search box. In contrast, if the cursor hovers over the elements located on the right side the search result page, it is less likely that the user's attention is following the cursor. Thus, to capture user interaction with the given element we include features that encode relative position of the cursor the element, cursor velocity, binary features indicating whether cursor is currently hovering the element or user is clicking on the element. Table 5.2 lists both Content and Interaction features used in our model. To account for a potential lag between interaction and eye gaze movement we concatenate features in the Interaction group at adjacent time d steps. The offsets for the adjacent time steps are $\{\pm 1, \pm 2, \pm 4, \pm 8, \pm 16\}$.

We train the *MICS* model using Stochastic Gradient Ascent algorithm with minibatch size = 100 and learning rate 0.001. To improve convergence speed we randomly shuffle training examples before start of the training.

To obtain the predicted gaze position using the *MICS* model the we use expected $\bar{\mathbf{x}}$ under the predicted attention distribution as described in Section 5.5.3 with number of integration steps $L = M = 100$.

5.6.4 Baseline Interaction Features

In our experiments we standard cursor movement that capture cursor trajectory at multiple time scales. This set of features mostly includes previously published characteristics [62, 105]. Foe each cursor data point we compute the following:

- Mouse cursor x and y positions
- The time since the page load
- The absolute values of cursor speed (vertical and horizontal), and movement direction (angle)

Group	**Feature Name**	**Description**
Content	Num{Links, Images,P }	Number of {a, img, p} tags in the given element
	IsTagName	Collection of binary features which equal 1 if the element's tag matches particular type (otherwise feature value is zero). The tags are $\langle a\rangle$, $\langle img\rangle$, $\langle p\rangle$, $\langle div\rangle$, $\langle span\rangle$, $\langle h1-h3\rangle$, $\langle em\rangle$, $\langle b\rangle$, $\langle li\rangle$, $\langle ol\rangle$, $\langle ul\rangle$
	Left, Top, Width, Height	Position and size information (in pixels)
	TimeOnPage	Time since the page load
	NumChildren	Number of child elements
	{Text, Image}Area	Total area of all {Text, Image} elements inside of the given element
	FontSize	Font size of the element's text
	TextToAreaRatio	Number of words (tokenized by white spaces) in the element divided by the element's area
Interaction	Cursor{X, Y}	Cursor position in pixels
	Speed{X, Y, Abs}	Horizontal, vertical and absolute speed of cursor movement
	Cursor{On, L, R, T, B}	Binary features indicating cursor position with respect to the element position (OnElement, Left, Right, Top, Bottom)
	CursorSame{Vert, Horiz}	Binary features indicating whether cursor position overlaps with the element vertically or horizontally
	DistX, DistY, DistEuclidean	Distance from the cursor to the element's center
	ClickOn	Non-zero if cursor click occurs on the element at given time step
	ClickDistX, ClickDistY, ClickDistEuclidean	Distance from the click position to the element center (zero if there is no click)
	TimeToScroll, TimeSinceScroll	Time since last scroll and time to the next scroll.
	OffsetFromScreenCenter{X, Y}	Vertical and horizontal offset of the element with respect to center of the viewport

Table 5.2: Content and interaction features used by MICS.

- The cursor distance traveled in the page up to this point
- The time and position of the most recent click on the page, if any
- The vertical scroll position, in pixels
- The time since the last scroll event, if any (if no scroll occurred the feature value is zero)

To account for longer range dependencies between the gaze and cursor movement for each time step we include features from previous time steps, logarithmically spaces in time, following the approach of [105]. The time step offsets were chosen as $\{\pm1, \pm2, \pm4, \pm8, \pm16\}$, capturing the 1.6 second contextual window of the cursor movements.

5.6.5 Evaluation Metrics

We evaluate our *MICS* in two tasks, prediction of aggregate attention distribution (saliency) and time dependent prediction of gaze positions (regression). In both tasks we adopt standard evaluation metrics.

To compare the performance of our *MICS* model on the aggregate attention prediction task, we employ three standard metrics used for evaluation of visual salience models [17]: Area under the ROC curve (AUC), Normalized Scanpath Saliency (NSS) [17], and the log-likelihood (LL) on holdout test data. We compute these metrics on fixation data described in Section 4.3.

The established approach to compute AUC in evaluating visual salience models is to measure the probability the model assigns to actual eye gaze positions (specifically, fixations) as positive examples, compared to "negative" positions (i.e., those not viewed). That is, for each positive example (observed fixation in the data), a negative example is chosen at random from somewhere else on the page [60]. More formally, given a prediction of distribution $P_{pred}(x, y)$ provided by a salience model, and a collection of n_f actual fixations for a page view, we uniformly sample n_f locations to form negative examples. Then, probabilities at fixated and "not-fixated" locations (given by $P_{pred}(x, y)$) are used to compute the ROC in a standard manner by varying the predicted probability threshold, which then gives the AUC value.

The Normalized Scanpath Salience (NSS) is another common metric used to evaluate salience models [17]. It measures how high a probability the salience model allocates to the actual observed eye gaze positions. Higher values of NSS indicate better agreement of the model with the actual gaze data. More formally, NSS is computed as:

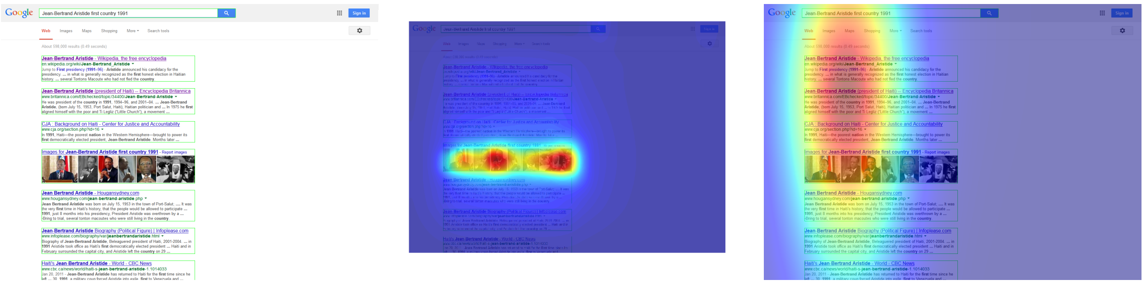
$$NSS = \frac{(P(x_i, y_i) - \mu_p)}{\sigma_p}$$

where μ_p and σ_p are the mean and the standard deviation of the predicted gaze distribution by a salience model, and $P(x_i, y_i)$ is the probability mass assigned by the model for the i-th fixation. Intuitively, $NSS = 1$ indicates that all of the user's fixations fall in the region whose predicted density is one standard deviation above average. In contrast, $NSS < 0$ indicates that the model performs no better than picking a random position. Higher values of this metric should indicate better models.

Model log-likelihood on holdout data is a standard way to evaluate intrinsic model quality or ability to capture the holdout data characteristics. Specifically, for the actual gaze positions in the holdout data, we evaluate the probability that the model assigns to the positive examples, computed as:

$$\mathcal{L} = \frac{1}{n_f} \sum_{i=1}^{n_f} log P_p(x_i, y_i)$$

.



Jean-Bertrand Aristide first country 1991

About 598,000 results (0.49 seconds)

Jean-Bertrand Aristide - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Jean-Bertrand_Aristide
Jump to First presidency (1991–96) - Aristide announced his candidacy for the presidency. ... in what is generally recognized as the first honest election in Haitian history. ... several Tontons Macoute who had not fled the country.

Jean-Bertrand Aristide (president of Haiti) -- Encyclopedia Britannica
www.britannica.com/EBchecked/topic/34400/Jean-Bertrand-Aristide
He was president of the country in 1991, 1994–96, and 2001–04. ... Jean-Bertrand Aristide, (born July 15, 1953, Port Salut, Haiti), Haitian politician and ... In 1975 he first aligned himself with the poor and Ti Legliz ("Little Church"), a movement ...

CJA : Background on Haiti - Center for Justice and Accountability
www.cja.org/section.php?id=16
In 1991, Haiti—the poorest nation in the Western Hemisphere—brought to power its first democratically elected president, Jean-Bertrand Aristide. Months later ...

Images for Jean-Bertrand Aristide first country 1991 - Report images

Jean Bertrand Aristide - Hougansydney.com
www.hougansydney.com/jean-bertrand-aristide.php
Jean Bertrand Aristide was born on July 15, 1953 in the town of Port-Salut; .... It was the very first time in Haiti's history, that the people would be allowed to participate ... 1991, just 8 months into his presidency, President Aristide was overthrown by a ... -Bring to trial, several tonton macoutes who were still living in the country.

Jean-Bertrand Aristide Biography (Political Figure) | Infoplease.com
www.infoplease.com/biography/var/jeanbertrandaristide.html
Biography of Jean-Bertrand Aristide, Beleaguered president of Haiti, 2001-2004. ... in 1991 Aristide took office as Haiti's first democratically elected president ... Haiti and in February surrounded the capital city, and Aristide left the country on 29 ...

Haiti's Jean-Bertrand Aristide - World - CBC News
www.cbc.ca/news/world/haiti-s-jean-bertrand-aristide-1.1014033
Jan 20, 2011 - Jean-Bertrand Aristide has returned to Haiti for the first time since he left ... 30, 1991, a military coup forced Aristide into exile, first to Venezuela and ...

(a) Original

(b) GBVS

(c) MICS

Figure 5.5: Example predictions of GBVS and MICS models for Web search result pages.

For comparing the regression performance of the *MICS* model against the baseline LR and KR models, we use the root mean squared error (RMSE) and mean absolute error (MAE) metrics, used in prior work for this task.

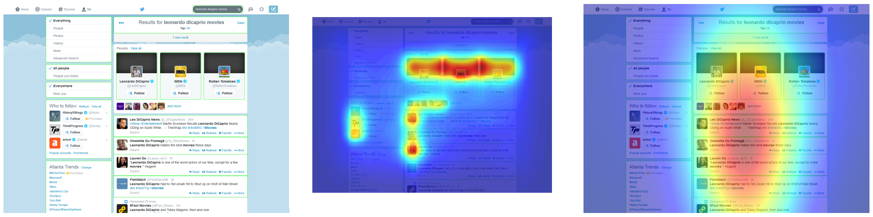
More formally, given a sequence of true and predicted gaze positions $\mathbf{x}^{(\mathbf{i})}_{\mathbf{gaze}}$ and $\mathbf{x}^{(\mathbf{i})}_{\mathbf{pred}}$, RMSE is computed as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |\mathbf{x}^{(\mathbf{i})}_{\mathbf{gaze}} - \mathbf{x}^{(\mathbf{i})}_{\mathbf{pred}}|^2}$$

where N is the number of gaze data points, $\mathbf{x}^{(\mathbf{i})}_{\mathbf{gaze}}$ is the actual gaze position at step i, and $\mathbf{x}^{(\mathbf{i})}_{\mathbf{pred}}$ is the predicted position, and the difference is the square of the Eucledian distance between the two. While RMSE is convenient from the optimization perspective (both LR and KR minimize the mean squared error, or MSE, on the training data), it dis-proportionally weights large errors. Therefore, we also consider mean absolute error, also used in prior work, which does not introduce this bias. The MAE is computed as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |\mathbf{x}^{(\mathbf{i})}_{\mathbf{gaze}} - \mathbf{x}^{(\mathbf{i})}_{\mathbf{pred}}|$$

where the sum is over the Euclidean distance between the actual and the predicted gaze positions. To achieve more robust estimates of models' performance, all the experiments were performed with 3-fold cross validation (CV). Each of the metrics is computed as the average across the hold-out (test) folds.



(a) Original (b) GBVS (c) MICS

Figure 5.6: Example predictions of GBVS and MICS models for Twitter pages.

Model	**AUC**	**NSS**	**Log-Likelihood**
GBVS	0.747	0.738	-3.11
MICS	**0.842** (+12.7%)	**1.266** (+71.5%)	**-2.79** (+10.3%)

Table 5.3: AUC, NSS and Log-Likelihood for GBVS and *MICS* models. *MICS* substantially and significantly outperforms the GBVS baseline on all evaluation metrics (p-value $<$ 0.001).

5.6.6 Comparison of MICS and GBVS

First, we compare whether *MICS* is able to capture aggregate pattern (i.e., distribution) of user attention on the page. Table 5.3 reports the performance of GBVS and *MICS* on the AUC, NSS and LL metrics. The *MICS* model consistently outperforms GBVS on all metrics, and achieves over 12.7% relative improvement on AUC, 71.5% improvement on NSS, and and 10.3% improvement on log likelihood. These improvements are all both substantial, and significant (p-value $<$ 0.001).

To gain better understanding on why our *MICS* model performs better than GBVS, we visualize the predictions of both models in Figures 5.5-5.6. Figure 5.5 shows an example Web search result page (a), with associated prediction given by GBVS (b) and *MICS* (c). Clearly, without having access to the features of the page elements used by *MICS* , GBVS assigns a large probability mass on search result from the Images vertical. In contrast, *MICS* is able to capture the well known position bias in search result examination, and assign a higher probability to the top search results. Recall, that *MICS* learns to capture this bias and associated uncertainty at training time. Also note that the variance of *MICS* prediction is asymmetric: the large variation in vertical position accounts for the uncertainty about the gaze position for this page. On the other hand, the relatively small variance in horizontal dimension accounts for the known effect that search results are not usually fully read - users often skim only the title or the first few words of the result before moving on to the next result [89].

In summary, our empirical results show that *MICS* is more suitable for modeling visual salience on Web pages than a state of the art computational visual salience tool (GBVS). In addition, our results show that *MICS* is able to accurately match the empirical gaze distributions on multiple different Web page layouts. Note that for this task, the interaction data was used by *MICS* only implicitly at training time (Section 3). We will exploit the interaction data more fully in the next

Domain	**Method**	$RMSE_x$	$RMSE_y$	**RMSE**	MAE_x	MAE_y	**MAE**
Web Search	LR	234.0	236.6	332.7 (N/A)	181.9	194.1	294.8 (N/A)
	KR	207.4	220.6	302.8 (-8%)	172.3	181.3	273.4 (-7%)
	MICS	156.1	202.6	**255.8 (-23%)**	128.8	160.1	**225.7 (-23%)**
News	LR	262.2	279.1	383.0 (N/A)	209.7	229.5	340.6 (N/A)
	KR	176.4	247.8	304.2 (-21%)	144.5	204.7	272.4 (-20%)
	MICS	174.5	208.3	**271.7 (-29%)**	138.1	167.0	**237.3 (-30%)**
Wikipedia	LR	219.7	242.0	326.8 (N/A)	173.6	195.5	288.1 (N/A)
	KR	290.0	272.5	398.0 (+22%)	242.5	223.9	360.8 (25%)
	MICS	87.2	277.4	**290.8 (-11%)**	70.2	210.9	**235.7 (-18%)**
Shopping	LR	249.1	259.2	359.5 (N/A)	196.3	211.2	319.5 (N/A)
	KR	281.5	285.9	401.2 (+12%)	225.6	231.1	359.3 (+12%)
	MICS	257.6	215.3	**335.7 (-7%)**	201.4	179.6	**298.6 (-7%)**
Social Network	LR	260.4	256.2	365.3 (N/A)	206.5	207.0	322.1 (N/A)
	KR	205.6	263.0	333.9 (-9%)	162.2	210.0	293.3 (-9%)
	MICS	146.9	187.0	**237.8 (-35%)**	113.3	146.7	**206.3 (-36%)**

Table 5.4: Predictions results for LR, KR and *MICS* , for different Web page domains. The *MICS* model consistently outperforms prior methods in all domains (differences in RMSE and MAE are significant p<0.001 with two tailed t-test).

section to address an even more challenging task of predicting where a particular user is looking at each specific time.

5.6.7 Comparison of MICS and Regression Models

Table 5.4 summarizes prediction performance for the baseline models LR and KR and our *MICS* model, averaged across the hold-out samples, in the cross validation setting. *MICS* performs significantly better than LR and KR in all of the domains ($p < 0.001$, two tailed t-test). Reduction in error varies from 7% in *Shopping* domain to 35% in *Social Network* domain $RMSE$=237.8 px. The lowest prediction error was obtained in the *Social Network* domain ($RMSE$=237.8px, MAE=206.3px), while the *Shopping* domain appeared to be the most difficult to predict resulting in the highest error ($RMSE$=335.7 px, MAE=298.6px). We believe that the reason for the large performance improvements lie in the additional power available to the *MICS* model. Both LR and KR models make strong assumptions about the relationship between gaze and cursor interactions, relying on a constant bias term independent of the actual content shown to the user. Since user attention distribution heavily depends on what is shown the screen (e.g, see Figure 1), a constant bias that works for different types of pages may not exist. In contrast, *MICS* , by design, follows the content, and is able to supply a multi-modal predictive distribution dictated by the Web page elements visible to the user.

Interestingly, on the Web search domain, *MICS* also exhibits substantial reduction in error on the horizontal dimension (RMSE$_x$ and MAE$_x$), making it even more appealing for evaluation –

when search results may be shown to the right of the organic search results [105]. Such results attempt to provide users with direct answers to their information needs without requiring users to click. Previously, it has been proposed [105, **?**] to utilize user attention for evaluation, providing a natural application of *MICS* for this task.

Our results demonstrate that it is possible to learn Web page element salience or attractiveness that is generalizable across different page types. This is even more encouraging since the Web search engines are constantly experimenting with various ways to improve user interface of search results and maintaining an attention model that can only work for a certain page configuration would severely impact its use cases. While *MICS* outperforms prior approaches in the gaze prediction task, it provides a general and principled way to integrate page content information into the attention model. The behavioral features allow MICS to make more sensible, time dependent predictions and capturing cursor-gaze coordination patterns.

5.7 Summary

In this chapter we described a novel approach for joint modeling of user attention from web page content and user interactions which we call *MICS* for Mixture of Content Saliency and Interactions. Presented approach outperforms state of the art baselines that use only content information or only user behavior information on standard evaluation metrics. MICS model provides significant reduction in root squared mean error and in mean absolute error. Our model automatically learns web page content attractiveness from the page element's features based and eye movement training data. Unlike previously proposed regression models, MICS is able to generalize across different Web page domains and layouts. In order model complex, context dependent, nature of eye movement our model effectively incorporates contextual information from user cursor interactions into the prediction. In addition to improved accuracy of gaze prediction our model enables novel application - an automatic Web page content optimization and attention guided design of the user interfaces.

Chapter 6

Applications to Web Search

In this Chapter we demonstrate several applications of mouse cursor data for detecting misleading search result snippets, document relevance prediction and attention biased automatic document summarization.

6.1 Restricted Focus Viewing

This section describes three practical applications of the ViewSer method (described in Chapter 3) to Web search. First, Section 6.1.1 describes collection of relevance rating used in our experiments for this section. Then, we describe how ViewSer could be used to analyze snippet attractiveness (Section 6.1.2), to improve result ranking (Section 6.1.3), and to detect misleading snippets (Section 6.1.4).

6.1.1 Relevance ratings collection

To validate our findings on a bigger dataset and explore some practical applications, we collected SERP examination data for an additional 50 queries taken from the HARD track of TREC 2005, resulting in a dataset of 75 queries. Separately from the ViewSer study, we collected comprehensive relevance judgments for all of the results on the first page of results, for all queries in the WEB Track and the HARD Track. The Mechanical Turk workers (MTurk) were recruited to perform the relevance labeling as described above.

