Modelling Search and Session Effectiveness

Alfan Farizki Wicaksono

Supervisors:
Professor Alistair Moffat
Professor Justin Zobel

Submitted in total fulfilment of the requirements of the degree of

Doctor of Philosophy

School of Computing and Information Systems
The University of Melbourne

October 2020

Copyright © 2020 Alfan Farizki Wicaksono

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the author.

Abstract

Search effectiveness metrics are used to quantify the quality of a ranked list of search results relative to a query. One line of argument suggests that incorporating user behaviour into the measurement of search effectiveness via a user model is useful, so that the metric scores reflect what the user has experienced during the search process. A wide range of metrics has been proposed, and many of these metrics correspond to user models.

In reality users often reformulate their queries during the course of the session. Hence, it is desirable to involve both query- and session-level behaviours in the development of model-based metrics. In this thesis, we use interaction data from commercial search engines and laboratory-based user studies to model query- and session-level search behaviours, and user satisfaction; to inform the method for evaluation of search sessions; and to explore the interaction between user models, metric scores, and satisfaction.

We consider two goals in session evaluation. The first goal is to develop an effectiveness model for session evaluation; and the second goal is to establish a fitted relationship between individual query scores and session-level satisfaction ratings. To achieve the first goal, we investigate factors that affect query- and session-level behaviours, and develop a new session-based user model that provides a closer fit to the observed behaviour than do previous models. This model is then used to devise a new session-based metric, sINST. In regard to the second goal, we explore variables influencing session-level satisfaction, and suggest that the combination of both query positional and quality factors provides a better correlation with session satisfaction than those based on query position alone. Based on this observation, we propose a novel *query-to-session aggregation function*, that is useful for scoring sessions when sequences of query reformulations are observed.

We also propose a meta-evaluation framework that allows metric comparisons based on empirical evidence derived from search interaction logs, and investigate the connection between predicted behaviour and observed behaviour, and between metric scores and user satisfaction at both query and session-levels.

Declaration

This is to certify that

1. the thesis comprises only my original work towards the PhD,
2. due acknowledgement has been made in the text to all other material used,
3. the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Alfan Farizki Wicaksono, October 2020

Credits

The material in Chapter 3 is based on the following published papers:

- Alfan F. Wicaksono and Alistair Moffat. Empirical Evidence for Search Effectiveness Models. In *Proc. CIKM*, pages 1571–1574, 2018.
- Alfan F. Wicaksono and Alistair Moffat. Exploring Interaction Patterns in Job Search. In *Proc. Aust. Doc. Comp. Symp.*, pages 1–8, 2018.
- Alfan F. Wicaksono. Measuring Job Search Effectiveness. In *Proc. SIGIR*, page 1453, 2019.
- Alfan F. Wicaksono, Alistair Moffat, and Justin Zobel. Modeling User Actions in Job Search. In *Proc. ECIR*, pages 652–664, 2019.

The material in Chapter 4 (Sections 4.4, 4.5, and 4.6) is currently under review.

The material in Chapter 5 (except Sections 5.5.4 and 5.6.2) is based on the following published paper:

- Alfan F. Wicaksono and Alistair Moffat. Metrics, User Models, and Satisfaction. In *Proc. WSDM*, pages 654–662, 2020.

Acknowledgements

All praise due to Allah (*the most glorified, the most high*) who blessed me with the ability to complete this thesis. I am grateful to my mother, my father, and my sisters, who always pray for me and provide support throughout my life; to my wife Nisa who always loves me; and to my son Budi who brings me joy.

I would like to thank my supervisors, Professor Alistair Moffat and Professor Justin Zobel, for their invaluable support and guidance throughout my doctoral study. I wish to be able to apply what I learn from Alistair and Justin; and to become a good supervisor for my future students. I would also like to thank Professor Trevor Cohn, who served as my committee chair.