To control worker accuracy in ViewSer group we obtained relevance ratings for documents of each query in our collection. Each MTurk HIT was to assess organic (non-sponsored) results for one query. Following the recommendation of [85], the authors labeled 10% of documents as a validation set in order to estimate the worker accuracy and verify the quality of their work. On average, results for each query were rated by 6 workers. Inter-rater agreement, computed with Fleiss Kappa was 0.39, which correspond to fair/moderate agreement. We conjecture that this level of agreement is caused by the difficulty of the tasks and the informational nature of the queries. These ratings were used to compute the worker's accuracy on validation set and to filter workers with low accuracy as unreliable. For the WEB track, 17 of 106 workers were filtered out, and for the HARD track 101 out of 263.

6.1.2 Snippet Attractiveness

In this section we present one possible usage of data collected with ViewSer to analyze snippet attractiveness. The importance of snippet attractiveness as a contributing factor of clickthrough patterns has been explored by a number of researchers [31, 35, 128]. Our work has the advantage that we can directly measure the ratio of the times a snippet was examined to the number of times it was clicked, which call the COV ratio. In other words, COV is defined as probability of clicking on result given that result was examined. We hypothesize that the COV ratio is, in fact, a measure of snippet attractiveness that is independent of the rank position.

Experimental Setup: As a first step, we validate our hypothesis that COV is not dependent on the rank position, and in fact can be used as an un-biased estimate of snippet attractiveness. To this end, we calculate Pearson correlation coefficient between the result rank position and number of times the result was examined, clicked, and ratio of these counts. Here we report our comparative analysis of the (COV) metric based on data from Eye-tracking and ViewSer groups. We also report Pearson correlation between the COV ratio and textual features of the snippets.

As [31, 128] indicate, there exists a strong position bias in the way results are viewed on the SERP - searchers browse the list of results from the top of the page to the bottom, which confirms previous findings [99], and more recently [65]. Such bias puts major obstacles of using click or result viewing time [65] feedback directly in search engine optimization. Different ways of eliminating of the presentation bias in clicks have been studied in [31, 128] as we highlight earlier in the text. In other words, application of additional techniques is required in order to extract useful signals from click data. It is reasonable to expect similar problems with viewing time measured using eye tracking or approximated with mouse hovering [53, 65]. However, our COV metric does not correlate with the result rank: the correlation between result rank and COV is 0.05 for the Eye-tracking group and 0.11 for the ViewSer group. This is a remarkable result, indicating that the COV ratio does not appear to be affected by result position bias.

We validated the COV ratio measured with ViewSer on our eye tracking data. Figure 6.1 shows the COV ratio broken down by result position. On average we have observed slightly higher COV values in ViewSer data in comparison to Eye-tracking. Overall, Pearson correlation coefficient between Eye-tracking and ViewSer groups computed for each individual result was 0.64, which indicates substantial correlation.

In order to understand how COV relates to the previous work on estimation of result attractiveness [31, 128], we analyzed the correlation between COV and the textual features of the snippets. Table 6.1 shows example features that we considered. While many of the features used have already been investigated in prior work, we have extended the feature list with features capturing the rich structure of the snippets. For example, the feature *summaryLinks* indicates whether a snippet has additional embedded hyperlinks to within-site navigation. Another example feature capturing complex snippets is *mapInSummary*, indicating whether a result contains a map with local search results. These features might be useful in predicting result attractiveness, as they can

Correlation	***Feature***
0.2009	*titleStartsWithQuery*
0.1195	*summarySentanceFragments*
0.1169	*urlQuery*
0.1118	*titleQueryMatch*
0.1080	*sumaryPunctuation*
0.1060	*homeInSummary*
0.1030	*mapInSummary*

Table 6.1: Snippet feature importance ranked by correlation to attractiveness.

change searcher's SERP examination behavior.

The Table 6.1 reports the correlation between the *COV* ratio and text features of the snippet. The feature *titleStartWithQuery* has the highest correlation of 0.2. Features *titleStartWithQuery, urlQuery, titleQueryMatch,* and *homeInSummary* have higher correlation with *COV* than other features, which confirms previous findings in references [31, 128].

As we show in the next subsection, using attractiveness as an additional feature can be helpful for important and practical web search tasks.

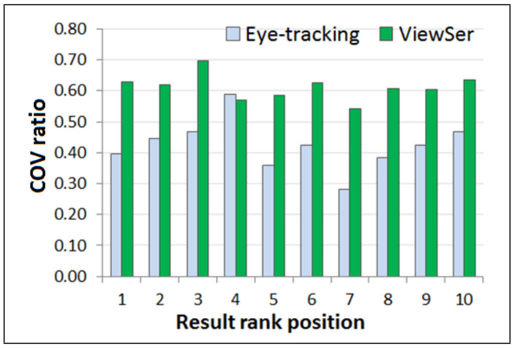


Figure 6.1: Clicks over Views (*COV*) for Eye-tracking and ViewSer groups, by rank position.

6.1.3 Search Result Re-ranking

Learning to rank has become a very popular approach to achieve better search results ranking. In this section we investigate whether attractiveness can be used as a feature in learning to rank framework to improve original ranking. Unfortunately, the *COV* statistic as an additional feature measured directly, based on user study, would not be practical for large scale LTR experiments, since it would require collecting viewing data for each individual result. Therefore, we build a regression model to predict result attractiveness based on the textual snippet features. For this purpose we built a regression model predicting *COV* ratio from textual features described in Section 6.1.2. Thus, we used two additional features for re-ranking: ***COV*** (Click over Views ratio) and

$\boldsymbol{A}$(estimated attractiveness) i.e., the estimated value of COV based on textual features.

Experimental Setup: We used the same query and document dataset as described in Section 5.1, providing labeled relevance data for 75 queries and 650 documents. We used *SVM-rank* [73] as the LTR method of choice. To estimate result attractiveness, we used the Gaussian Process regression model with radial basis function kernel. The correlation of the estimated and true COV values was 0.6 (3 fold cross-validation). Improving the estimation of snippet attractiveness will be part of our future work.

Results: Table 6.2 reports the NDCG [71] averaged across 3 folds for the baseline ranking system (Google) as well as for the re-ranking method, using directly measured COV and the the estimated attractiveness (A). The average NDCG of the original ranking was 0.8408. Our first experiment was to train a ranker based on the document position feature and the COV ratio, which yielded a significant improvement of 6% over the baseline. This substantial ranking improvement was surprising, given that the search engine was already highly optimized. Following the recommendation of [80], we performed significance test between the original ranking and our system, showing significance at $p < 0.05$. Once attractiveness (A) model was trained on the training data (2 folds) we use it to compute the attractiveness feature for the test fold. As reported in Table 6.2, the re-ranking performance based on the estimated value of snippet attractiveness, outperformed the Google baseline ranking by 5.25%, reaching NDCG of 0.8848 and being statistically significant with p¡0.05. The slight gap between P + COV and P + A results is due to the expected regression error, that can be further reduced with more training data or richer features. We also tried to add number of clicks received by document as an additional feature to the ranker, but the performance was slightly lower. Nevertheless, the demonstrated improvements are remarkable, considering the relatively small amount of training data that was required to estimate the snippet attractiveness and in turn improve the ranking over a state-of-the-art Google ranking.

Table 6.2: Ranking system comparison. P - rank position, COV - Clicks over Views, C - clicks, A - estimated attractiveness. * indicates significance at $p < 0.05$, ** indicates significance at $p < 0.01$.

Features	NDCG
Baseline (P)	0.8408($-$)
P + COV (ceiling)	0.8920(+6.09%)**
P +A	0.8848(+5.24%)$*$
P + A + Clicks	0.8840(+5.14%)*

6.1.4 Detecting Bad Snippets

In this section we describe our experiments on detecting bad (i.e., misleading) search snippets. Intuitively, good snippets should clearly summarize the result document so that searcher would be able to understand whether it is worth clicking or not. Specifically, we consider good snippets to be those, that attract clicks on relevant documents, or discourage clicks on non-relevant documents. Conversely, bad snippets would discourage clicks on relevant documents, while attracting clicks on non-relevant documents. More formally, we define snippets to be *Bad* or *Good* based on the snippet *COV* ratio (defined in Section 6.1.2), and the result relevance (manually labeled as described in Section 6.1.1):

$$Label(COV, REL) = \begin{cases} Bad & \text{if } (REL \geq 0 \text{ and } COV < \theta_2) \\ & \text{OR } (REL = 0 \text{ and } COV > \theta_1) \\ Good & \text{otherwise} \end{cases}$$

Where the parameters θ_1 and θ_2 are set empirically by manually examining a sample of the snippets and the documents. Thus, *Good* snippets for relevant documents would have higher (i.e., greater than θ_1) *COV* ratio, since a searcher would be more willing to visit the document after examining the snippet. Similarly, a *Good* snippet allows a searcher to identify a non-relevant document, resulting in lower *COV* ratio (i.e., less than θ_2). In contrast, a snippet is considered to be *Bad* if it fails to inform the searcher about the document relevance. Hence, a snippet for a relevant document that exhibits a low *COV* ratio (i.e., less than θ_2) is considered to be *Bad*. We experimented with different values of the θ_1 and θ_2 parameters and determined that the setting θ_1=0.85 and θ_2 =0.35 provides the closest match to our definition, on a subset of the data. With this parameter setting, our dataset contained 589 *Good* snippets and 61 *Bad* snippets.

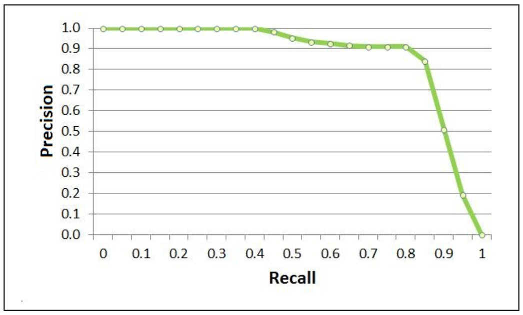
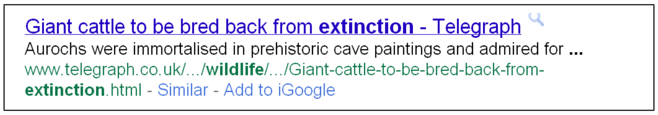


Figure 6.2: Precision vs. Recall for detecting bad snippets using ViewSer data.

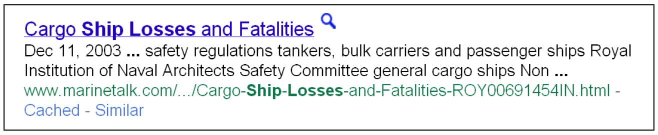
Figure 6.3 shows an example of a snippet that appeared in the results to the query "`wildlife extinction`" where the information need was described as *"The spotted owl episode in America highlighted U.S. efforts to prevent the extinction of wildlife species. What is not well known is the effort of other countries to prevent the demise of species native to their countries. What other countries have begun efforts to prevent such declines"*. The snippet summarizes a news article talking about recreating aurochs from reconstructed DNA, which is an attempt to save the specie, so the document was judged as a relevant to the query. However, the text summary of the snippet talks



Giant cattle to be bred back from **extinction** - Telegraph
Aurochs were immortalised in prehistoric cave paintings and admired for **...**
www.telegraph.co.uk/.../**wildlife**/.../Giant-cattle-to-be-bred-back-from-**extinction**.html - Similar - Add to iGoogle

Figure 6.3: An example of a bad snippet: document is relevant while the COV ratio is low (0.18). This snippet was returned for the query "`wildlife extinction`" at the 4th position. The snippet summarizes a news article talking about recreating aurochs from reconstructed DNA, which is an attempt to save the specie, so the document is relevant to the query. However, text summary of the snippet is not representative, causing searchers to skip the document.

about a seemingly unrelated fact that *"aurochs were immortalized in prehistoric cave paintings"*, which caused many of the searchers to skip this document. Another example of a bad snippet is shown in Figure 6.4, where a snippet appears relevant, but the actual document is not.



Cargo **Ship Losses** and Fatalities
Dec 11, 2003 **...** safety regulations tankers, bulk carriers and passenger ships Royal Institution of Naval Architects Safety Committee general cargo ships Non **...**
www.marinetalk.com/.../Cargo-**Ship-Losses**-and-Fatalities-ROY00691454IN.html - Cached - Similar

Figure 6.4: An example of a bad snippet: the document is not relevant while the COV ratio is high (0.85). This snippet was returned for the query "`ship losses`" at the 3rd position. The stated information need was: *"Identify instances in which weather was a main or contributing factor in the loss of a ship at sea"*. The snippet's summary lists relevant keywords, but the actual document does not discuss factors contributing to ship losses.

Experimental Setup: We treat this as a classification problem, where we attempt to predict the snippet label based on the textual features of the snippet, and our estimate of the snippet attractiveness, A, defined in Section 6.1.2. Specifically, we use the features listed in Table 6.1, as well as the A feature, as input to classification. We experimented with different classifiers such as Naive Bayes, Logistic Regression, SVM, and others, using 5-fold cross validation.

Results and Discussion: Interestingly, the highest accuracy was achieved by the LogitBoost [40] classifier, resulting in 97.7% accuracy (P = 97.6%, R = 97.7%, F1 = 97.6%, AUC = 93.6%). This improvement is significant over the majority baseline classifier, which had Accuracy of 90.6% (P = 82.1%, R = 90.6%, F1 = 86.2%, AUC = 43.9%). The Precision-Recall curve computed for the LogitBoost classifier is reported in Figure 6.2, showing that more than 35% of *Bad* snippets can be detected with 100% precision.

As a potential confounding factor, we did not consider whether a snippet contains an answer to the query directly on the SERP, removing the need to click on a document even when it is relevant. While this scenario could potentially violate our definition of a *Good* snippet, for the experiments in this work, this case is extremely unlikely: the search tasks, especially those from the HARD set, are relatively complex, and are not likely to be answered directly in the snippet. As another research direction, we believe that using only the shallow text features for snippet quality classification leaves significant room for improvement, for example, by incorporating the readability and language model features proposed in [78]. We plan to explore these questions further in our future work.

6.2 Motifs for Relevance Prediction and Ranking

Estimation of document relevance from large scale user behavior data is of critical importance to Web search engines. In this sections we demonstrate how automatically discovered mouse cursors motifs can be used to improve estimation of search result relevance and ranking. While earlier research showed that the result clickthrough data [75, 4] and dwell time (the time spent on the clicked result document) [32] are indicative of document relevance, their effectiveness are limited, as these techniques are agnostic about how users view the clicked documents. Recently, to address this problem, Guo and Agichtein [55] developed the Post-Click Behavior (*PCB*) model that exploit the post-click "low-level" behavioral signals, such as mouse cursor movements, which captures "reading" and "skimming" patterns that are indicative of document relevance. The *PCB* model substantially outperformed alternative approaches that only incorporated result clickthrough and dwell time information. The authors identified different patterns of viewing relevant and irrelevant documents through examining the visualizations of mouse cursor trajectories, and designed features accordingly. While the *PCB* model includes many aggregated measures of page examinations, such as cursor maximum y coordinate, distance travelled, and scrolling speed, it is not able to capture the detailed patterns of the mouse cursor trajectories. To ensure proper comparison with prior work of Huang et al. [65] and Guo et al. [55], we adhere to the same evaluation metrics: Pearson correlation coefficient and Normalized Discounted Cumulative Gain (NDCG). In the rest of the section, we describe how motif-based features can be incorporated into the existing *PCB* model and significantly improve the quality of personalized relevance prediction.

Experimental Setup and Dataset

Our test dataset in subsequent experiments, was constructed in a user study with 21 participants, following the methodology described by Feild et al. [39] and by Guo et al. [55]. The dataset contains 566 queries and corresponding 1,340 page views, including search engine result pages (SERPs), with dwell times of at least 5 seconds. Each user was asked to provide a relevance rating for the web page (on five point scale), immediately prior before navigating to the next web page. User were only asked to provide their ratings on landing pages (non-SERP). This allowed us to analyze mouse movements performed by a user on the landing page and relate them to the explicit relevance rating given by the very same user. Overall, there are 854 relevance judgments provided by user study participants. We use these relevance judgments in the relevance estimation and re-ranking experiments reported in Section 6.2.

Motif-based Relevance Prediction System

We now describe our system for predicting relevance of a document, based on both manually engineered features described previously in the *PCB* system [55], and using our automatically discovered motifs as features. The resulting system, can use either a combination of these features (*PCB+Motifs*), or either set individually (*PCB* or *Motifs*, respectively). These features are summarized below.

PCB Features

The features or predictors of the *PCB* model include the document viewing time – also known as dwell time, the characteristics of mouse cursor movements and scrolling behavior, such as ranges of mouse cursor movements for x and y, cursor movement speed, scrolling direction and frequencies, hovering certain area of interest on a web page by mouse cursor and various click statistics. These features aim to capture the searcher engagement with the examined document and viewing patterns such as "reading" and "skimming", which are shown to be indicative of document relevance [55]. The complete list of *PCB* features and model details can be found in reference [55].

Motif Features

All of the motifs discovered from the user study dataset were encoded as *features.* We only considered mouse cursor data from landing pages as relevance judgements are not defined for search result pages. Each document was then represented with a vector of features, each feature corresponding to one motif. The feature values were computed as the minimum distances between a motif and the observed mouse cursor trajectory, using the following formula:

$$\begin{aligned}\text{MinDist}(motif, \mathit{mouse}) = \\ &\min_{0<t<T-w} \text{DTW}(motif, \mathit{mouse}[t, t+w])\end{aligned} \tag{6.1}$$

where T is the mouse cursor trajectory length, w is the motif length (5000ms in our case) and $DTW(\cdot,\cdot)$ is the Dynamic Time Warping distance between the motif and sub-sequence. Therefore, the smaller the minimum distance in the formula (6.1), the higher the match between the motif and the mouse trajectory (and corresponding feature value). In fact, if the motif perfectly matches any subsequence of a page view, $MinDist$ is equal to zero. This is analogous to a bag-of-words document representation that is widely used in information retrieval, except that in our case the "words" represent the common mouse cursor movements represented as motifs that occur in a page examination time series (analogous to a “document”), and the value interpretations are reversed (lower is better).

Evaluating Prediction Quality

We now report the performance of the different systems on predicting document relevance. As evaluation metric, we use the Pearson correlation coefficient ρ, between the predicted and the true relevance labels, defined as:

$$\rho_{f,y} = \frac{\sum_{(x,y)\in D}(f(x) - \mu_f)(y - \mu_y)}{(|D| - 1)\sigma_f \sigma_y}$$

where μ is the observed sample mean and σ is the observed sample standard deviation. Pearson correlation is the evaluation metric of choice in previous work [55, 65], thus appropriate for comparison to previously proposed methods, namely **PCB** and cursor hover [65].

Predicting Relevance

We formulate the prediction problem as regression, and conduct 10-fold cross validation. The regression algorithm we used is Ridge Linear Regression, which is a variant of Multiple Linear Regression. Ridge linear regression is reported to be more robust to predictor collinearity and overfitting.

Regression Results

Table 6.3 summarizes performance of four regression models with distinguished with different feature subsets. As we can see, the best performing model results from combining the automatically extracted motifs and the PCB predictors (*PCB + motifs*), achieving the correlation of 0.468 between the actual personally judged relevance and the estimated relevance. Our model improved the prediction effectiveness over the state-of-the-art PCB model by over 19%. The predictions using the automatically extracted motifs alone (*motifs*) correlated with the actual relevance judgments at 0.394, which is comparable with the PCB model. This demonstrates that the our approach indeed enables discovery of valuable patterns that are not easily identifiable through manual effort. To compare with prior work of Huang et al. [65], we calculated Pearson’s correlation between cursor hover rate and explicit label provided by the participants. We have not found any substantial correlation between hover rate and relevance labels measured for all participants. However, correlation coefficient calculated for each participant separately, varies from -0.23 to 0.27 which explains negligible small correlation across the users. In our comparison we report average absolute value for the all participants, thus giving an advantage to the hover rate. Nevertheless, it provides very little information with correlation and is outperformed by PCB models with a substantial margin.

Feature Group	**Pearson correlation** [1]	**p-value**
Hover [65]	0.12	$p < 0.001$
PCB [55]	0.392 (n/a)	$p < 0.001$
motifs	0.394 (+0.5%)	$p < 0.001$
PCB + motifs	**0.468 (+19.4%)**	$p < 0.001$

Table 6.3: Pearson's correlation between the actual personally judged document relevance ratings and the predicted relevance ratings for the PCB baseline model and the automatically extracted mouse motifs.

Search Result Ranking

We now turn to the other practical application of motif discovery, result ranking. As in the relevance prediction experiments, we compare four models: hover [65], motifs, *PCB* and combined *PCB* and motifs models. For consistency, we use the same Linear Ridge Regression classifier as for the experiments above.

We evaluate the quality of produced rankings with Normalized Discounted Cumulative Gain (NDCG), at different cut-off positions. The data us stratified by user, through holding out all training examples for a user from the training set and using them to test ranking performance. The same motifs were used as before, as they were discovered on a disjoint, unlabeled dataset described in section 6.2. We repeat the training and testing procedure for each of 21 users in our dataset and report NDCG@k averaged across all the users, resulting in a leave-one-out form of cross validation.

Normalized Discounted Cumulative Gain at K (NDCG@k)

is a standard metric for assessing quality of search results ranking. The metric is parametrized by cutoff position k, that is, to calculate NDCG@k we consider only top k results. NDCG@k is given by

$$NDCG@k = \frac{DCG@k}{IDCG@k}, DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

where $IDCG@k$ is the $DCG@k$ value of the ideal document ranking according to the relevance labels, and rel_i is relevance score for a i-th document. $DCG@k$ penalizes ranking with relevant documents appearing at lower position in the list with logarithmic discount appearing the denominator in the formula. The $NDCG@k$ value of 1.0 indicates a perfect ranking. In our ranking experiment we perform cross validation by user - for each run we hold out documents seen by a single user and train the model on the rest of the data. At test stage we compute relevance scores for the held out documents and calculate $NDCG@k$ based on the actual relevance labels. After doing it for each user, we report $NDCG@k$ averaged across all users in our dataset.

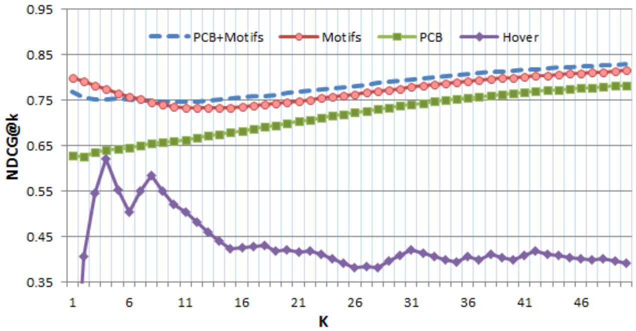


Figure 6.5: Comparison of result rankings based on Cursor Hover, PCB nd motifs models.

Ranking Results

Figure 6.5 reports $NDCG@k$ for the values of k ranging from 1 to 50. Interestingly, despite similar performance measured by Pearson correlation, ranking based on motif features behaves quite differently from the ranking based on the PCB model. Specifically, the motif model has significantly higher NDCG values at positions 1 through 10 (with the improvement on NDCG@1 over the original PCB model of 27%, p-value < 0.001). However, this improvement decreases with higher ranks k. The combined model PCB+motifs outperforms both PCB and motifs models at lower rank positions. The relative improvement of the combined model over the PCB are: $NDCG$@10 - 12.9%, $NDCG$@20 - 9.4%, $NDCG$@30 - 7.4%, $NDCG$@40 - 6.5% and $NDCG$@50 - 5.9%. Finally, the ranking based on cursor hover rates performs consistently poorly, especially at lower ranks.

6.3 Attention Biased Document Summarization

In this section we describe our approach of using mouse cursor data for attention biased generation of search result summaries (often called snippets). Throughout the section we use behavior or behavioral data in the reference to mouse cursor data that we use to approximate user attention in this task. First, we formalize the problem of generating "useful" snippets. Then, we describe the key parts of our approach (Section 6.3.2), and the infrastructure we developed to accomplish the required data collection (Section 6.3.3).

6.3.1 Problem Statement

Following the literature on snippet quality [96], good snippets must satisfy the aspects of *Representativeness*, *Readability*, and *Judgeability*:

1. *Representativeness*: measures how well the snippet summarizes parts of the web page relevant

to the search query. A representative snippet would clearly show why the page was found by a search engine in response to the query.

2. *Readability* measures the ease with which the text of the snippet can be read and understood. Note, that readability does not depend on the search query [79].

3. *Judgeability* measures how well the snippet helps a user to understand whether the page is helpful for the specific search intent, and to decide whether to click on a link or not. An ideal snippet would either contain an answer to a user's information need (or a clear indication that an answer is present in the document), or else clearly show that the page is not relevant .

Our primary goal is to optimize the Representativeness and the Judgeability criteria by *biasing* the selected snippets towards the regions of most interest to the user, as inferred from the page examination data. That is, our goal is not to replace the existing text-based snippet generation approaches, but rather to add additional evidence (when available) about the parts of the document to privilege.

6.3.2 Approach

Our approach operationalizes the snippet quality criteria above by incorporating both textual and behavioral evidence using a robust machine learning-based approach. Specifically, we combine together the traditional text-based snippet generation features, and the inferred user interest in specific parts of a document.

First, following [79], a fragment scoring system is trained based on text-based features, using human judges, resulting in a strong text-only baseline that generates *candidate fragments* to be included into the snippet (System 6.3.4). Separately, examination behavior data is collected over the landing pages, using our logging infrastructure described in the next section. Then, a behavior model is trained to infer the document fragments of interest to the user, based on user examination data (Section 6.3.4). Finally, the behavior-based prediction of interest in each candidate fragment is combined with the original (text-based) fragment score, in order to generate the final *behavior-biased* snippet candidate ranking (Section 6.3.4). Note that by decoupling the behavior modeling from the candidate generation method, our approach can be used with any other snippet generation approach that provides scores for the candidate fragments, which could be combined with the behavior scores for the final ranking step.

While general and flexible, our approach makes three key assumptions. First, our method is primarily targeted (and evaluated for) informational queries – that is, queries for which the user expects to find an answer in the text of the page, and optimizes the snippets accordingly. Second, we assume that document visits can be grouped by query intent, so that behavior features on the landing pages can be aggregated together for all the searchers with the same information need.

While a number of methods have been proposed to cluster queries (and results clicks) by intent (e.g., [110]), we acknowledge that these techniques are not perfect, and may introduce noise in practice. Finally, we assume that user interactions on landing pages can be collected by a search engine or a third party. While this naturally introduces potential privacy concerns, this assumption is not far-fetched: already, browser plug-ins and toolbars collect user interactions on web pages; major organizations can (and often do) use proxies for external web access; and common page widgets like banner adds and visit counters inject JavaScript code to monitor basic user interactions and can be easily extended to collect more detailed data. While the privacy and security considerations of these methods are beyond the scope of this work, we merely point out that these behavior gathering tools already exist and are widely deployed.

6.3.3 Page Examination Behavior Logging

A key component of our system is a mechanism for collecting searcher interactions on web pages, and tying them precisely to the page content at the word level. As a starting point, we adapt the publicly available EMU toolbar for the Firefox browser [50], that is able to collect mouse cursor movements over any visited webpage. Unfortunately, out-of-the-box EMU functionality is not sufficient, as the user interactions are not connected to the underlying page content: the available JavaScript API does not provide the text position under the cursor, which could depend on screen resolution, size of browser window, browser version, and personal browser settings.

To associate the tracked mouse cursor positions with corresponding text fragments we employed the following technique. After the HTML page is rendered in the browser window, our JavaScript code modifies the document DOM tree ,so that each word is wrapped by a separate DOM element tags. Then for each DOM Element, the window coordinates of that element are evaluated and saved in the Element's attributes. Then, the processed HTML page with the coordinates of each DOM Element is saved to the server by an asynchronous request. The saved coordinates are updated if the page layout is changed due to a *resize* window event or an AJAX action.

Thus, for each page visit we know the searcher's intent (question), the search engine query that the user issued, the URL, the contents of the document, the bounding boxes of each word in the HTML text, and the log of behavior actions: mouse cursor coordinates, mouse clicks, and scrolling, and an answer to the question that the user found in a page and submitted in the game interface. Next, we show how to use this information to infer patterns of browsing behavior that capture portions of document that are of most interest to the user.

6.3.4 Behavior-Biased Snippet Generation

We now present the details of our Behavior-Biased snippet generation system (BeBS). First, we describe the text-only snippet generation system (Sections 6.3.4 and 6.3.4). Then, we introduce the method for inferring the most interesting or useful parts of the document from user behavior

Feature	Description	Feature Group
ExactMatch	1 if fragment contains query as a substring, otherwise 0	Metzler - Kanungo
TermOverlap	Overlap of query terms and the fragment	
SynOverlap	Overlap of query terms expanded with synonyms and the fragment	
LanguageModelScore	Fragment score under the language model as in [101]	
Length	Total number of terms	
Location	Relative location of the fragment in the document	
BM25ScoreFragment	BM25 score of the fragment	Relevance
BM25ScoreSentence	BM25 score of the sentence from which fragment was extracted	
BM25ScorePerWord	BM25 of the fragment divided on number of words in the fragment	
NumMatches	absolute count of query terms matched in the fragment	Query Match
SentenceBegDistance	Number of words between beginning of the sentence and first word in the fragment	
SentenceEndDistance	Number of words between end of the sentence and last word in the fragment	
QueryTermDistanceAvg	Average distance of query terms in the fragment measured in words	
QueryTermDistanceMin	Minimum distance of query terms in the fragment measured in words	
QueryTermDistanceMax	Maximum distance of query terms in the fragment measured in words	
NumDistinctTerms	Number of distinct terms in the fragment	Readability
NumPunctChar	Number of punctuation characters	
PercentPunctChar	Percent of punctuation characters	
NumLetterChar	Number of letter ([a-zA-z]) characters	
NumWordsCap	Number of words with first letter capitalized	
PercentWordsCap	Percent of words with first letter capitalized	
PunctPerWord	Number of punctuation characters per word in the fragment	

Table 6.4: Text-based features for text fragments

(Section 6.3.4), to incorporate into the combined snippet generation process (Section 6.3.4).

Text-Based Snippet Generation

In order to generate snippets, we extend the approach presented in Metzler and Kanungo [101]. The downloaded HTML pages are pre-processed and indexed with Natural Language Tool Kit (NLTK [15]). Extracted text is divided into sentences using *Punk* unsupervised sentence splitter [84]. We index the text of the web page excluding the $\langle script \rangle >$ and $\langle style \rangle >$ tags.

For a given query we first select all the sentences that have at least one match of query terms for further snippet fragment generation. Once the set of sentences selected, our system generates all possible snippet fragment candidates by applying a sliding window moving along each sentence. We vary fragment length from 3 words to a maximum character length provided as an input parameter. We discard all fragments that do not contain any query term matches. Along with fragment generation our system scores each fragment using $TextScore$ function described in Section 6.3.4. This score is used to generate the final snippet.

The problem of summary generation has been studied extensively in natural language processing and summarization research communities. It has been shown [122] that it is equivalent to a weighted set cover problem which is, in turn, known to be $NP-Hard$. There are several possible approaches available for this problem, including greedy weighted set cover and relaxations, primarily based on integer linear programming. We resort to a greedy algorithm due to relative simplicity of implementation. Our system can easily be extended with more advanced set cover solver if needed. As the set cover algorithms requires score computation for set of selected fragments, we recompute the scoring function for the union of selected candidate fragments to find a set that greedily maximizes the score for entire snippet.

Fragment Scoring

The fragment scoring required for snippet generation relies on a machine learning approach based on set of text features representing various quality aspects of fragment candidate. We extend the method of [101] by adding additional features capturing relevance of the fragment (relevance group), properties of query match (query match group) and readability of the fragment (readability group). These features are summarized in Table 6.4. For $TextScore$ score computation we used the Gradient Boosting Regression Tree model [42] (GBRT). GBRT is a powerful family of models that has been successfully used in many applications including sentence selection for search result summarization [101] and search result snippet readability assessment [79]. We train a GBRT model on a subset of training query-URL pairs to predict snippet fragment scores.

Gradient Boosting Regression Tree GBRT performs a numerical optimization in function space instead of parameter space. We provide a brief overview of the algorithm and refer to the original paper for detailed information [42]. A regression tree model $f(x), x \in R^n$, partitions the space of covariates into disjoint intervals $R_k, k = 1, 2, ..., K$ associated with leaf nodes of the tree. Each interval is assigned a value ϕ_k, such that $f(x) = \phi_k$ if $x \in R_k$. Thus, the tree model can be written in

$$T(x;\Theta) = \sum_{j=k}^{K} \phi_k I(x \in R_k)$$

where $\Theta = \{R_k, \phi_k\}_{k=1}^{K}$, and I is an indicator function. For a given loss function $L(y_i, \phi_i)$ the parameters Θ are result of the following optimization problem:

$$\hat{\Theta} = argmin_{\Theta} \sum_{k=1}^{K} \sum_{x_i \in R_k} L(y_i, \phi_k)$$

In our experiments we employ the squared loss function to train the regression trees. A gradient boosted regression tree is an ensemble model[42] that incorporates a series of regression trees, and can be written as:

$$f_M(x) = \sum_{m=1}^{M} T(x;\Theta_m)$$

where at each stage m, Θ_m is estimated to fit the residuals from the $m-1$th stage:

$$\hat{\Theta}_m = argmin_{\Theta_m} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \phi_{k_m})$$

and M is the number of stages(regression trees) in the model. In practice, one adds $T(x;\Theta_m)$ multiplied by ρ - the learning rate input parameter specified for the algorithm, resulting in the final predictor:

$$f_M(x) = \sum_{m=1}^{M} \rho T(x;\Theta_m)$$

In our implementation we used $M = 200$ regression trees, and $\rho = 0.01$ to train the GBRT model for fragment scoring.

Inferring Relevant Text Fragments from Search Behavior

To infer text fragment importance from user's browsing behavior, supervised machine learning is applied. For each page visit of a user, the visited HTML document is represented as a set of short text *fragments*. A fragment is labeled as "interesting" (attractive) for the user if the user submitted an answer in the current session, and the answer has common words with the fragment. Other fragments are labeled as not interesting. The answer text and the fragment are compared after stemming and stopword removal.

For each fragment a set of behavior features that could represent fragment interestingness is created. The key feature is a duration of time interval when a mouse cursor was over the specific

Feature	Description
MouseOverTime	Time duration when the mouse cursor was over the text fragment
MouseNearTime	Time duration when the mouse cursor was close to the text fragment in the window ($x \pm 100px$, $y \pm 70px$)
MouseOverEvents	The number of mouse events during *MouseOverTime*
MouseNearEvents	The number of mouse events during *MouseNearTime*
DisplayTime	Time duration when the text fragment has been visible in the browser window (depends on scrollbar position)
DispMiddleTime	Time duration when the text fragment was visible in the the middle part of the browser window

Table 6.5: Behavior features for text fragments

text fragment, or very close to the fragment. We also adapt the features to measure scrollbar and event activity from references [25], and [56] to detect “reading” vs. “skimming behavior, and adapt those features to represent behavior near a specific page coordinate. The complete list of the fragment behavior features are presented in Table 6.5.

The feature generation algorithm joins a sequence of behavior events and a set of bounding boxes for each word and DOM Element of a page. The algorithm builds a spatial R-Tree index of element bounding boxes, which allows for each event to efficiently find matching DOM Elements in a specified coordinate range.

We used Gradient Boosting Regression Tree algorithm [42] to predict the probability that a fragment is interesting for a user. The set of page visits is divided into training and test set, so that the training and test set URLs are disjoint. The training set is created from only those page visits where the document text has a nonempty intersection with the user's answer, and the answer is correct. The trained regression algorithm is applied to all page visits in the test set. When the algorithm is applied to the test set, it has no information about user's intent, answer, and current query, and uses only behavioral features of the current page visit. The predicted probability of fragment's interestingness is then used as a feature for the snippet generation algorithm.

Combining Text and Examination Evidence

The final step in our approach is to *combine* the text-based score $TextScore(f)$ for a candidate fragment (Section 6.3.4) with the interestingness score $BScore(f)$ (Section 6.3.4), inferred from the examination data. In our current implementation we combine these scores by linear combination:

$$FScore(f) = \lambda \cdot BScore(f) + (1 - \lambda) \cdot TextScore(f)$$

Note that while the $TextScore(f)$ is not normalized, and could have values in range $[1, 5]$, the $BScore(f)$ is normalized to interval $[0, 1]$.

The parameter λ affects two characteristics of the algorithm: snippet *coverage* and *quality*. Snippet *coverage* is defined as the ratio of snippets for which the snippets produced by a baseline algorithm are not equal to the snippets produced by the algorithm with behavior features. Snippet *quality* is measured by judgeability, readability, and representativeness metrics, by manual pairwise assessments. As λ approaches zero, coverage also approaches zero (as text-based features dominate candidate selection), and the algorithm effectively backs off to the baseline. In contrast, when λ is large, *quality* might decrease by weighing behavior features too highly. We performed manual assessments for five different parameter values of $\lambda \in [0, 1]$ to select the best value. Other more sophisticated ways to combine text and behavior evidence are possible, such as jointly learning over both text and behavior features. However, we chose to follow the simpler linear approach for interpretability of the results (e.g., by varying the λ parameter directly).

6.3.5 Data Collection and Experimental Setup

This section presents the methodology used for acquiring search behavior data for training our system (Section 6.3.5), describes the resulting behavioral data (Section 6.3.5), and the explicit snippet judgments dataset required for training and evaluation (Section 6.3.5).

Acquiring Search Behavior Data

To collect the search behavior data, we used the infrastructure created and published by [1], and modified it for our task. The participants played a search contest "game" consisting of 12 search tasks (questions) to solve. The stated goal of the game was to submit the highest possible number of correct answers within the allotted time. After the searcher decided that they found the answer, they were instructed to type the answer together with the supporting URL, into the corresponding fields in the game interface. Each search session (for one question) was completed by either submitting an answer or clicking the "skip question" button to pass to the next question.

Participants were recruited through the Amazon Mechanical Turk (MTurk) website. As a first step, the workers had to solve a ReCaptcha puzzle to verify that they are human and not an automated "bot". A browser verification check was performed to verify that the browser is compatible with our JavaScript tracking code. During the data postprocessing stage, we filtered out the users who did not answer even the easy, trivial questions, as it indicated either poor understanding of the game rules, or an attempt to make a quick buck without effort.

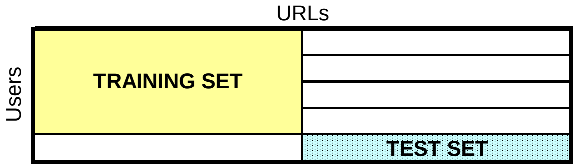


Figure 6.6: Illustration of the cross-validation set up: the training and test sets are disjoint by both users and URLs.