I gratefully acknowledge the generosity of the creators of the datasets used in this thesis; and Dr. Paul Thomas (Microsoft) in particular for his assistance. This work was supported by the University of Melbourne, by the Australian Research Council, and by `Seek.com`. I attended ADCS 2018 in Dunedin using travel support from ADCS. I also attended SIGIR 2019 in Paris with support from SIGIR and from University of Melbourne Google-funded travel grants.

Thanks are also due to Dr. Damiano Spina (RMIT University) and Dr. Bahar Salehi (The University of Melbourne) who supported my research; to IR researchers at SEEK, Dr. Sargol Sadeghi and Dr. Vincent Li, who provided access to the `Seek.com` datasets; and to my colleagues at the University of Melbourne, Alicia and Unni, who helped me in my study. Finally, I would also like to thank John Papandriopoulos who provided a template for this thesis.

To those who sincerely search for the eternal truth throughout their lives.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

The search engine has become a primary tool for information seeking and discovery. People who wish to find an explanation about a particular topic can use a Web search engine, such as `Bing.com` or `Google.com`, in order to fulfil their information needs. The success of current search engines cannot be separated from the research and development in the area of *Information Retrieval* (IR).

In particular, search engine improvement cannot be achieved without a reliable *evaluation*. Two kinds of IR evaluation have emerged: *online* and *offline* evaluations. Online evaluation usually involves comparing the quality difference between two systems in a production environment, and using implicit feedback, such as clicks or query reformulations [111, 118, 167]. On the other hand, offline evaluation does not require user feedback. This offline paradigm usually involves three components: (1) a test collection consisting of a corpus, queries, and relevance judgements; (2) *effectiveness metrics* quantifying the quality of the search results rankings; and (3) statistical tests justifying that one system is better than the other. These two evaluation approaches complement each other.

Online evaluation is a powerful technique for the evaluation of IR systems, since it reflects many aspects, including interaction and presentation. However, this approach is usually time-consuming and expensive. In addition, online evaluation, such as A/B testing, might alter users' search experience to the extent that they become disenchanted with the search system. In contrast, offline evaluation is suitable for tasks that need to be repeated multiple times, such as tuning a retrieval heuristic. Further, this kind of evaluation has a long-standing history dating back to the 1960s [50], and serves a basis for IR evaluation events, such as TREC[1], CLEF[2], and NTCIR[3]. This thesis is primarily about offline evaluations and effectiveness metrics.

[1] `https://trec.nist.gov`

[2] `http://clef.isti.cnr.it`

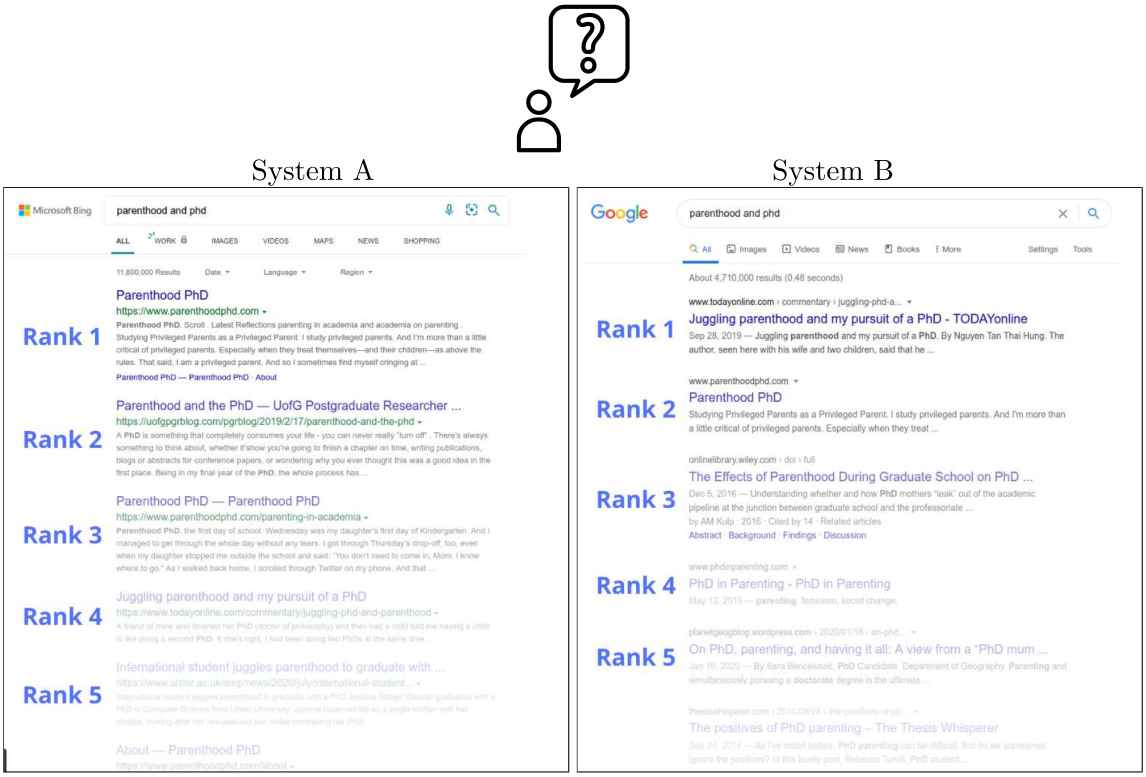
[3] `http://research.nii.ac.jp/ntcir/index-en.html`