To capture all of the participants' search actions, they were instructed to use only our search interface. Our search interface performs web search using the public API of a popular web search engine, and shows the result pages using the original page design, layout and stylesheets, so the user's search experience is not affected. The Apache Web server proxy functionality was used by configuring the modules *mod_proxy*, *mod_proxy_html*, and *mod_sed* so that the users could search and browse the Web in a usual way, while the URLs in the html links were automatically replaced to request the URL through our proxy. As the requested documents were returned through our proxy, JavaScript code was embedded to track the user's interactions, including mouse movements and scrolling, as well as the properties of the visited page. The interaction events were logged by our proxy and written to the log.

Browsing Behavior Dataset

A total of 109 MTurk participants finished the game. After filtering out users who did not follow the game rules, we obtained 1175 search sessions performed by 98 users. Our data for these users consists of 3294 queries, 1598 unique queries, 2997 SERP clicks on 662 distinct URLs. For 2289 page visits (76%) and 508 distinct URLs, the document behavioral data is collected. For the rest 24% of page visits, the behavioral data were not collected due to conflicts between our JavaScript tracking code and other code presented on the page. For each page view there were on average 400 atomic browsing events (mouse movements, scrolling, key pressing) on average.

The set of URLs with collected behavior data was divided randomly into equal-sized training and test set. The training set was used to train the regression algorithm for predicting fragment interestingness, and the test set was used to assign interestingness score to fragments and generate snippets. The test set consists of $508/2 = 254$ different URLs, and for each of them there is a collected browsing behavior. Each URL might be visited from different queries, and for each query-URL pair a snippet generation algorithm produced a snippet. So the comparative experiments for snippet quality evaluation were performed on a set of 707 different query-URL pairs.

Fragment Quality Data Collection

We collected 949 fragment quality judgments through the Amazon MTurk service. The assessors were asked to re-rank 10 text fragments randomly chosen by our fragment generator system to obtain a reliable training set. Each fragment was judged by 3 assessors. The fragments used in for this data collection were generated from query-URL instances taken from our training set and do not overlap with our test set. We specifically asked assessors to re-rank fragments to avoid inaccuracies caused by using absolute scale. In order to train the fragment scorer we performed rank aggregation by computing average rank of the fragment among rankings produced by different judges.

6.3.6 Results

We now present the empirical results. First, we report the intermediate result of using behavior data to infer the interesting (useful) fragments in the document. Then, we report the main results where the quality of the generated snippets with and without using behavior data is compared using human judgments (Section 6.3.6).

Prediction of Fragment Interestingness

This experiment evaluates how well we can predict interesting fragments by observing the user's document examination behavior. We define the fragment to be interesting if it is related to the answer for the question. For each visited page we collect the user's answer (if submitted), and all the correct answers from all the users who answered this question. Then we automatically compare those answers to each text fragment in the document.

For this experiment, the document text is represented as a set of short text fragments, each consisting of five words. Those 5-word sequences are, on one hand, almost unique in a typical web document, and thus could be used as an identifier of a text position in a document, and on the other hand are short enough to match to local behavior patterns. For each fragment a set of behavior features is computed as described in section 6.3.4.

The cross-validation experiment set up as illustrated in Figure 6.6. The set of users and URLs are divided into a training set and a test set, so that the training set and the test set are disjoint for both sets of users and URLs. 10-fold cross-validation was performed, so that each user-URL pair appeared in a test set for some cross-validation split.

In the training set, a fragment's label is set to $label\left(fragment_i\right) = 1$ if the user submitted an answer in the current session, the answer is correct, and the answer has common words with the fragment. If the user submitted a correct answer, but the answer shares no words with a document fragment, then $label\left(fragment_i\right) = 0$. If the user did not submit an answer, or the answer is incorrect, we excluded the fragment from the training set. The answer and the fragment were compared after stemming and stopword removal. Similarly, in the test set we use for evaluation only those fragments that share words with the submitted search query

The Gradient Boosting Regression Tree algorithm was trained on the training set of fragments, and applied to the test set. So each fragment in the test set receives a *fragment interestingness*

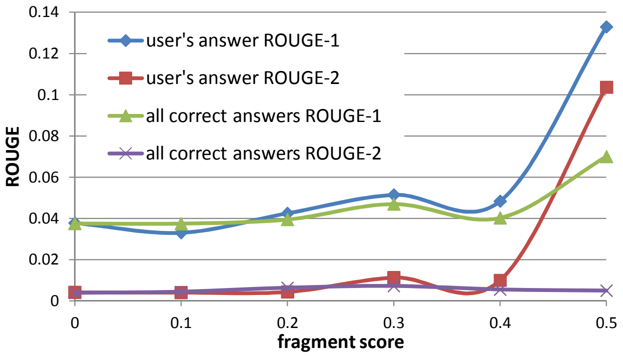


Figure 6.7: Intersection of fragments with submitted answer vs. fragment interestingness score $BScore$ predicted by behavior

Feature Group	NDCG@10
Single Feature Group	
Metzler-Kanungo	0.719
Relevance	0.741
Readability	**0.743**
QueryMatch	0.719
All Except One Feature Group	
All-Metzler-Kanungo	0.725
All-Relevance	0.736
All-Readability	**0.743**
All-QueryMatch	0.717
All	**0.764**

Table 6.6: Feature ablation results for fragment text scoring (10-fold cross validation)

$BScore(fragment_i) \in \mathbb{R}$.

We evaluate the interestingness of fragments by comparing the fragment's text with user's answer (if it exists), and to all the correct answers submitted by all users for the same question. We use the standard ROUGE-1 and ROUGE-2 metrics[97] for evaluation of the fragment intersection with answers, as these metrics are commonly used for evaluation of automatic summarization and annotation algorithms.

ROUGE-N metric for a fragment and a set of answers A, is computed as the recall of the answer set covered by the fragment word N-grams:

$$\text{ROUGE-N}\,(fragment, A) = \frac{\sum\limits_{a \in A} \sum\limits_{gram_n \in fragment} \text{Count}_{match}(gram_n)}{\sum\limits_{a \in A} \sum\limits_{gram_n \in fragment} \text{Count}(gram_n)}$$

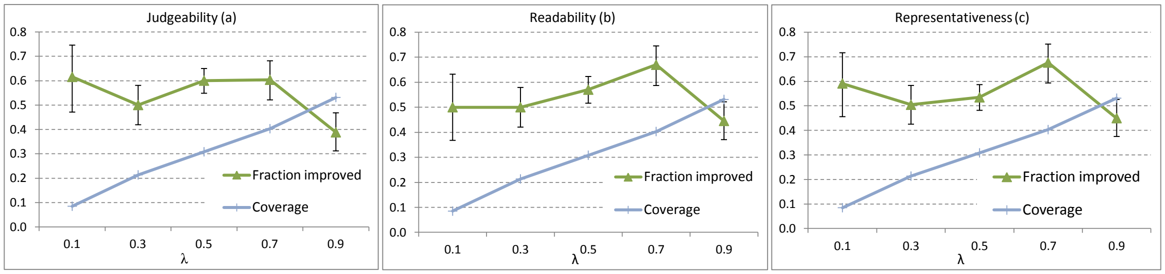


Figure 6.8: Snippet quality vs. coverage for different behavior weight λ

Figure 6.7 shows the relationship between the interestingness of a fragment and the behavior score. The graph shows that when the score is high (≥ 0.5), the average intersection between the fragment and user's answer is much higher than those when the fragment score is low. All ROUGE-N metrics increase when the behavior score increases, but the ROUGE-2 values over all correct answers are always very small (changing from 0.003 to 0.007). We note that ROUGE-1 is much greater than ROUGE-2 for high scores, as the interesting fragment might contain useful information for the answer, but the user reformulates the obtained information and submits reformulated answer. The ROUGE-N values for a user's answer are much greater than those for all correct answers, as other users might obtain valuable information from other documents, and some questions have distinct correct answers.

This experiment shows that we can predict fragment interestingness by using behavior features only. In the next section, we apply the computed behavior scores for a practical task of improving search result summaries.

Snippet Quality Evaluation

Evaluation Setup: Our evaluation follows the snippet quality desiderata outlined in Section 6.3.1. Specifically, the snippet quality is evaluated by performing blind paired preference tests. For each query and URL, a pair of snippets produced by two different algorithms were evaluated by an assessor. A pair of snippets were presented on a page in random order, so the assessors did not know which algorithm produced which snippet.

Each assessor was asked to examine the web page, the search query, and the pair of snippets, and to answer three questions that correspond to our snippet quality criteria:

- Which of the snippets better summarize the parts the web page relevant to the search query? You need to read the web page before answering this question.

- Which of the snippets is written better – and is more readable?

- Imagine that you have the following search intent: "*question*". Which snippet helps you to identify relevant content better, and helps you decide whether to click on this result or not?

Baseline vs. behavior with $\lambda = 0.7$			
	Judg.	Read.	Repr.
baseline is better	44	35	34
similar	23	27	29
behavior is better	67	71	71
no answer	0	1	0
ratio of improved	0.60*	0.67*	0.68*
p-value	0.018	0.0003	0.0002

Baseline vs. behavior with $\lambda = 0.5$			
	Judg.	Read.	Repr.
baseline is better	108	107	122
similar	160	149	207
behavior is better	162	142	140
no answer	41	73	2
ratio of improved	0.60*	0.57*	0.53
p-value	0.0006	0.0015	0.14

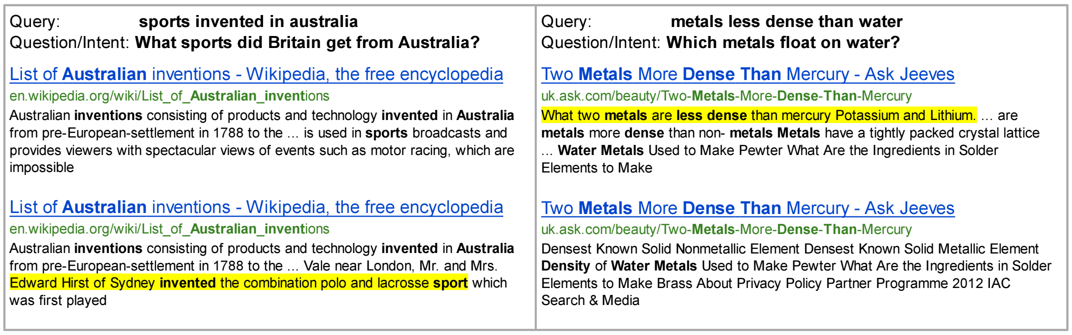
Table 6.7: Pairwise preference tests for snippets with behavior features added, number of judgements

You must consult the list of the correct answers for this question before answering.

For each question there were three possible answers: ”Snippet 1 is better”, ”Both snippets are similar for this criterion”, and ”Snippet 2 is better”.

We hired 14 pre-qualified Amazon MTurk workers, who have previously shown accurate results and high agreement with our “gold standard” subset of labeled snippets by the authors. As an additional test, we also included a small portion of exactly the same snippets to check the quality of MTurk workers, to verify that for the *same* snippets the worker submitted ”Both snippets are similar” label for each criterion. As a result, we collected pairwise preference labels for 2959 snippet pairs, 8525 atomic judgements (three judgements for each snippet pair, excluding ”no answer” responses). The total cost of MTurk workers was $217. One half of the obtained judgements were used for development and debugging purposes, and the other half were used for testing and reporting the results in the next section.

Evaluation Metrics: As a main evaluation metric we use the fraction of labels that give *preference* to the BeBS system, compared to the baseline. The preference ratio metric is evaluated for each criterion in (judgeability, readability, representativeness). The reason is that we found that the snippet quality criteria are difficult for assessors to judge absolutely, but can be easily compared as preferences.



Query: **sports invented in australia** Question/Intent: **What sports did Britain get from Australia?**	Query: **metals less dense than water** Question/Intent: **Which metals float on water?**
List of **Australian** inventions - Wikipedia, the free encyclopedia en.wikipedia.org/wiki/List_of_**Australian_invent**ions Australian **inventions** consisting of products and technology **invented** in **Australia** from pre-European-settlement in 1788 to the ... is used in **sports** broadcasts and provides viewers with spectacular views of events such as motor racing, which are impossible	Two **Metals** More **Dense Than** Mercury - Ask Jeeves uk.ask.com/beauty/Two-**Metals**-More-**Dense**-**Than**-Mercury What two **metals** are **less dense** than mercury Potassium and Lithium. ... are **metals** more **dense** than non- **metals Metals** have a tightly packed crystal lattice ... **Water Metals** Used to Make Pewter What Are the Ingredients in Solder Elements to Make
List of **Australian** inventions - Wikipedia, the free encyclopedia en.wikipedia.org/wiki/List_of_**Australian_invent**ions Australian **inventions** consisting of products and technology **invented** in **Australia** from pre-European-settlement in 1788 to the ... Vale near London, Mr. and Mrs. Edward Hirst of Sydney **invented** the combination polo and lacrosse **sport** which was first played	Two **Metals** More **Dense Than** Mercury - Ask Jeeves uk.ask.com/beauty/Two-**Metals**-More-**Dense**-**Than**-Mercury Densest Known Solid Nonmetallic Element Densest Known Solid Metallic Element **Density** of **Water Metals** Used to Make Pewter What Are the Ingredients in Solder Elements to Make Brass About Privacy Policy Partner Programme 2012 IAC Search & Media

Figure 6.9: Comparison of snippets produced by a baseline algorithm (top), and by using behavior features (bottom) for two different queries. The relevant snippet parts are highlighted in yellow.

Analysis of the Text-based System

Before studying the benefit of behavior-based improvements, we optimized the text-based baseline method (Section 6.3.4 by varying the combination of features used. The results of the feature ablation experiments are reported in Table 6.6. As the table shows, using *all* of the text-based features achieves highest judge preference for the resulting snippets, thus, we use all of the features described in Table 6.4 for the subsequent experiments.

Evaluation of the BeBS System

This experiment compares our baseline algorithm described in section 6.3.4 with the BeBS algorithm that combines behavior features to fragment score using the λ parameter for the relative weight of the behavior- and text-based scores. Recall, that λ affects two characteristics of the algorithm: coverage and snippet quality. Figure 6.8 reports the judgeability, readability, representativeness, and coverage for five values of λ. The binomial distribution two-sided statistical significance test with confidence level 0.9 was computed , and the corresponding confidence intervals are presented on the graph. The graph shows that the behavior features with $\lambda = 0.7$ provide significant improvement on all three snippet quality metrics. For this value, coverage equals 40%, which means that the behavior features provide improvement in representativeness for $0.4 * 0.68 = 27\%$ of all snippets, and produce worse snippets for $0.4 * (1 - 0.68) = 13\%$ of all snippets.

When $\lambda = 0.5$, judgeability and readability also improve, but the improvement in representativeness is small and not statistically significant. When $\lambda = 0.9$, coverage grows up to 53%, but results in more noise and degrades snippet quality. When λ is set to a low value, the coverage drops, and we have too few modified snippets from the baseline to observe statistically significant differences in snippet quality. The graph shows that for $\lambda \in \{0.1, 0.3\}$, the confidence intervals cross $y = 0.5$ axis, and this means that the difference in snippet quality is not statistically significant. The Table 6.7 reports the detailed assessment data for the two best runs.

Feature	Gini coefficient
DispMiddleTime	0.51
MouseOverTime	0.34
DisplayTime	0.12
MouseNearTime	0.02
MouseOverEvents	0.01
MouseNearEvents	0.01

Table 6.8: Feature importance for behavioral features

Finally, we show two examples that demonstrate how behavior features affect snippets. The first example (Figure 6.9, left) shows two snippets produced by a baseline algorithm (top), and by using the behavior features (bottom) for the search query "sports invented in australia" issued in the session with intent to find an answer for a question "What sports did Britain get from Australia?". The example shows that the bottom snippet includes an answer to the question into the snippet text: "`Polocrosse was invented in Australia, and this sport is indeed popular in England`". The landing page contains 8 matches for the query word "sport", and more than 50 matches for each of words "invent", and "Australia". The relevant fragment is located near the bottom of the document (three scrolling screens down), but it is included into the snippet because several users who read that landing page scrolled to that position, and inspected the text near this fragment thoroughly, resulting in a behavioral score.

The second example shows a case when the algorithm with behavior features produced worse results compared to the baseline. The user issued a query "metals less dense than water", and the baseline algorithm produced a good snippet that contain an answer "Potassium and Lithium", that is relevant to both search query and search intent. But some users did not found the answer on the landing page, and instead examined the attractive section of the page corresponding to "Popular Searches", with the list of suggested queries. That resulted in a high behavior scores for that fragment, and consequently BeBS produced a snippet with poorly readable and irrelevant text. Additional behavior data, and further tuning of the behavior score prediction may improve these situations, as we plan to explore in future work.

Behavior Feature Importance Analysis

To estimate relative importance of behavior features for snippet generation, we analyzed the Gini importance index [19] for each behavior feature from the table 6.5. The table 6.8 shows that the most important features are *DispMiddleTime* - the time duration when the text fragment was visible in the middle of the browser window, and *MouseOverTime*, the duration of the mouse cursor was hovering over the text fragment. While the first feature has been previously shown [25] to be beneficial for re-ranking search results, we are encouraged to find it to be also beneficial for snippet generation. The *MouseOverTime* feature has been shown to be correlated with user interest [54], thus confirming our hypothesis that searcher interest can be used to generate better snippets.

6.4 Summary

This chapter demonstrates several practical applications of the ViewSer technique and presents applications of mouse cursor data for prediction of document relevance and attention biased document summarization. Results obtained in each of the applications show that mouse cursor data enables significant improvements in important web search problems.

Chapter 7

Applications of Attention Tracking in Medical Domain

In this chapter we present important applications of attention tracking to the behavioral testing. In addition to the eye tracking version of the Visual Paired Comparison (VPC) task, we describe Web based version of the task, which we call VPCW. The web based version of the VPC task does not require eye tracking and relies on restricted focus viewing. We show that VPCW is sensitive enough to measure human's preference for novel stimuli when administered to health control subjects, while showing significantly lower novelty preference, when administered to subjects with memory impairment. Besides being cost effective, compared to the eye tracking version of the test, VPCW has a potential to reach much wider population of subjects within and outside the clinic.

7.1 Background and Motivation

Alzheimer's disease currently affects over 5.2 million Americans, with marked increases in prevalence expected over the next several decades due to the growing elderly population. A critical goal of Alzheimers disease (AD) research is to improve methods for early diagnosis, and to identify those at highest risk, because the best chances for effective treatment, and ultimately prevention, depend upon starting treatment before significant neurodegeneration has occurred. Early detection of AD using non-invasive methods, such as eye tracking, could play a major role in providing treatment that may slow down the disease progression rate. Although there has been substantial progress in developing genetic, imaging and cerebrospinal fluid biomarkers for AD [109] much of the current work is aimed at detecting presence of the disease using invasive or imaging methods. In contrast, this work builds on a recently introduced approach of Crutcher et al. [33, 129] that finds behavioral changes in the way subjects examine visual stimuli presented to the during the Visual Paired Comparison (VPC) task.

The Visual Paired Comparison (VPC) task has been used to test visual recognition memory in infants, adults, rodents, and monkeys [9, 106, 21, 130, 20]. The task doesnt require specific training, but relies on the subjects innate preference for novelty. Each trial in the VPC task consists of two phases. During the familiarization phase, subjects are presented with two identical visual stimuli, usually simple images (clipart back and white images approximately half of which namable, other is abstract unnameable), side by side on a computer screen. Subjects are shown the images for a specified amount of time. After a delay, the test phase occurs, and the subject is again shown two side by side images, where one is the familiar image from the familiarization phase and the other is a new, novel, image. Throughout the task, eye movements are monitored using infrared eye tracking equipment to measure gaze locations. Multiple trials are given, using different stimuli for each trial. As established in previous studies [9, 106, 21, 130, 20], during the test phase, normal control subjects typically spend a greater proportion of time looking at the novel image. This indicates that they remember the repeated, and now less interesting, image. By contrast, subjects with impaired memory do not exhibit a novelty preference, but instead view both the familiar and novel stimuli about equally.

Studies in monkeys have shown that this task is sensitive to even very minimal damage to brain structures in the medial temporal lobe (MTL, [130]). In humans, the memory impairment associated with aMCI has also been linked to structural changes beginning in the MTL ([18]). Previously, research by Crutcher et al. demonstrated that aMCI patients show decreased novelty preference on the VPC task compared to age-matched controls and to patients with Parkinsons disease who had no memory impairment [33]. With a delay of 2 minutes between familiarization and test, age-matched controls and patients with Parkinsons disease spent more than 70% of the time looking at the novel stimuli, relative to the familiar stimuli. Amnestic MCI patients, however, viewed novel and the familiar stimuli for almost equal amounts of time, i.e., they did not show novelty preference.

A more recent study by Zola et al. [129] showed that the novelty preference score obtained using the VPC task is able to predict which subjects are likely to exhibit a change in cognitive status from normal to aMCI, and from aMCI to AD, up to three years in advance, with high accuracy. In the same study, performance on the VPC task also revealed those subjects who were unlikely to experience a decline in cognitive function. Thus, the VPC task using infrared eye tracking has been established as a valuable tool for measuring, monitoring and predicting memory impairment.

In this version of the VPC task, participants are presented with visual stimuli grouped in 20 trials. Each trial consists of familiarization phase followed by a blank screen, referred to as delay phase, and a test phase. During the familiarization phase, two identical images are shown side by side for a period of 5 seconds. Each trial may have either a 2 second delay or a 2 minute delay. After the delay, in the test phase, two images appear side by side, the familiar image together with a novel image. Subjects with cognitive impairment are less likely to remember images seen in the familiarization phase and spend almost equal time on both images during the test phase. Control subjects tend to spend more time examining the novel image during the test phase.

Next, we show that more subtle characteristics of eye movements, in conjunction with machine learning methods, can significantly improve the accuracy of detecting patients with an existing AD

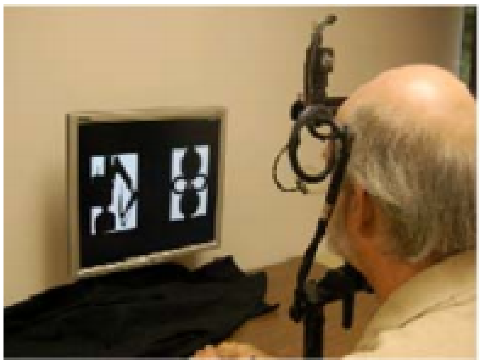


Figure 7.1: Eye tracking equipment used in the VPC task.

diagnosis.

7.1.1 Data

An ASL eye tracker (120 Hz sampling rate) was used for eye movement recording. Additional details about the eye tracking equipment, subject inclusions criteria, VPC stimuli, and the experimental procedure are reported in Crutcher et al. [33].

We analyze data for the following subject groups:

1. The MCI group: 10 subjects diagnosed with mild cognitive impairment (mean age = 72.2 years, SD = 6.9).
2. The AD group: 20 subjects diagnosed with Alzheimers Disease (mean age = 72.4 years, SD = 10.0)
3. The NC group: 30 normal age-matched control subjects (mean age = 70.9 years, SD = 7.1)

7.1.2 Eye Movement Features

We pre-processed the eye movement data and computed the following features:

- **Novelty preference (NP)**: The novelty preference is computed as the fraction of the total looking time spent gazing at the novel image region, and the median novelty preference of the 10 trials with 2-minute delay was used as the NP feature for classification algorithms. The 2-minute delay interval was chosen because that was the delay in which MCI patients demonstrated an impairment relative to control subjects [33].
- **Fixation duration (FD)**: Fixations in the test phases of ten trials (with 2 minute delays) were collected, and the median of fixation duration across all trials was used as an input

feature. The use of this feature is motivated by a previously reported significant difference in fixation durations between NC and AD subjects (Scinto et al., 1994). The change in fixation durations is thought to be related to changes in visual spatial attention, saccade initiation, or inefficiency in planning strategy during visual search observed for AD subjects (Ogrocki et al., 2000).

- **Re-fixations (RF)**: The fixation sequence is used to capture the times when the gaze position re-visits (re-fixates) on previously seen parts of the stimuli. Our algorithm detects re-fixation if there are fixations in the proximity of a previously made fixation and the distance between the centers of the two fixations is less than a specified threshold (5 units in the eye tracker coordinate system, or approximately 2 degrees of visual angle). The depth of re-fixation refers to the number of fixations that occurred between the current fixation and the most recent fixation at the same location. Mean re-fixation depth was computed for each trial and the median across 10 trials was used as an input feature. The use of re-fixations was motivated by the hypothesis that the poor memory in the impaired subjects may be reflected in more frequent or deeper re-fixations. However, we are not aware of prior work that exploits this information, and we thus explored the value of this feature empirically.

- **Saccade orientation (SO)**: After the eye movement trajectory was segmented into fixations and saccades, it could be analyzed further. The saccades were defined by the corresponding endpoints of the fixations. To characterize the saccades, we considered the orientation of the saccades that is, the angles of individual saccades. For this feature, we considered only the absolute value of the saccade angle, ignoring the direction of the movement (i.e., up or down). Specifically, we determined the ratio of vertical saccades (those with the angle of 90 degrees ±7 degrees) to the overall number of saccades in the test phase. The vertical saccades in the VPC task tend to occur within the same stimulus, whereas others are more likely to move the gaze across stimuli, e.g., switch between the novel and the familiar image. The median value of the vertical saccade fractions over all the test trials for a subject was used as the SO feature in classification.

7.1.3 Classifier Evaluation Procedure

Classifier evaluation procedure: our data consisted of 30 control subjects, 20 AD with dementia and 10 MCI subjects. Because our goal was to estimate the classification performance in distinguishing the MCI subjects from the NC subjects, our overall experimental procedure consisted of training the classification algorithms on subsets of the NC subjects and all of the available AD subjects, and then testing the algorithm's prediction on the hold-out (unseen) data consisting of the remainder of the NC subjects and all of the MCI subjects. As classifier evaluation in single train and test cycle is not able to provide reliable estimate of accuracy, we employ a variant of the bootstrap method [37] to repeatedly sample different subsets of training and test data for repeated trials of Cross

Method	Accuracy	Sensitivity	Specificity	AUC
Baseline	0.71	0.712	0.707	0.71
SVM	0.869 (+32%)	0.967 (+61%)	0.772 (+5%)	0.869 (+32%)

Table 7.1: Classification performance of the proposed approach using NP+SO+RF+FD features compared to the novelty preference baseline using only NP.

Validation (CV) in order to obtain more robust estimates of the classifier performance. Specifically, the randomized 3-fold cross validation (CV) scheme was implemented as follows:

1. The NC data were randomly split into 3 folds with 10 subjects in each fold.
2. Two of these folds were combined (comprising the data from the 20 NC subjects), with data from all of the available AD subjects, resulting in the training dataset of 20 NC subjects, and 20 AD subjects. This set of 40 NC and AD subjects was used for training each of the classification algorithms to distinguish the AD subjects from the NC subjects.
3. The remaining (hold-out) part of the NC data (10 subjects) and all of the MCI data (10 subjects) were used to test the classification algorithm predictions and to compute the evaluation metrics.

This process was repeated 100 times, thus resulting in a good sampling of the different partitions of the NC subjects to be included as part of training vs. hold-out test sets. The evaluation metric values (computed in step 3 of each CV step) were averaged across the 100 repetitions and are reported as the final performance in the results section.

Performance Metrics

We measure classification performance with following set of metrics:

- **Accuracy**: the fraction of correctly classified subjects out of all the subjects in the test set.
- **Sensitivity**: the ratio of correctly classified impaired subjects to the total number of subjects in the test set.
- **Specificity**: the ratio of correctly classified normal subjects to the total number of subjects in the test set.
- **Area under the ROC curve** (AUC): the area under the receiver operation curve (ROC), which is a common way to combine the specificity and sensitivity performance of a classification algorithm.

7.1.4 Results

Table 7.1 reports the classification performance when using the Support Vector Machine (SVM) classification algorithm, with all eye movement features (SVM). By exploiting the patterns in the eye movement features SVM is able to achieve Accuracy of 0.869, Sensitivity of 0.967, Specificity of 0.772, and AUC of 0.869. For comparison, we also report the performance of the "baseline" method using only novelty preference. The results demonstrate that SVM, when using the extended eye movement features (novelty preference, saccade orientation, re-fixations, and fixation duration), exhibits relative improvements of 32% on Accuracy, 61% on Sensitivity, 5% on Specificity, and 32% on AUC metrics compared to using the novelty preference information alone. Each of these differences were significant at p<0.001 (two tailed t-test). Repeating the procedure with different numbers of cross validation folds (5-fold or 10-fold CV) produced similar results (data not shown).

7.2 Web based Visual Paired Comparison Task

Collecting data in the VPC requires the use of an eye tracker to precisely monitor subjects' eye movements. An example of a VPC task administered using the ASL eye tracking system is shown in Figure 7.1. The subject's gaze position is captured while the subject is examining the novel and the familiar images. Unfortunately, eye tracking systems, such as the ASL system shown in Figure 7.1, are expensive, require trained personnel, and are not readily available in clinics or research facilities. Interestingly, in a different setting, we showed that restricted focus viewing can accurately approximate un-restricted examination. We apply this idea to develop a web-based adaptation of the VPC task, which we call VPCW, that emulates the VPC experience while replacing eye tracking with mouse tracking.

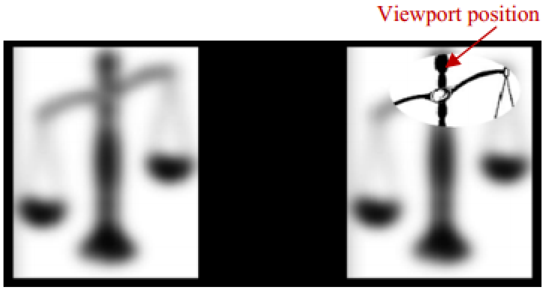


Figure 7.2: Example of vpcw interface showing blurred images and oval-shaped viewport.

As in the VPC task, each trial of VPCW included a familiarization stage, where the subject was shown two identical images side by side, a delay, and then a test stage, where the subject was shown a familiar image and a novel image, side by side. As described earlier, unlike the original VPC task, the VPCW task uses an oval-shaped viewport to reveal a portion of the screen to the subject, while blurring the rest of the screen using a low-pass image filter. The viewport and the blur were designed to simulate foveal and peripheral vision during normal (un-restricted) image examination. Figure 7.2 shows the oval shaped viewport used in the VPCW.

Similar to the VPC task, each trial used one of two delay intervals between the familiarization and test phases. For the VPCW, 17 trials were administered including 5 trials with delay intervals of 10-seconds and 12 trials with delay intervals of 60-seconds. The viewing-time duration of the familiarization and the test phases was 10 seconds. Viewport movements were recorded during both the familiarization and test phases, in order to calculate behavioral metrics. The image stimuli for the VPCW were the same as used for the VPC study reported in Crutcher et al. [33].

Prior to starting the VPCW task, subjects were asked to complete a computer mouse cursor movement calibration task. In this task, the subjects were asked to direct the mouse cursor to a bright yellow circle briefly shown on the screen. The calibration procedure included 9 of these circle locations and was designed to evaluate the subjects skills at using a computer mouse, as well as their reaction time and other fine-grained characteristics of directing the mouse cursor. Cursor movements, as well the position of the target, were recorded during the task. At the end of the calibration task, the subjects were automatically directed to the main VPCW task. A brief tutorial video provided instructions for the VPCW task. The video instructed the subjects to use the computer mouse to move the viewport in order to explore whatever interested them on the computer screen. No other instructions were provided.

7.2.1 Data

To investigate whether the VPCW is sensitive to memory impairment we analyze VPCW data collected from 25 Normal Controls (NC) and 8 amnestic MCI subjects. The primary source for recruiting nearly all participants was the Alzheimer's Disease Research Center (ADRC) at Emory University, Atlanta, Ga. An additional group of 16 subjects, the community control group (COM), was recruited from the surrounding community and served as an additional, self-report control group, without a history of dementia or cognitive impairment. The COM participants were not under the care of a physician for any neurologic condition and reported no cognitive complaints. Informed consent for this study was obtained for all participants in accordance with the regulations of the Institutional Review Board at Emory University.

Participants recruited through the Emory ADRC completed a neuropsychological battery that included the following subtests: Animal Fluency, Boston Naming Test 30 item (BNT-30), and Word List Memory (WLM). Additional neuropsychological tests included Trail-Making Tests Parts A and B (TMT-A, TMT-B), Digit Span (Forward and Backward), and the Clock Drawing Test. In addition, the Geriatric Depression Scale (GDS) was administered to assess for the presence of depressive symptomatology. Group demographic information and neuropsychological performance for the groups are summarized in Table 7.2. In addition to the above, MCI patients also received a full neurological examination. Clinical diagnoses of MCI or NC were established following standardized assessment and review by three clinicians, expert in evaluation and management of Geriatric Neurology patients. Clinical diagnosis of amnestic MCI (aMCI) required evidence of a decline in baseline function in memory and possibly additional cognitive domains, with the severity of symptoms or consequent functional limitations insufficient to meet DSM-III (R) criteria for Dementia. Exclusion criteria across all subject groups included a history of substance abuse or learning disability, dementia, neurological (e.g. stroke, tumor) or psychiatric illness.

In addition to cursor movement data collected as part of VPCW task we collected eye movement data while subjects were performing the VPCW. The eye movements were recorded using Tobii T60 eye tracker. Due to various issues with the eye tracker (software or calibration problems) we were not able to collect eye movement data for all of the subjects. Only 35 subjects (24 of which are normal controls) had eye movement data valid for the analysis purposes.

Data Processing

As described above, the VPCW task session included two components, a calibration task and the VPCW task. The data for both tasks includes the mouse cursor movements captured with the associated timestamps for each data point. That is each data point includes cursor position (X and Y) and the timestamp. Cursor data points recorded in the calibration task were interpolated every 100 ms using a nearest-neighbor interpolation method. Additional metrics derived from the calibration task, described below, were computed based on the interpolated data. To process the image viewing task data, in order to compute novelty preference metrics and other examination behavior characteristics, five areas of interest (AOIs) were defined: Left-Fam, Left-Novel, Right-Fam, Right-Novel and Blank (area outside of the image). The mouse (and corresponding viewport) movements data points were associated with the respective AOIs if the cursor position was within the AOI boundary.

VPCW Behavioral Metrics Collected

VPCW records the data to provide for two groups of behavioral metrics: based on the data from the computer mouse “calibration” task and based on the VPCW novelty preferences.

Cursor Mouse Calibration

The mouse calibration task was used to assess the subject’s reaction and mouse cursor movement skills. While the use of calibration is appropriate to evaluate a subject’s pointing skills (and detect possible VPCW failure to test), data collected in the calibration tasks also serves another important purpose to assess the subject’s executive function and reaction time. To verify this idea, we analyze several metrics: *ReachTargetTime*, *ReactionTime* and *PeakVelocity*. *ReachTargetTime* is computed as the time, in seconds, it took the subject to position the computer mouse cursor over the target, averaged across the nine calibration trials. *ReactionTime* is computed as time, in seconds, it took the subject to start moving mouse cursor after the target circle initially appeared on the screen. Finally, *PeakVelocity* is computed as a maximum velocity of the mouse cursor movement, measured in any of the trials of the calibration task.

Novelty Preference Metrics

Similar to the VPC task, we define VPCW novelty preference as the proportion of time a subject spent viewing the novel image (excluding time spent on blank areas). For each subject, we calculated the average novelty preference for all trials (*NpAll*), computed from the first five seconds of viewing time (measured from the time the novel and familiar images appeared on the screen). Following the prior work on using the VPC task [33], we analyzed the mean novelty preference for trials with a short delay of 10 seconds (*NpShort*), separately from trials with a long delay of 60 seconds (*NpLong*). When computing *NpShort* and *NpLong* scores, we discarded the trials where the total viewing time on the images (i.e., the total time the viewport we placed over the parts of the screen with either the novel or the familiar image) was less than one second.

In addition to the raw novelty preference metric adopted from the original VPC task, we derived an adjusted novelty preference metric. We observed that some subjects exhibited a side preference (e.g., by always starting examination from the image on the left) making it challenging to distinguish the actual novelty preference from idiosyncrasies that can occur when using mouse cursor movement to simulate image viewing. To eliminate potential examination bias originating from these idiosyncrasies, we also calculate adjusted novelty preference (*NpLongAdjusted*) by adjusting the time spent on viewing the left and right images using the average viewing times for left and right images, respectively, across all trials of the subject.

For example, to compute the adjusted time when the novel image is on the left (t_{left}), the original viewing time is divided by the left side bias (p_{left}), computed as the proportion of the time the subject spends viewing the left image across all trials. The right side bias (p_{right}) is defined similarly. More formally, for VPCW trials where the novel image was shown on the left, the *NpLongAdjusted* value was calculated as

$$NpAdjusted = \frac{t^*_{left}}{t^*_{left} + t^*_{right}}$$

where

$$\begin{aligned} t^*_{left} &= \frac{t_{left}}{t_{right} + t_{left}} \frac{1}{p_{left}(subject)} \\ t^*_{right} &= \frac{t_{right}}{t_{right} + t_{left}} \frac{1}{p_{right}(subject)} \end{aligned}$$

For the trials where the novel image was on the right side, *NpAdjusted* was computed equivalently, using the adjusted viewing time t_{right} in the numerator of the above formula, and no other changes.

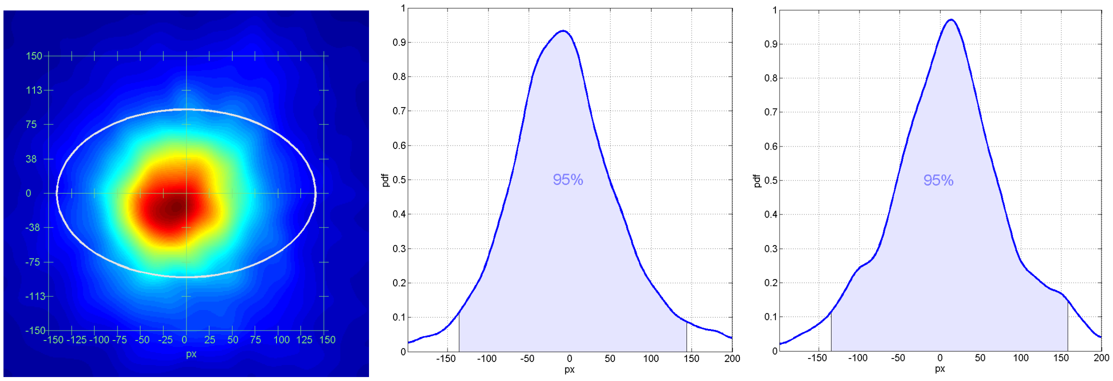
Classification Models

In order to take full advantage of implicit behavioral metrics for improved screening accuracy, we used machine learning techniques to automatically construct classification models that incorporate various behavioral metrics. We use the Classification and Regression Tree (CART, [19]) method to automatically determine optimal classification thresholds based on VPCW behavioral data and associated labels of impairment (impaired vs. normal control). Due to the disparity of positive (impaired) and negative (normal control) subjects in our sample, the accuracy of classification models may misrepresent relative performance differences between the models ([**?**]). Hence, in addition to accuracy, we used the Area Under the Curve (AUC), where the Curve is the Receiver Operating Curve (ROC) to report model performance.