In the 1950s, when the term “information retrieval” was first introduced by Mooers [156], two fundamental IR effectiveness metrics, *precision* (the proportion of retrieved documents that are relevant) and *recall* (the proportion of relevant documents in the collection that are retrieved), were described together by Kent et al. [120] along with the concept of *relevance* – a debatable evaluation criterion that should be carefully defined (see, for example, the approach of Schamber et al. [191] and the review of Saracevic [187, 188]) – in order to measure the effectiveness of a search system. Before that, Gull [69] also described the use of recall for comparing the effectiveness of two search systems based on library catalogue entries. This indicates that the development of evaluation metrics was an early priority in the history of IR.

In the 2000s, there have been increasing concerns for the development of metrics from a user modelling perspective, as well as for the relationship between metric scores and user-reported satisfaction ratings. From this perspective, a “good” metric is one that has a plausible user model (a model that describes how users interact with search engine results) [44, 103, 151, 240], and that also has a strong correlation with user satisfaction [6, 91]. Many user-oriented metrics are based on the *user model* embodied in the *expected search length* metric proposed by Cooper [53] in 1968, where users sequentially scan down the ranking until some stopping rank position. Several authors argue that metrics should take into account that users are less likely to examine documents retrieved further down the ranking than top ranked documents, so that the metrics are useful in the context of interactive IR [102, 117, 151]. Figure 1.1 compares the effectiveness of two search results pages for a particular query, including the idea that the examination probability is non-increasing as a function of rank position.

Between 2010 and 2020, a range of model-based metrics were developed [22, 47, 105, 106, 115, 153, 181, 195, 195, 241]. With the increasing number of metrics came the need for classifying and comparing them. In the same period, there has also been a growing interest in generalising existing metrics (and their corresponding user models) in order to develop a framework for search effectiveness metrics. The benefit of having a framework is that existing metrics can be compared under the same score interpretation and under the same characteristics [39, 47, 153, 169, 180].

Recently, Moffat et al. [153, 155] described the C/W/L framework, establishing the relationship between metrics and user models, and showing that many existing metrics, such as precision, fit this structure. With this framework, the user behaviour is characterised by three interrelated quantities, including the *conditional continuation probability* function $C(i)$ that denotes the conditional probability that the user examines the document



parenthood and phd

11,800,000 Results

Parenthood PhD
https://www.parenthoodphd.com
Parenthood PhD. Scroll . Latest Reflections parenting in academia and academia on parenting . Studying Privileged Parents as a Privileged Parent. I study privileged parents. And I'm more than a little critical of privileged parents. Especially when they treat themselves—and their children—as above the rules. That said, I am a privileged parent. And so I sometimes find myself cringing at ...
Parenthood PhD — Parenthood PhD · About

Parenthood and the PhD — UofG Postgraduate Researcher ...
https://uofgpgrblog.com/pgrblog/2019/2/17/parenthood-and-the-phd
A **PhD** is something that completely consumes your life - you can never really "turn off" . There's always something to think about, whether it'show you're going to finish a chapter on time, writing publications, blogs or abstracts for conference papers, or wondering why you ever thought this was a good idea in the first place. Being in my final year of the **PhD**, the whole process has ...

Parenthood PhD — Parenthood PhD
https://www.parenthoodphd.com/parenting-in-academia
Parenthood PhD. the first day of school. Wednesday was my daughter's first day of Kindergarten. And I managed to get through the whole day without any tears. I got through Thursday's drop-off, too, even when my daughter stopped me outside the school and said: "You don't need to come in, Mom. I know where to go." As I walked back home, I scrolled through Twitter on my phone. And that ...

Juggling parenthood and my pursuit of a PhD
https://www.todayonline.com/commentary/juggling-phd-and-parenthood
A friend of mine who finished her **PhD** (doctor of philosophy) and then had a child told me having a child is like doing a second **PhD**. If she's right, I had been doing two PhDs at the same time ...

International student juggles parenthood to graduate with ...
https://www.ulster.ac.uk/amp/news/2020/july/international-student...
International student juggles parenthood to graduate with a PhD Jyotsna Talreja Wassan graduates with a PhD in Computer Science from Ulster University. Jyotsna balanced life as a single mother with her studies, looking after her two-year-old son whilst completing her PhD.

About — Parenthood PhD
https://www.parenthoodphd.com/about

parenthood and phd

About 4,710,000 results (0.48 seconds)

www.todayonline.com › commentary › juggling-phd-a...
Juggling parenthood and my pursuit of a PhD - TODAYonline
Sep 28, 2019 — Juggling **parenthood** and my pursuit of a **PhD**. By Nguyen Tan Thai Hung. The author, seen here with his wife and two children, said that he ...

www.parenthoodphd.com
Parenthood PhD
Studying Privileged Parents as a Privileged Parent. I study privileged parents. And I'm more than a little critical of privileged parents. Especially when they treat ...

onlinelibrary.wiley.com › doi › full
The Effects of Parenthood During Graduate School on PhD ...
Dec 5, 2016 — Understanding whether and how **PhD** mothers "leak" out of the academic pipeline at the junction between graduate school and the professoriate ...
by AM Kulp · 2016 · Cited by 14 · Related articles
Abstract · Background · Findings · Discussion

www.phdinparenting.com
PhD in Parenting - PhD in Parenting
May 13, 2015 — **parenting**, feminism, social change.

planetgeogblog.wordpress.com › 2020/01/16 › on-phd...
On PhD, parenting, and having it all: A view from a "PhD mum ...
Jan 16, 2020 — By Sara Benceković, **PhD** Candidate, Department of Geography. **Parenting** and simultaneously pursuing a **doctorate** degree is the ultimate ...

thesiswhisperer.com › 2014/09/24 › the-positives-of-ph...
The positives of PhD parenting – The Thesis Whisperer
Sep 24, 2014 — As I've noted before, **PhD parenting** can be difficult. But do we sometimes ignore the positives? In this lovely post, Rebecca Turvill, **PhD** student ...

Figure 1.1: Comparison of two ranked lists of results generated from two different systems, `Bing.com` and `Google.com`, for the query "parenthood and phd" (searched on 2020-10-20). Note that the level of opacity increases with rank position, illustrating that the user is less likely to examine documents retrieved further down the ranking than top ranked documents.

at rank $i + 1$, given that they have just examined the one at rank i.