7.2.2 Results

Validation of VPCW

Before discussing whether VPCW is able to measure lack of novelty preference due to the memory impairment, we analyze attention during the VPCW task. More specifically, we compute distribution of eye gaze position centered around the VPCW viewport. Figure 7.3a shows distribution of subject's attention in the VPCW task. We mark boundaries of the oval shaped viewport highlighting the fact that most of the subject's attention is concentrated within the VPCW viewport.



(a) (b) (c)

Figure 7.3: (a) - distribution of attention in VPCW task. Large values are coded with with red color; small values are coded with blue. (b) - distribution of distance between eye and viewport center on dimension x; (c) - distribution of distance between eye and viewport center on dimension y;

Figures 7.3b and 7.3c show probability density functions for horizontal (b) and vertical dimensions (c). The Figures confirm that, indeed, large vast majority of attention occurs within the viewport. We now, turn to discuss our findings on ability of VPCW to detect memory impairment from the cursor movement data.

Detecting Memory Impairment

Table 7.2 shows descriptive statistics for 33 participants (25 NC subjects and 8 aMCI subjects) including demographics, neuropsychological test scores and the associated significance (p) values for the two subject groups (NC and aMCI). Subjects in the NC group were on average 3.5 years older ($M=72$), than subjects in the aMCI group ($M=68.5$). Education levels of the NC and aMCI subjects were found to be significantly different: NC had 17.0 years of education, while aMCI had 14.8 years of education on average ($p<0.05$). Scores of the mini mental state examination (MMSE) were quite similar for subjects in both groups ($p>0.05$). Significant differences were observed between the NC and aMCI groups on the CERAD Animal Fluency Test ($p<0.008$), Word List Memory test ($p<0.05$), Trail Making test (A) ($p<0.05$), Digit Span Backward test ($p<0.04$) and Clock Drawing test ($p<0.01$). By contrast, no significant differences were observed on CERAD Boston Naming Test, Trail Making Test B and Digit Span Forward Tests (all p-values>0.05). The 16 community controls recruited as part of the study had mean age 73.0 (2.16) and years of education 15.7 (0.83), which was not statistically different from the other groups.

	NC (N=22)	**aMCI** (N=8)	p-value
Age at visit	72.0(0.78)	68.5(1.09)	0.031
Education	17.0(0.37)	14.8(1.11)	0.032
MMSE	29.2(0.17)	28.3(0.50)	NS
CERAD Animal Fluency	20.0(0.87)	14.500(1.32)	0.008
CERAD Boston Naming Test	27.5(0.38)	25.167(2.02)	NS
Word List Memory (Total)	22.5(0.84)	17.167(1.62)	0.01
Word List Memory (Delayed Recall)	8.2(0.51)	5.200(1.07)	0.023
Trail Making Test A	31.9(1.96)	44.1(8.47)	0.049
Trail Making Test B	76.8(5.27)	105.5(22.64)	NS
Digit Span Forward	10.9(1.94)	8.8(0.92)	NS
Digit Span Backward	7.0(0.39)	5.1(0.43)	0.037
Clock Drawing Test	12.5(0.33)	9.5(1.38)	0.006
Geriatric Depression Scale	1.5(0.23)	2.5(0.51)	NS

Table 7.2: Subjects demographics and neuropsychological assessment scores. Legend: NC normal control, aMCI - amnestic mild cognitive impairment. Significance (p) values are reported when below 0.05, otherwise reported as NS (not significant).

Table 7.3 summarizes the main VPCW results, and reports the means and standard errors for each of the behavioral metrics for the two subject groups (NC and aMCI). For each of the behavioral metrics we perform a one tailed t-test for samples with unequal variance. The one tailed t-test is appropriate, since we aim to test a directional hypotheses based on previous studies using the VPC test. Namely, we hypothesize that NC subjects exhibit significantly higher novelty preference, compared to the aMCI subjects. This is indeed the case for trials with long delays, but not short delays, as reported in Table 2. The p-values are reported in Table 2 with statistically significant values (p¡0.05) highlighted in bold font. The main findings are discussed in detail below.

Behavioral Metric	**NC**(N=25)	**aMCI** (N=8)	p-value
NpLong	0.59(0.03)	0.54(0.02)	0.086
NpShort	0.55(0.02)	0.58(0.02)	0.148
NpLongAdjusted	0.62(0.02)	0.58(0.02)	0.044
ReachTargetTime	1.33(0.04)	1.52(0.12)	0.075
ReactionTime	0.42(0.02)	0.50(0.04)	**0.042**
PeakVelocity	2.84(0.18)	2.09(0.25)	0.013

Table 7.3: The mean scores and standard errors values for the VPCW behavioral metrics. For each behavioral metric we report the p-value using t-test.

Novelty Preference: Results in Table 7.3 show that NpLongAdjusted differed significantly between the subject groups (p<0.05). Importantly, adjusted novelty preference scores for trials with a long delay (*NpLongAdjusted*) were higher for NC subjects (M=0.62) than for aMCI (M=0.58). This finding validates our hypothesis that control subjects spend a larger proportion of time viewing novel images compared to aMCI subjects, and reproduces previously reported findings [33, 129]. It is worth noting that relatively large p-value is likely due to the relatively small sample size of our data, as the effect size is quite large (Cohen's D = 0.53). As such, we expect the differences in *NpLongAdjusted* to be more pronounced as we recruit more subjects.

Consistent with the findings of Crutcher et al. [33] using the eye tracking-based VPC task, we did not find significant differences in novelty preference for trials with a short delay (*NpShort*, all p-value > 0.05). While novelty preference for long delay trials without adjustment (*NpLong*) did not meet the significance threshold, it showed a substantial difference in group means (NC 0.58 vs. aMCI 0.54) and relatively small p-value of 0.08, which with larger sample may reach the significance level.

Calibration Task: The metrics derived from the calibration task showed significant differences between the subject groups. *PeakVelocity* was found to strongly correlate with memory impairment aMCI subjects were significantly slower in moving the cursor (*M*=2.09), compared to the NC subjects (M=2.84, p=0.03). We found similar trend (aMCI subjects moved the cursor slower) in the values of *ReactionTime* and *ReachTargetTime* time. *ReactionTime* differed significantly with p-value<0.05. *ReachTargetTime* was 14% higher for aMCI subjects, though not statistically significant (p>0.05).

Overall, the data shows that performance on the VPCW task is sensitive to the memory impairment exhibited by aMCI subjects. Thus, these data support the hypothesis that VPCW successfully captures the difference in novelty preference that was found in the original VPC task using eye tracking [33]. Moreover, as indicated in our analysis, multiple behavioral metrics differed significantly between NC and aMCI subjects. Next, we investigate whether incorporating behavioral metrics into an automated classification model can yield a robust and scalable screening method based on the VPCW data collected.

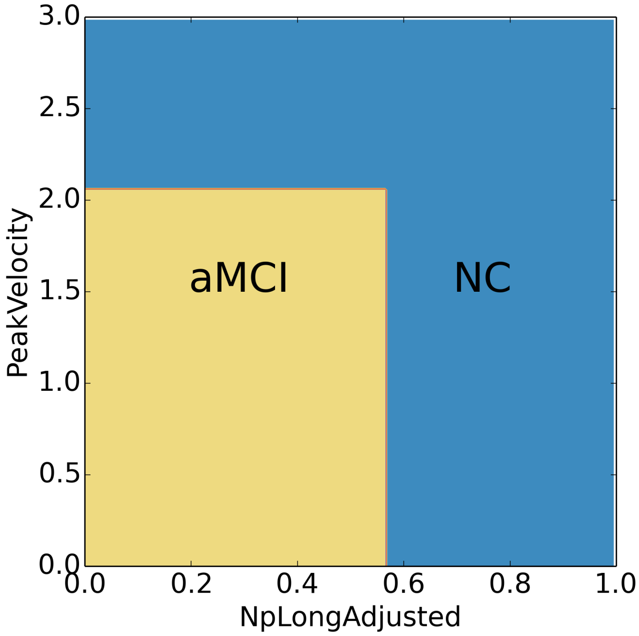


Figure 7.4: Decision boundary derived by the VPCW CART model to distinguish NC and aMCI patients.

Classification Models Trained on VPCW Data Accurately Detect Memory Impairment

Detecting memory impairment by training a classification model to discriminate between NC and aMCI, based on the data collected in the VPCW. Single metric scores using *NpLongAdjusted* and *PeakVelocity* achieve AUCs of 0.690 and 0.760 respectively. The CART model combining these two scores achieves an AUC of 0.855 (accuracy 0.878, sensitivity 0.86 specificity 0.999). Figure 3 shows the corresponding decision boundary for the CART model to distinguish between NC vs. aMCI subjects. The CART model classifies a subject as aMCI if that subject has an *NpLongAdjusted* score of 0.56, and a PeakVelocity score of less than 2.04, otherwise it classifies the subject as an NC. The classification thresholds are consistent with our analysis of VPCW features. That is, the impaired subjects generally have low *NpLongAdjusted* and low *PeakVelocity* scores, compared to the control subjects. These results show that classification models based on VPCW offer high classification accuracy for distinguishing NC subjects from aMCI subjects.

Method	**NC vs. aMCI**
MMSE	0.760
CERAD animal fluency	0.855
CERAD BNT	0.683
WLM Total	0.833
WLM Delayed Recall	0.865
Digit Span Forward	0.638
Digit Span Backward	0.826
Clock Drawing Test	0.725
NpLong	0.645
PeakVelocity	0.760
CART (*NpLong* + *PeakVelocity*)	**0.900**

Table 7.4: Comparison of areas under the ROC curve (AUC) using VPCW and neuropsychological assessments to distinguish aMCI patients from NC subjects.

VPCW Performs Competitively with In-Clinic Neuropsychological Tests

Utility of VPCW as a screening tool for aMCI detection versus neuropsychological tests currently used for in-clinic assessment. Table 3 reports the AUC values for the most effective neuropsychological tests, as well as for NpLong, the performance of the VPCW CART model (based on NpLongAdjusted and PeakVelocity scores of VPCW). The VPCW CART model offers the best classification performance for memory impairment detection in relatively simple (NC vs. aMCI) achieving the AUC of 0.855. By comparison, the most effective neuropsychological tests, WLM Delayed and Digit Span Backward, offer AUC up to 0.865 and 0.828 respectively. While individual neuropsychological test scores are not used for diagnosis in isolation, comparing VPCW performance favorably to existing in-clinic assessments supports our claim that VPCW could be a valuable screening tool on its own, or as part of a neuropsychological test battery.

Figure 7.5 shows a graph with the ROC curves plotted for VPCW models using NpLongAdjusted, PeakVelocity, and the combination of these using the CART decision tree algorithm (VPCW: CART), compared to the two best performing neuropsychological tests (WLM Delayed and Digit Span Backward Test). Note that while VPCW: NpLongAdjusted and VPCW: PeakVelocity individually are not as accurate as neuropsychological tests, their combination allows the VPCW: CART algorithm to reach high accuracy and AUC performance rivaling the in-person, manually administered neuropsychological assessments. Interestingly, VPCW: CART offers the same sensitivity of 0.6 at the zero false alarm rate as the Digit Span Backward test, thus demonstrating the utility of VPCW for first-line automated screening to complement existing neuropsychological assessments.

We now turn to analyzing VPCW performance results in more depth, by first comparing the

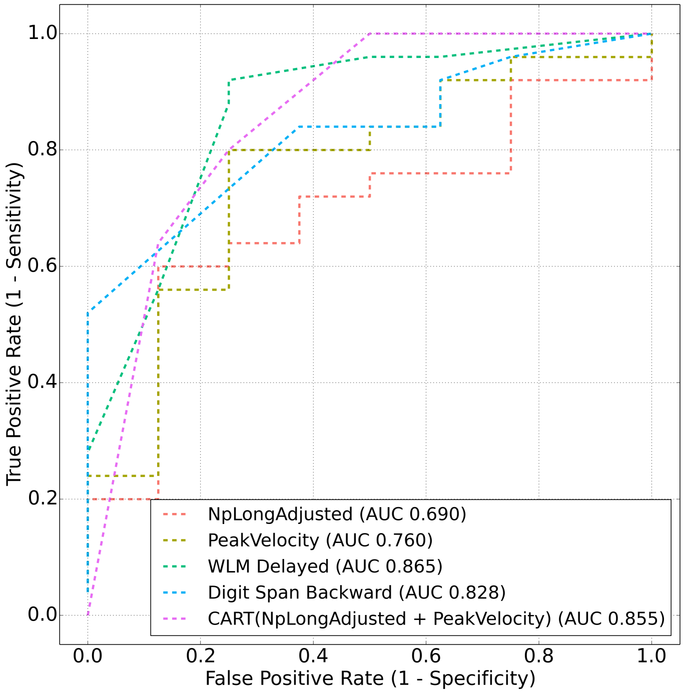


Figure 7.5: ROC curves for *NpLong*, *PeakVelocity*, and combination of the VPCW scores using CART, compared to the two best performing neuropsychological tests (WLM Delayed and Digit Span Backward Test) in the NC vs. aMCI classification task.

Internet-based performance to the original eye tracking-based VPC task, and then evaluating the VPCW task under more challenging settings that may be encountered when used to screen a broad population.

Comparison of eye tracking-based VPC and Internet-based VPCW

In order to understand VPCW performance, we first compared the main measure of the VPC task, the Novelty Preference score, to that elicited by the VPCW task. The VPCW NP scores, reported in Table 7.3, were lower than the previously reported scores measured by the eye-tracking-based VPC studies. Specifically, in the original eye-tracking based VPC task [15], the average *NpLong* (Novelty Preference after the “long” 2-minute delay) scores were 0.68 (SE=0.01) for NC subjects, and 0.62 (SE=0.03) for aMCI subjects. By comparison, the VPCW task scores for *NpLong* were 0.63 (SE=0.02) for NC subjects, and 0.50 (SE=0.03) for aMCI subjects. However, the differences in performance between NC and MCI subjects, as measured by VPC and VPCW, remain consistent: the average difference between the groups for the VPC task was 0.05, and for the VPCW task is 0.13. Moreover, we find the Novelty Preference (*NpLong*) as measured with VPCW to have higher discriminative power (AUC 0.78), compared to the 0.67 AUC in the VPC task (though measured on a different set of subjects). In the eye tracking-based VPC task, combining *NpLong* with additional features of eye movements, such as fixation duration, saccade orientation and number of re-fixations, resulted in a significantly more accurate model achieving AUC of 0.869. The similar performance improvement holds for the VPCW: the CART model incorporating *NpLong* and *ReachTargetTime* metrics achieves a competitive performance of 0.886 on AUC, similar to the discriminative performance of the original, eye tracking-based reported in [92].

Robustness of VPCW

In a second, and even more challenging setting, we included the VPCW data for 16 community control subjects (COM) into the classification, to further diversify and expand the "control" group to 41 subjects. The aim was to explore whether VPCW could remain accurate when screening a more diverse general population compared to the patients typically encountered in a neurology clinic. On this task, the individual VPCW behavioral metrics, (*NpLongAdjusted* and *PeakVelocity*) achieve AUC's of 0.698 and 0.720, respectively. The VPCW CART model combining these two scores achieves AUC of 0.841 (accuracy 0.897, sensitivity 0.909, and specificity 0.975). While these results are slightly lower than in the settings described above, where only clinic subjects were used as both impaired and control subjects, they show that VPCW remains robust even in the challenging setting of testing a diverse population containing NC and community control subjects.

Potential Limitations

It is reasonable to expect that some elderly subjects might be uncomfortable using a computer mouse and are unable to adequately perform the VPCW task. To address this issue, and assess subjects skills when using a computer mouse, we administered the calibration (pointing) task prior to administering the VPCW task. As described previously, during the calibration procedure, a subject is asked to direct the mouse cursor to a nine different target points that are shown, one at a time, on the computer screen. While some of the subjects in our study had little or no previous computer experience, most of the subjects were able to hit the calibration targets: 48 of 49 subjects hit all nine calibration targets. One community control subject hit 8 out of 9 calibration targets. However, all of these 49 subjects were able to successfully complete the VPCW task itself, indicating that VPCW does not present difficulties for the target elderly population.

7.3 Summary

In this chapter we presented important applications of attention tracking to the behavioral testing. The research trajectory from VPC to web based VPC exemplifies the path that could be adopted by other behavioral and attention tests, e.g. to enable early detection of attention deficit hyperactivity disorder, fetal alcohol spectrum disorder and Parkinson's disease[118]. Deployment of such methods for screening of large population has potential to drastically improve the public health and enable early diagnosis of attention and memory disorders.

Chapter 8

Conclusions

This chapter summarizes the contributions, their limitations, and discusses relationship between techniques described in the thesis.

8.1 Summary of Contributions

This thesis presented several techniques for measuring and tracking attention of online users at scale. In addition to the methodological advancements, we described several important applications in Web search and medical domains.

More specifically, we showed (in Chapter 3 and Chapter 7) that restricted focus viewing is a powerful idea that can be very useful in Web search and medical domains. Empowered with crowdsourcing, restricted focus viewing makes large scale usability evaluation not only easily scalable to thousands of participants, but also fully operational and deployable on-demand. However, restricted focus viewing comes with limitations that might restrict potential application of the technique. For example, despite remarkable success in mimicking search result and image examination behavior, it is not clear what exact aspects of eye gaze examination behavior are being sacrificed. It is reasonable to expect that the amount of blur and viewport (oculus) size should impact user experience in RFV studies. In particular, variable amount of blur may induce different behavior in users. For instance, small amount of blur may enable users to see through the vague appearance of the underlying content, and eye gaze to *program* (plan) saccades more effectively. However, with relatively strong blur, when most of the content is masked from the viewer, possibility of "saccade" planning may disappear and users would be forced to move cursor more smoothly, gradually exploring the content. It is advised that these settings would be optimized, for example, using a controlled statistical experiment, for the application at hand.

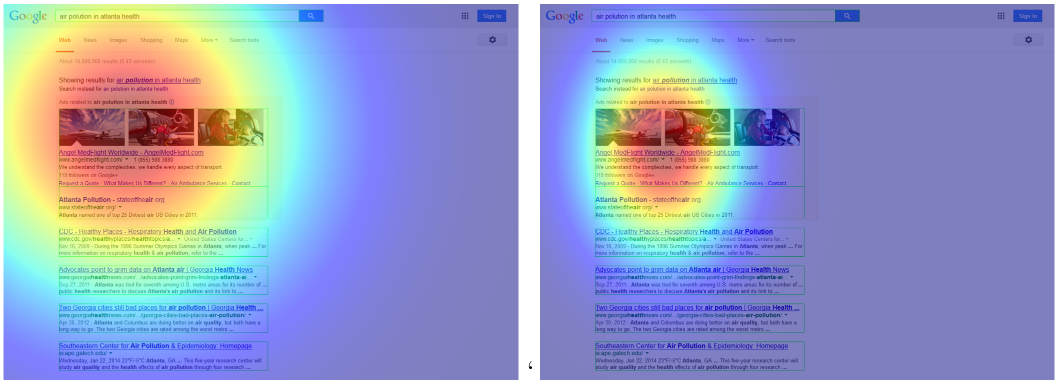
Restricted focus viewing requires modifying the user interface by blurring UI elements, which may potentially alter user experience and bias the measurements. This is done in order to encourage the user to move her mouse cursor to a point of interest, which in turn triggers an actionable event (cursor movement) that logging system is able to record. As a result, an additional burden – to move mouse cursor in a situation when a user might not necessary need it – may potentially create an unrealistic user behavior that is of limited value. On the other hand, in the unrestricted setting, user's mouse cursor and gaze positions might not necessary be coordinated (co-occur), to an extent they are coordinated in the RFV system. Thus, RFV provides natural trade-off between study realism and scalability of participant recruitment.

Regardless whether mouse cursor and gaze positions are coordinated, analysis of cursor movement collected from large scale user population may reveal additional insights about user behavior and enhance an online system. Our approach for mining frequent cursor movement patterns (*motifs*) advances currently available techniques for understanding behavior of online users. In contrast to commonly taken approach for analysis of mouse cursor movements with descriptive statistics, such as speed, acceleration, range and other features of cursor movement, our approach extracts frequent movement patterns that represent typical cursor behavior that might be too difficult to represent with manual features (i.e., descriptive statistics). Our approach, however, does not take into account page content or relative location on page where the cursor movement pattern occur. While this decision may appear as a limitation (and in some sense it is), we make this design choice on purpose due to concerns of practical implementation and information privacy. In fact, by keeping *motifs* independent from page content we considerably simplify the problem by eliminating the need to store detailed information about Web page element visual appearance (as rendered by the user's internet browser). The amount of storage associated with page element data well surpasses typical amount required to store mouse cursor data. Moreover, if cursor tracking instrumentation is deployed via a browser toolbar, the Web page information may not always be available, e.g., in a situation when a user visits secure content, such as online banking or other password protected resources.

Unlike restricted focus viewing, mouse cursor tracking offers recording user attention at much larger scale, as was demonstrated by Huang et al.[65]. However, the accuracy of cursor tracking with respect to actual gaze position remains relatively low and varies significantly depending on type of user action (or absence of action, when cursor is kept still), even when sophisticated regression models on top of cursor features are applied [62, 105]. As we demonstrated in this thesis, it is possible to further improve accuracy of gaze prediction models by incorporating Web page content information into the model, while retaining highly scalable nature of the mouse tracking approach.

In addition to higher accuracy, compared to the regression models, our approach (*MICS*) better manages the intrinsic uncertainty about the probable gaze position by automatically adjusting the variance of the predicted distribution. Whereas regression models suffer from large squared errors, which reach 300px on average (as we saw in Chapter 5). The latter renders prediction of a regression model virtually useless for inferring which result user viewed, since the regression models assume response variable being normally distributed around predicted position with standard deviation of 300px[1]. Comparing typical size of a search result on a Web page (500px × 80px) to 300px error makes the argument even more clear - due to the large uncertainty of prediction up to six results might be identified as "viewed" with relatively high probability. In contrast, *MICS* naturally

[1] Estimated empirically from experiments in Chapter 5.



air polution in atlanta health

About 14,600,000 results (0.43 seconds)

Showing results for air *pollution* in atlanta health
Search instead for air polution in atlanta health

Ads related to **air polution in atlanta health**

Angel MedFlight Worldwide - AngelMedFlight.com
www.angelmedflight.com/ 1 (855) 988 3880
We understand the complexities, we handle every aspect of transport.
119 followers on Google+
Request a Quote - What Makes Us Different? - Air Ambulance Services - Contact

Atlanta Pollution - stateofthe**air**.org
www.stateofthe**air**.org/
Atlanta named one of top 25 Dirtiest **air** US Cities in 2011.

CDC - Healthy Places - Respiratory **Health** and **Air Pollution**
www.cdc.gov/**health**yplaces/**health**topics/a... United States Centers for...
Nov 16, 2009 - During the 1996 Summer Olympics Games in **Atlanta**, when peak ... For more information on respiratory **health** & **air pollution**, refer to the ...

Advocates point to grim data on **Atlanta air** | Georgia **Health** News
www.georgia**health**news.com/.../advocates-point-grim-findings-**atlanta-ai**...
Sep 27, 2011 - **Atlanta** was tied for seventh among U.S. metro areas for its number of ... public **health** researchers to discuss **Atlanta's air pollution** and its link to ...

Two Georgia cities still bad places for **air pollution** | Georgia **Health** ...
www.georgia**health**news.com/.../georgia-cities-bad-places-**air-pollution**/
Apr 30, 2012 - **Atlanta** and Columbus are doing better on **air quality**, but both have a long way to go. The two Georgia cities are rated among the worst metro ...

Southeastern Center for **Air Pollution** & Epidemiology: Homepage
scape.gatech.edu/
Wednesday, Jan 22, 2014 23°F/-5°C **Atlanta**, GA ... This five-year research center will study **air quality** and the **health** effects of **air pollution** through four research ...

(a) Regression Model (b) *MICS*

Figure 8.1: Regression model exhibits larger variance of predicted gaze position, compared to the *MICS* model which automatically adjusts variance depending on behavioral and page content features.

manages the uncertainty and adjusts variance of corresponding mixture distribution components depending on user interactions and Web page elements displayed on the screen at the moment. Figure 8.1 illustrates the point by showing predicted distribution of regression and *MICS* models side-by-side for the same Web page. We see that prediction given by *MICS* model exhibits much smaller variance. More precisely, *MICS* adjusts prediction variance depending on values of behavioral (contextual) and page content features. This allows *MICS* to better fit the data and naturally account for moments when cursor and gaze are coordinated, by setting low variance when user performs action, hence, gaze is likely to follow cursor, and high variance when cursor is kept still. The latter fact relates to vast amount of work on cursor - gaze coordination [27, 28, 115, 53, 62, 105]. Moreover, with access to page content information *MICS* goes beyond the existing work and is able to learn on which *page elements* gaze and cursor are coordinated.

Restricted focus viewing and *MICS* naturally fit into broader context of attention measurement on the Web. Figure 8.2 shows different methods for measuring online user attention and arranges them along two important dimensions: accuracy of attention measurement and number of users a method can potentially be administered to (scalability). Eye tracking is the most accurate method, however, it is the most expensive method, hence, most limited in terms participant recruitment. On the other hand, ViewSer, based on idea of restricted focus viewing, is less accurate than eye tracking, but almost order of magnitude more efficient in terms of data collection and considerably less expensive than eye tracking. At other extreme, mouse tracking (content agnostic) can be deployed to large user population almost instantaneously, though, it is plagued with relatively low accuracy of attention measurements. Finally, content aware models, represented by *MICS* model, integrate both - cursor tracking information and page content information, which allows them to significantly improve measurement accuracy, compared to the mouse tracking (regression) methods,

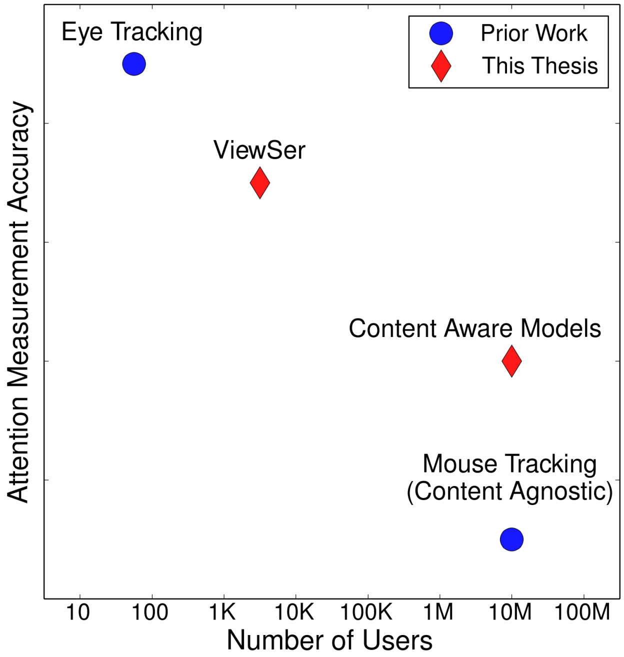


Figure 8.2: Accuracy vs. scalability trade-off between different methods for attention measurement. Eye tracking is the most accurate method, however, it is the most expensive method in terms participant recruitment and equipment costs; ViewSer is less accurate than eye tracking, but almost order of magnitude more efficient and much less expensive; Mouse tracking allows recording of cursor movements from millions of online users, however, it suffers from relatively low accuracy; content aware models (including *MICS*) integrate both: cursor tracking information and page content information, and significantly improve measurement accuracy, compared to the Mouse Tracking models, while being as scalable as mouse tracking methods.

while retaining scalability advantage of the mouse tracking methods.

As we show in Chapter 7, application of restricted focus viewing to image viewing successfully simulates natural viewing behavior and elicits significantly higher novelty preference for normal control subjects, compared to subjects with memory impairment. There is, however, a number of important limitations associated with study reported in Section 7.2. Similarly to ViewSer, we expect RFV factors - amount of image blur and viewport size - are expected to have significant affect on examination behavior. Also, in our study we did not fully explore effect of VPCW trial duration and inter-trial delay on novelty preference. Exploration of these factors becomes trivial given the access to the target population - VPCW task can be administered remotely to the patients during their primary care visits or at other places. VPCW is different from ViewSer in several aspects.

First, unlike ViewSer in Web domain, which acts as an intermediate layer between user and the Web page content, restricted focus viewing in VPCW is more natural, since there is no additional task that user is required to perform, beyond image viewing. Thus, the added cognitive load due to restricted focus viewing is minimal in VPCW, compared to ViewSer. Second, the same amount of blurring applied to each Web page element, regardless of its size, may conceal visually prominent contextual cues which in turn may impact visual search strategies. However, in VPCW such contextual cues are preserved, since all stimuli images are of the same size and approximately same spatial frequency.

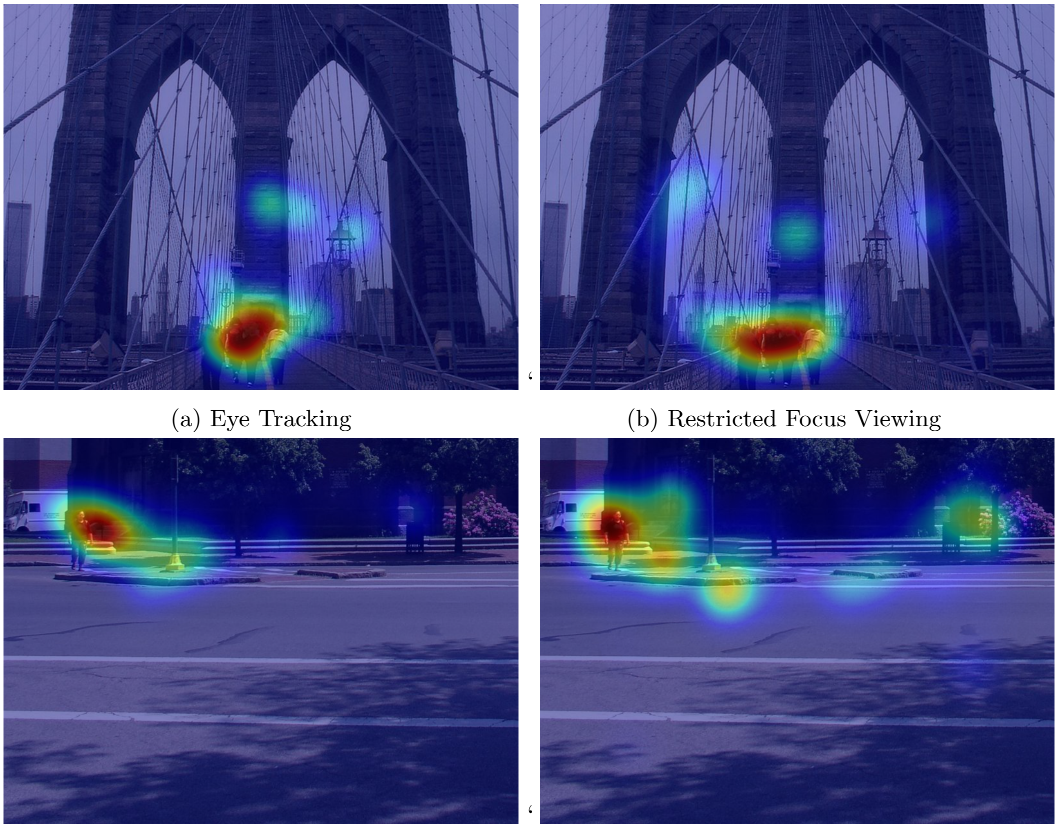
8.2 Current and Future Work

We now turn to discuss future and current research that stems from the work presented in the thesis.

A promising direction that emerges from work on restricted focus viewing and attention modeling from page content is attention tracking on *mobile devices*. Due to relatively small screen sizes such devices are able to display only limited portion of the content at any point of time, which is closely related to the idea behind RFV, although in this case with no need to obscure the content with blur. We expect content aware models to be of great importance in such applications due to several reasons. First, mouse tracking is not available on mobile phones or tablets, as the interaction with the device is performed using tactile interfaces (e.g. touchscreen). Second, as number of element visible to the user decreases (due to small screen size) inference about which particular element is being viewed becomes easier. Some initial work [90] in this direction has already shown promising results on accuracy of attention measurement on mobile phones. With a relatively simple weighting scheme (weights are proportionate to element's size and visible portion) authors showed that viewing time on a search result displayed on a mobile phone is well correlated with the time this result was visible to user multiplied by proportion of the screen occupied by the result (result coverage).

Inspired by the success of VPCW in detection of memory impairment, we expect that VPCW or similar task can be used to screen for other impairment including attention deficit disorder or Parkinson's disease. Indeed, recent research study on using eye movements [124] for detecting Parkinson's dementia and attention deficit hyperactivity disorder from eye movements suggest its feasibility. While in this work we were able to provide only indirect evidence[2] that restricted focus viewing of images closely closely approximates unrestricted viewing for images, some recent work suggests that it is indeed true. Figure 8.3 shows heatmaps for two example images. Two Figures (a,c) show distribution of attention measured with eye tracking (ground truth) and two other Figures (b,d) show attention measured with restricted focus viewing. Clearly, RFV induces

[2] By eliciting significantly higher novelty preference for health control subjects, compared to memory impaired subjects.



(a) Eye Tracking

(b) Restricted Focus Viewing

(c) Eye Tracking

(d) Restricted Focus Viewing

Figure 8.3: Heatmaps showing attention measured with eye tracking (a,c) and restricted focus viewing (b,d).

very similar examination behavior compared to the unrestricted viewing, hence, encouraging further research on the topic. Lastly, it would be desirable to understand to what extent stimuli image features affect attention, and, as a result novelty preference obtained in the VPCW task. That is, we may find that some subjects (health control and memory impaired) consistently exhibit higher novelty preference for the familiar image in some trials (image pairs), simply because one image is more attractive or entertaining. Answering this question may lead to more effective design of the image stimuli for new versions of the VPC and VPCW tasks.

Analysis of user interactions on mobile devices remain relatively unexplored area. Despite recent research effort by Guo and colleagues [57, 90], many aspects of user behavior on mobile devices are yet to be understood. Tactile interactions, such screen touches, swipe and zooming actions represent new level of interactions that may require developing more appropriate models of user behavior. However, some of the approaches could be reused. For example, our work on mining frequent cursor movement patterns may be adopted for mining common sequences of touch interactions or gaze interactions[3]. Our approach for modeling attention from interactions and content may also be adopted for tactile interactions, i.e. touches. Fortunately, our approach naturally accounts for zooming actions, since we eliminate Web page elements outside that are not visible to the user (outside of the browser viewport).

Methods for scalable attention tracking open new ways to optimize engagement of online users with Web site content. To the moment, very limited research is done in this direction. What *parts* of a news article make it particularly interesting to the online readers? What *fragments* of a video clip make it particularly enjoyable to the viewers? These questions could be appropriately addressed by analyzing time the users spend on each piece of the content they consume during news article reading or watching the video. Furthermore, if answers to these questions are determined, one could build more effective models to optimize “for user engagement” and address so called “cold start” problem, widely known in the recommendation system community.

To summarize, in this thesis we developed two alternative methods for scalable attention measurement on the Web. First method, called ViewSer, is based on idea of restricted focus viewing and allows accurate measurement of Web page examination for thousands of participants. Second method utilizes Web page content and user interaction behavior data to accurately infer most likely position of user’s gaze on a Web page. Lastly, we developed a scalable approach for mining frequent cursor movement patters facilitating analysis large amounts cursor data collected from large user populations. In addition to methodological contributions we developed several important applications in Web search and medical domains. First, we showed how search result examination data can be used to infer quality of the search result snippets. Second, we showed cursor movement data could be user to infer search result relevance and enhance search result ranking. Third, we demonstrated how Web page examination data could be used to improve automatic document summarization. Finally, we showed how attention measured with restricted focus viewing could be used in high throughput behavioral screening for memory impairment.

This thesis deals with important problem of scalable and accurate measurement of online user attention. Some of the techniques presented in this have been already adopted in the industry and others are undergoing active development. This work advances state-of-the-art in user attention tracking and offers researchers and practitioners with a new set of attention measurement tools, applicable to wide a range of real world problems.

[3] `https://www.google.com/glass/`

Bibliography

[1] Mikhail Ageev, Qi Guo, Dmitry Lagun, and Eugene Agichtein. Find it if you can: A game for modeling different types of web search success using interaction data. In *Proc. of SIGIR*, 2011.

[2] Mikhail Ageev, Dmitry Lagun, and Eugene Agichtein. The answer is at your fingertips: Improving passage retrieval for web question answering with search behavior data. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1011–1021, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[3] Mikhail Ageev, Dmitry Lagun, and Eugene Agichtein. Improving search result summaries by using searcher behavior data. In *Proc. of SIGIR*, 2013.

[4] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26. ACM, 2006.

[5] Eugene Agichtein, Elizabeth Buffalo, Dmitry Lagun, Cecelia Manzanares, and Stuart Zola. Vpc-w: a web-based visual paired comparison task for early detection of amnestic mild cognitive impairment. 2010.

[6] Omar Alonso, Daniel E Rose, and Benjamin Stewart. Crowdsourcing for relevance evaluation. In *ACM SigIR Forum*, volume 42, pages 9–15. ACM, 2008.

[7] Ernesto Arroyo, Ted Selker, and Willy Wei. Usability tool for analysis of web designs using mouse tracks. In *CHI '06 extended abstracts on Human factors in computing systems*, CHI EA '06, pages 484–489, New York, NY, USA, 2006. ACM.

[8] Anne Aula, Päivi Majaranta, and Kari-Jouko Räihä. Eye-tracking reveals the personal styles for search result evaluation. In *Human-Computer Interaction-INTERACT 2005*, pages 1058–1061. Springer, 2005.

[9] Jocelyne Bachevalier, Mimi Brickson, and Corinne Hagger. Limbic-dependent recognition memory in monkeys develops early in infancy. *Neuroreport*, 4(1):77–80, 1993.

[10] Hila Becker, Mor Naaman, and Luis Gravano. Learning similarity metrics for event identification in social media. In *Proc. of WSDM*, 2010.

[11] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The r*-tree: an efficient and robust access method for points and rectangles. In *Proc. of SIGMOD*, 1990.