A series of model-based metrics has been proposed by specifying $C(i)$ functions. These metrics (and their publication years) are, respectively, *rank-biased precision* [151] (2008), INSQ [152] (2012), INST [153, 155] (2013), and a metric based on an *information foraging model* [20] (2018). Rank-biased precision has a fixed $C(i)$ at all ranks, but is sensitive to different levels of the user's persistence via its parameter. The $C(i)$ function of INSQ varies with rank position i, and is sensitive to the user's initial goal for undertaking a search activity (goal sensitive). An extension of the INSQ metric, INST, is not only goal sensitive but also *adaptive*, meaning that the user behaviour changes as the user encounters relevance in the ranking. A metric based on the information foraging model is also goal sensitive and adaptive, with an additional property that the user keeps inspecting the ranking as long as the rate of gain exceeds their minimum expectation (rate sensitive).

Existing frameworks, including C/W/L, have provided a foundation to compare the underlying user models across metrics. However, these frameworks focus on evaluating the quality of a single SERP with respect to a single query. Most of the metrics described previously, such as precision and INST, were also developed based on this assumption. However, real interaction between an IR system and a user seeking information usually involves query reformulations for a single information need, leading to a *search session* with multiple queries [95, 121]. Researchers attending the Third Strategic Workshop in Information Retrieval in Lorne (SWIRL) in 2018 discussed the long-term issues of the IR field [9]. As stated in its meeting report by Allan et al. [9, p. 28], one agreed strategic direction is the development of metrics that measure the "success of the search session as a whole", and that are sensitive to the different types of search task.

Furthermore, it remains unclear how to measure the accuracy of user models, that is, how close any model is, in terms of hypothesised behaviours in the C/W/L structure, to the observed user behaviour. An investigation is also needed to see whether metrics that fit observed behaviour tend to produce scores that are correlated with user satisfaction. This meta-evaluation issue is critical for the development of query- and session-based metrics that embody user models, since meta-evaluation provides a justification for "good" effectiveness metrics.

This thesis addresses the problem of using search interaction logs for modelling user behaviour and satisfaction, informs methods for the evaluation of multi-query sessions, and explores the interaction between metric scores, user models, and user satisfaction. This first introductory chapter discusses the overarching issue and research questions, describes the contributions in this thesis, and concludes with an overview of how this thesis is organised.

1.1 Research Questions

The overarching issue of this thesis is: to what extent can search interaction logs be used to model user behaviour and satisfaction, and to derive evidence that allows metric comparisons? More specifically, the following interrelated research questions are considered:

1. Is it possible to use search interaction logs to model user behaviours? If so, what factors affect both query- and session-level user behaviours?

2. What factors affect session satisfaction?

3. To what extent do metric scores correlate with user satisfaction at both query- and session-levels?

4. To what extent do user models predict observed behaviours?

In considering these questions, we develop metric-based user models from the perspective of C/W/L framework via the argument that a good model is one that reflects three hypothetical probabilities: those associated with *viewing*, *continuing*, and *stopping* behaviours [153, 155]; and via the notion of user satisfaction. We also explore the correlation between metric scores and user satisfaction for meta-evaluation of metrics. In this context, the ground-truth of satisfaction lies in the user's mental state, and requires an approximation [117]. One way to approximate the ground-truth of satisfaction is by employing Likert scale user-reported satisfaction ratings. Previous work has constructed lab-based datasets containing five-point style ratings (ranging from *unsatisfied* to *very satisfied*) at both query- and session-levels [46, 104, 105, 139, 145]. These pre-existing datasets are employed in our experiments[4].

1.2 Contributions

The main contributions of our investigation are:

1. We propose three heuristics for computing empirical conditional continuation probabilities (query-level behaviour) from a set of impression sequences (Section 3.3);

2. We propose a new *impression model* for inferring impression distributions from click sequences, and demonstrate that this model is useful for computing empirical continuation probabilities when impression sequences are not available but click sequences are (Sections 3.4–3.6);

3. We extend the query-based C/W/L framework by adding a new quantity, *conditional reformulation probability* (session-level behaviour). Using impression model and methods for computing continuation probabilities, we utilise interaction logs from both commercial search engines and lab-based user studies to investigate factors that affect query- and session-level behaviours (Sections 4.4 and 4.5);

[4] In making use of data collected and made available by other researchers, it is assumed that ethics and other necessary permissions including informed consent have been appropriately obtained by those researchers. Outside of the `Seek.com` data, collection of which was covered by SEEK's user terms and conditions, and by a research agreement between them, RMIT University, and The University of Melbourne, no new data was collected during the course of this project.

4. Using the extended C/W/L structure and findings from observation data, we propose a new session-based effectiveness metric for offline evaluation using session test collection, where each topic is associated with a fixed sequence of queries. Our proposed metric is the first session-based metric that is goal sensitive and adaptive (Section 4.6);

5. We also propose a novel query-to-session aggregation method that is useful for scoring sessions when knowledge of how many times the user reformulated and what queries they submitted is available. This method is based on the combination of individual query qualities and positional factors in the session (Sections 4.7 and 4.8);

6. We investigate the relationship between user model and observed behaviour, and between metric scores and session satisfaction at both query- and session-levels; we also explore whether the metrics with user models that fit observed behaviour also tend to be the metrics that correlate well with satisfaction ratings (Chapter 5); and

7. We propose a method for measuring user model accuracy, and investigate the effect of adaptivity for improving metric accuracy (Section 5.5).

1.3 Thesis Structure

Chapter 2. A wide range of fundamental issues in IR offline evaluation are introduced in Chapter 2. It surveys the use of test collections for assessing the quality of IR systems in an offline fashion, and reviews both traditional and modern search effectiveness metrics. Before addressing the dualism between metrics and user models, Chapter 2 describes research that explores user search behaviours. The connection between metrics and user models is described through the lens of C/W/L framework, and Chapter 2 reviews meta-evaluation approaches that involve a wide range of criteria, including user satisfaction and user model accuracy.

Chapter 3. The first part of our contributions is presented in Chapter 3. First, it describes our proposed methods for computing empirical conditional continuation probability, one of the key quantities in the C/W/L structure, from logged viewing behaviours[5].

[5] This is based on the following published papers:

- Alfan F. Wicaksono and Alistair Moffat. Empirical Evidence for Search Effectiveness Models. In *Proc. CIKM*, pages 1571–1574, 2018.
- Alfan F. Wicaksono. Measuring Job Search Effectiveness. In *Proc. SIGIR*, page 1453, 2019.

However, users' viewing behaviours may not be observable, while clicking behaviours almost always can be, particularly from commercial search engine logs. Chapter 3 then demonstrates that it is possible, to some extent, to predict viewing behaviours from observed clicking behaviours[6]. Next, a new impression model is proposed for inferring view distributions from clickthrough actions. Finally, Chapter 3 shows that the impression model is useful for computing empirical continuation probabilities from click logs[7]. Figure 1.2 shows dependency diagram for three contribution chapters in this thesis, including Chapter 3. Two tools developed in Chapter 3, heuristics for inferring empirical continuation probabilities and impression models, are used for behavioural analysis in Chapter 4 and for meta-evaluation in Chapter 5.

Chapter 4. The second key contribution in this thesis is about the evaluation for multi-query sessions. As shown in Figure 1.2, two goals for session evaluation are considered: (1) the first goal is to develop a user model for evaluation using session test collections, where each topic corresponds to a fixed sequence of queries, with each simulated user assumed to follow that sequence when reformulating queries[8]; and (2) the second goal is to establish a fitted relationship between individual query scores and session-level satisfaction ratings. To address the first goal, Chapter 4 describes an extension to the existing query-based C/W/L structure. Chapter 4 then presents our analysis of commercial search interaction logs in regard to variables influencing query- and session-level behaviours. Note that this investigation requires the impression models and methods for inferring empirical continuation probability proposed in Chapter 3 (see Figure 1.2). A goal-sensitive and adaptive session-based effectiveness metric is then developed using observational results derived from search logs. Finally, Chapter 4 addresses the second goal, exploring factors affecting session-level satisfaction, and proposes a novel session satisfaction model that combines both positional and quality factors. This satisfaction model is useful for aggregating individual query scores in the session when the sequence of query reformulations is known.