[12] R. Bednarik and M. Tukiainen. Validating the restricted focus viewer: A study using eye-movement tracking. *Behavior Research Methods*, 39(2):274–282, 2007.

[13] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a cpu and gpu math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, volume 4, 2010.

[14] Mikhail Bilenko, Sugato Basu, and Raymond J Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proc. of ICML*, 2004.

[15] S. Bird. Nltk: the natural language toolkit. In *Proc. of the COLING/ACL.*, pages 69–72, 2006.

[16] A. Blackwell, A. Jansen, and K. Marriott. Restricted focus viewer: a tool for tracking visual attention. *Theory and Application of Diagrams*, pages 575–588, 2000.

[17] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. 2013.

[18] H Braak and E Braak. Neuropathological stageing of alzheimer-related changes. *Acta neuropathologica*, 82(4):239–259, 1991.

[19] Leo Breiman and Leo Breiman. Bagging predictors. In *Machine Learning*, pages 123–140, 1996.

[20] Nicola J Broadbent, Stephane Gaskin, Larry R Squire, and Robert E Clark. Object recognition memory and the rodent hippocampus. *Learning & Memory*, 17(1):5–11, 2010.

[21] Elizabeth A Buffalo, Seth J Ramus, Larry R Squire, and Stuart M Zola. Perception and recognition memory in monkeys following lesions of area te and perirhinal cortex. *Learning & Memory*, 7(6):375–382, 2000.

[22] G. Buscher, A. Dengel, and L. van Elst. Query expansion using gaze-based feedback on the subdocument level. In *Proc. of SIGIR*, 2008.

[23] Georg Buscher, Edward Cutrell, and Meredith Ringel Morris. What do you see when you're surfing?: using eye tracking to predict salient regions of web pages. In *Proc. of the 27th international Conf. on Human factors in computing systems*, CHI '09, pages 21–30. ACM, 2009.

[24] Georg Buscher, Susan T Dumais, and Edward Cutrell. The good, the bad, and the random: an eye-tracking study of ad quality in web search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49. ACM, 2010.

[25] Georg Buscher, Ludger van Elst, and Andreas Dengel. Segment-level display time as implicit feedback: a comparison to eye tracking. In *Proc. of the 32nd international ACM SIGIR Conf. on Research and development in information retrieval*, SIGIR '09, pages 67–74. ACM, 2009.

[26] Olivier Chapelle and Ya Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web*, pages 1–10. ACM, 2009.

[27] Mon Chu Chen, John R Anderson, and Myeong Ho Sohn. What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing. In *CHI'01 extended abstracts on Human factors in computing systems*, pages 281–282. ACM, 2001.

[28] LIU Chen-Chung and Chen-Wei Chung. Detecting mouse movement with repeated visit patterns for retrieving noticed knowledge components on web pages. *IEICE transactions on information and systems*, 90(10):1687–1696, 2007.

[29] E.H. Chi, P. Pirolli, and S.K. Lam. Aspects of augmented social cognition: Social information foraging and social search. In *Proceedings of the International Conference on Online Communities and Social Computing*, pages 60–69. Springer-Verlag, 2007.

[30] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. of CVPR*, 2005.

[31] C.L.A. Clarke, E. Agichtein, S. Dumais, and R.W. White. The influence of caption features on clickthrough patterns in web search. In *Proc. of SIGIR*, pages 135–142. ACM, 2007.

[32] Mark Claypool, Phong Le, Makoto Wased, and David Brown. Implicit interest indicators. In *Proc. of the 6th international Conf. on Intelligent user interfaces*, IUI '01, pages 33–40. ACM, 2001.

[33] Michael D Crutcher, Rose Calhoun-Haney, Cecelia M Manzanares, James J Lah, Allan I Levey, and Stuart M Zola. Eye tracking during a visual paired comparison task as a predictor of early dementia. *American journal of Alzheimer's disease and other dementias*, 24(3):258–266, 2009.

[34] E. Cutrell and Z. Guan. What are you looking for?: an eye-tracking study of information usage in web search. In *Proc. of CHI*, 2007.

[35] E. Cutrell and Z. Guan. What are you looking for?: an eye-tracking study of information usage in web search. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 407–416. ACM, 2007.

[36] Susan T. Dumais, Georg Buscher, and Edward Cutrell. Individual differences in gaze patterns for web search. In *Proc. of the third symposium on Information interaction in context*, IIiX '10, pages 185–194. ACM, 2010.

[37] Bradley Efron and Robert Tibshirani. *An introduction to the bootstrap*, volume 57. CRC press, 1993.

[38] Benno Erdmann and Raymond Dodge. *Psychologische Untersuchungen über das Lesen auf experimenteller Grundlage.* Niemeyer, 1898.

[39] Henry A. Feild, James Allan, and Rosie Jones. Predicting searcher frustration. In *Proc. of SIGIR*, pages 34–41. ACM, 2010.

[40] J. Friedman, T. Hastie, and R. Tibshirani. Special invited paper. additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2):337–374, 2000.

[41] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[42] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):pp. 1189–1232, 2001.

[43] Jeremy Goecks and Jude Shavlik. Learning users' interests by unobtrusively observing their normal behavior. In *Proc. of the 5th international Conf. on Intelligent user interfaces*, IUI '00, pages 129–132. ACM, 2000.

[44] Joseph H Goldberg, Mark J Stimson, Marion Lewenstein, Neil Scott, and Anna M Wichansky. Eye tracking in web search tasks: design implications. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 51–58. ACM, 2002.

[45] Laura A. Granka, Thorsten Joachims, and Geri Gay. Eye-tracking analysis of user behavior in www search. In *Proc. of ACM SIGIR'2004*, SIGIR '04, pages 478–479. ACM, 2004.

[46] Laura A. Granka, Thorsten Joachims, and Geri Gay. Eye-tracking analysis of user behavior in www search. In *Proc. of SIGIR*, SIGIR '04, pages 478–479, New York, NY, USA, 2004. ACM.

[47] Zhiwei Guan and Edward Cutrell. An eye tracking study of the effect of target rank on web search. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 417–420. ACM, 2007.

[48] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. Click chain model in web search. In *Proceedings of the 18th international conference on World wide web*, pages 11–20. ACM, 2009.

[49] Qi Guo and Eugene Agichtein. Exploring mouse movements for inferring query intent. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 707–708. ACM, 2008.

[50] Qi Guo and Eugene Agichtein. Exploring mouse movements for inferring query intent. In *Proc. of SIGIR*, 2008.

[51] Qi Guo and Eugene Agichtein. Ready to buy or just browsing?: detecting web searcher goals from interaction data. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 130–137. ACM, 2010.

[52] Qi Guo and Eugene Agichtein. Ready to buy or just browsing?: detecting web searcher goals from interaction data. In *Proc. of SIGIR*, pages 130–137, New York, NY, USA, 2010. ACM.

[53] Qi Guo and Eugene Agichtein. Towards predicting web searcher gaze position from mouse movements. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 3601–3606. ACM, 2010.

[54] Qi Guo and Eugene Agichtein. Towards predicting web searcher gaze position from mouse movements. In *Proc. of CHI*. ACM, 2010.

[55] Qi Guo and Eugene 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.

[56] Qi Guo and Eugene Agichtein. Beyond dwell time: estimating document relevance from cursor movements and other post-click searcher behavior. In *Proc. of WWW*, 2012.

[57] Qi Guo, Haojian Jin, Dmitry Lagun, Shuai Yuan, and Eugene Agichtein. Mining touch interaction data on mobile devices to predict web search result relevance. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 153–162. ACM, 2013.

[58] Qi Guo, Ryan P. Kelly, Selden Deemer, Arthur Murphy, Joan A. Smith, and Eugene Agichtein. Emu: the emory user behavior data management system for automatic library search evaluation. In *Proc. of JCDL*, JCDL '09, pages 389–390, New York, NY, USA, 2009. ACM.

[59] Antonin Guttman. R-trees: a dynamic index structure for spatial searching. *SIGMOD Rec.*, 14(2), June 1984.

[60] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *Advances in neural information processing systems*, pages 545–552, 2006.

[61] J. Heer and M. Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the ACM SIGCHI International Conference on Human Factors in Computing Systems*, pages 203–212. SIGCHI, 2010.

[62] Jeff Huang, Ryen White, and Georg Buscher. User see, user point: gaze and cursor alignment in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1341–1350. ACM, 2012.

[63] Jeff Huang, Ryen White, and Georg Buscher. User see, user point: gaze and cursor alignment in web search. In *Proc. of CHI*, pages 1341–1350, New York, NY, USA, 2012. ACM.

[64] Jeff Huang, Ryen W. White, Georg Buscher, and Kuansan Wang. Improving searcher models using mouse cursor activity. In *Proc. of SIGIR*, 2012.

[65] Jeff Huang, Ryen W White, and Susan 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*, pages 1225–1234. ACM, 2011.

[66] Jeff Huang, Ryen W. White, and Susan Dumais. No clicks, no problem: using cursor movements to understand and improve search. In *Proc. of CHI*. ACM, 2011.

[67] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001.

[68] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(11):1254–1259, 1998.

[69] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[70] A.R. Jansen, A.F. Blackwell, and K. Marriott. A tool for tracking visual attention: The restricted focus viewer. *Behavior Research Methods*, 35(1):57–69, 2003.

[71] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[72] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142. SIGKDD, 2002.

[73] T. Joachims. Training linear svms in linear time. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 217–226. SIGKDD, 2006.

[74] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proc. of the 28th annual international ACM SIGIR Conf. on Research and development in information retrieval*, SIGIR '05, pages 154–161. ACM, 2005.

[75] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161. ACM, 2005.

[76] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[77] Marcel Adam Just and Patricia A Carpenter. Eye fixations and cognitive processes. *Cognitive Psychology*, 8(4):441–480, 1976.

[78] T. Kanungo and D. Orr. Predicting the readability of short web summaries. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 202–211. ACM, 2009.

[79] Tapas Kanungo and David Orr. Predicting the readability of short web summaries. In *Proc. of WSDM*, 2009.

[80] Gabriella Kazai, Natasa Milic-Frayling, and Jamie Costello. Towards methods for the collective gathering and quality control of relevance assessments. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 452–459. ACM, 2009.

[81] D. Kelly and K. Gyllstrom. An examination of two delivery modes for interactive search system experiments: remote and laboratory. In *Proceeding of the annual ACM SIGCHI Conference on Human factors in Computing Systems*, pages 1531–1540. SIGCHI, 2011.

[82] E. Keogh and C.A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and information systems*, 7(3):358–386, 2005.

[83] Sang-Wook Kim, Sanghyun Park, and Wesley W Chu. An index-based approach for similarity search supporting time warping in large sequence databases. In *Proc. of ICDE*, 2001.

[84] T. Kiss and J. Strunk. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525, 2006.

[85] A. Kittur, E.H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceeding of the annual ACM SIGCHI Conference on Human factors in Computing Systems*, pages 453–456. CHI, 2008.

[86] Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of Intelligence*, pages 115–141. Springer, 1987.

[87] Dmitry Lagun, Mikhail Ageev, Qi Guo, and Eugene Agichtein. Discovering common motifs in cursor movement data for improving web search ranking. In *Proc. of WSDM*, 2014.

[88] Dmitry Lagun and Eugene Agichtein. Viewser: enabling large-scale remote user studies of web search examination and interaction. In *Proc. of SIGIR*, pages 365–374. ACM, 2011.

[89] Dmitry Lagun and Eugene Agichtein. Effects of task and domain on searcher attention. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 1087–1090. ACM, 2014.

[90] Dmitry Lagun, Chih-Hung Hsieh, Dale Webster, and Vidhya Navalpakkam. Towards better measurement of attention and satisfaction in mobile search. In *Proceedings of the 37th International ACM SIGIR Conference on Research &#38; Development in Information Retrieval*, SIGIR '14, pages 113–122, New York, NY, USA, 2014. ACM.

[91] Dmitry Lagun, Cecelia Manzanares, Stuart M Zola, Elizabeth A Buffalo, and Eugene Agichtein. Vpw: an interactive prototype of a web-based visual paired comparison cognitive diagnostic test. In *Proceedings of the 1st ACM International Health Informatics Symposium*, pages 829–832. ACM, 2010.

[92] Dmitry Lagun, Cecelia Manzanares, Stuart M Zola, Elizabeth A Buffalo, and Eugene Agichtein. Detecting cognitive impairment by eye movement analysis using automatic classification algorithms. *Journal of neuroscience methods*, 201(1):196–203, 2011.

[93] J. Le, A. Edmonds, V. Hester, and L. Biewald. Ensuring quality in crowdsourced search relevance evaluation. In *Proceedings of the ACM SIGIR Workshop on Crowdsourcing for Search Evaluation (CSE 2010)*, pages 17–20. SIGIR, 2010.

[94] Luis Leiva. Automatic web design refinements based on collective user behavior. In *Proc. of the 2012 ACM annual Conf. on Human Factors in Computing Systems Extended Abstracts*, CHI EA '12, pages 1607–1612. ACM, 2012.

[95] Daniel Lemire. Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern recognition*, 42(9):2169–2180, 2009.

[96] S. Liang, S. Devlin, and J. Tait. Evaluating web search result summaries. *Advances in Information Retrieval*, pages 96–106, 2006.

[97] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[98] Lin Lonardi, Jessica, Keogh Eamonn, Stefano, and Pranav Patel. Finding motifs in time series. In *Proc. of the 2nd Workshop on Temporal Data Mining*, 2002.

[99] Lori Lorigo, Maya Haridasan, Hrönn Brynjarsdóttir, Ling Xia, Thorsten Joachims, Geri Gay, Laura Granka, Fabio Pellacini, and Bing Pan. Eye tracking and online search: Lessons learned and challenges ahead. *Journal of the American Society for Information Science and Technology*, 59(7):1041–1052, 2008.

[100] George W McConkie, Paul W Kerr, Michael D Reddix, and David Zola. Eye movement control during reading: I. the location of initial eye fixations on words. *Vision research*, 28(10):1107–1118, 1988.

[101] D. Metzler and T. Kanungo. Machine learned sentence selection strategies for query-biased summarization. In *SIGIR Learning to Rank Workshop*, 2008.

[102] Vidhya Navalpakkam and Elizabeth Churchill. Mouse tracking: measuring and predicting users' experience of web-based content. In *Proc. of ACM CHI*, CHI '12, pages 2963–2972, New York, NY, USA, 2012. ACM.

[103] Vidhya Navalpakkam and Laurent Itti. A goal oriented attention guidance model. In *Biologically motivated computer vision*, pages 453–461. Springer, 2002.

[104] Vidhya Navalpakkam and Laurent Itti. Modeling the influence of task on attention. *Vision research*, 45(2):205–231, 2005.

[105] Vidhya Navalpakkam, LaDawn Jentzsch, Rory Sayres, Sujith Ravi, Amr Ahmed, and Alex Smola. Measurement and modeling of eye-mouse behavior in the presence of nonlinear page layouts. In *Proceedings of the 22nd international conference on World Wide Web*, pages 953–964. International World Wide Web Conferences Steering Committee, 2013.

[106] William H Overman, Jocelyne Bachevalier, Frank Sewell, and Jana Drew. A comparison of children's performance on two recognition memory tasks: Delayed nonmatch-to-sample versus visual paired-comparison. *Developmental psychobiology*, 26(6):345–357, 1993.

[107] Bing Pan, Helene A Hembrooke, Geri K Gay, Laura A Granka, Matthew K Feusner, and Jill K Newman. The determinants of web page viewing behavior: an eye-tracking study. In *Proceedings of the 2004 symposium on Eye tracking research & applications*, pages 147–154. ACM, 2004.

[108] Alex Poole and Linden J Ball. Eye tracking in hci and usability research. *Encyclopedia of human computer interaction*, pages 211–219, 2006.

[109] Domenico Praticò. Alzheimer's disease and the quest for its biological measures. *Journal of Alzheimer's Disease*, 33:S237–S241, 2013.

[110] Filip Radlinski, Martin Szummer, and Nick Craswell. Inferring query intent from reformulations and clicks. In *Proc. of WWW*, 2010.

[111] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proc. of the 18th ACM SIGKDD international Conf. on Knowledge discovery and data mining*, KDD '12, pages 262–270. ACM, 2012.

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

[113] Erik D Reichle, Alexander Pollatsek, Donald L Fisher, and Keith Rayner. Toward a model of eye movement control in reading. *Psychological review*, 105(1):125, 1998.

[114] Erik D Reichle, Keith Rayner, and Alexander Pollatsek. The ez reader model of eye-movement control in reading: Comparisons to other models. *Behavioral and brain sciences*, 26(4):445–476, 2003.

[115] Kerry Rodden, Xin Fu, Anne Aula, and Ian Spiro. Eye-mouse coordination patterns on web search results pages. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*, pages 2997–3002. ACM, 2008.

[116] Kerry Rodden, Xin Fu, Anne Aula, and Ian Spiro. Eye-mouse coordination patterns on web search results pages. In *Proc. of CHI*, 2008.

[117] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1), 1978.

[118] Alexander C Schütz, Julia Trommershäuser, and Karl R Gegenfurtner. Dynamic integration of information about salience and value for saccadic eye movements. *Proceedings of the National Academy of Sciences*, 109(19):7547–7552, 2012.

[119] Bracha Shapira, Meirav Taieb-Maimon, and Anny Moskowitz. Study of the usefulness of known and new implicit indicators and their optimal combination for accurate inference of users interests. In *Proc. of the 2006 ACM symposium on Applied computing*, SAC '06, pages 1118–1119. ACM, 2006.

[120] Jin Shieh and Eamonn Keogh. i sax: indexing and mining terabyte sized time series. In *Proc. of KDD*, 2008.

[121] Alex J Smola and Bernhard Schölkopf. Sparse greedy matrix approximation for machine learning. 2000.

[122] H. Takamura and M. Okumura. Text summarization model based on maximum coverage problem and its variant. In *Proc. of the 12th Conf. of the European Chapter of the ACL*, pages 781–789, 2009.

[123] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, 1980.

[124] Po-He Tseng, Ian GM Cameron, Giovanna Pari, James N Reynolds, Douglas P Munoz, and Laurent Itti. High-throughput classification of clinical populations from natural viewing eye movements. *Journal of neurology*, 260(1):275–284, 2013.

[125] Kuansan Wang, Nikolas Gloy, and Xiaolong Li. Inferring search behaviors using partially observable markov (pom) model. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 211–220. ACM, 2010.

[126] Ryen W White and Georg Buscher. Text selections as implicit relevance feedback. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1151–1152. ACM, 2012.

[127] Ryen W. White and Georg Buscher. Text selections as implicit relevance feedback. In *Proc. of SIGIR*, 2012.

[128] Yisong Yue, Rajan Patel, and Hein Roehrig. Beyond position bias: examining result attractiveness as a source of presentation bias in clickthrough data. In *Proc. of WWW*, WWW '10, pages 1011–1018. ACM, 2010.

[129] Stuart M Zola, CM Manzanares, P Clopton, JJ Lah, and AI Levey. A behavioral task predicts conversion to mild cognitive impairment and alzheimers disease. *American journal of Alzheimer's disease and other dementias*, 28(2):179–184, 2013.

[130] Stuart M Zola, Larry R Squire, Edmond Teng, Lisa Stefanacci, Elizabeth A Buffalo, and Robert E Clark. Impaired recognition memory in monkeys after damage limited to the hippocampal region. *The Journal of Neuroscience*, 20(1):451–463, 2000.