[6] This is based on the following published paper:

- Alfan F. Wicaksono and Alistair Moffat. Exploring Interaction Patterns in Job Search. In *Proc. Aust. Doc. Comp. Symp.*, pages 1–8, 2018.

[7] This is based on the following published paper:

- Alfan F. Wicaksono, Alistair Moffat, and Justin Zobel. Modeling User Actions in Job Search. In *Proc. ECIR*, pages 652–664, 2019.

[8] This work is currently under review.

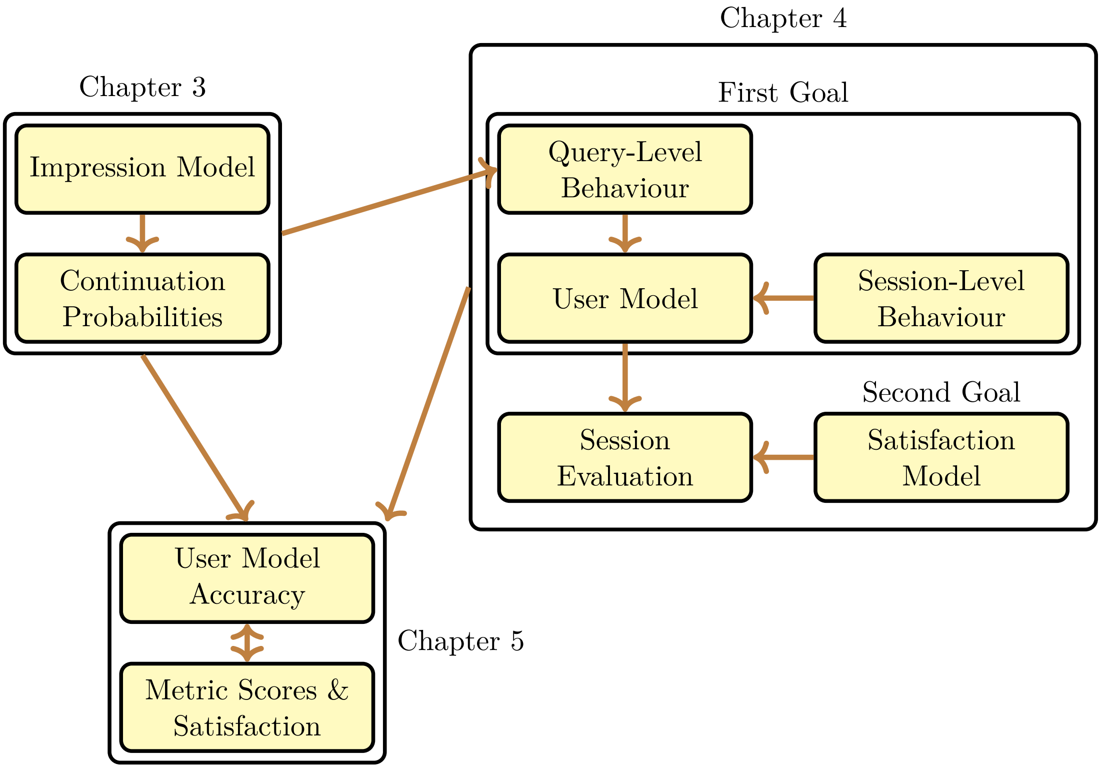


Figure 1.2: Dependency diagram for the thesis contributions.

Chapter 5. Meta-evaluation is the main issue addressed by Chapter 5[9]. First, it explores the relationship between metric scores and user-reported satisfaction ratings at both query- and session-levels (see the box with label "User Model Accuracy" in Figure 1.2). Note that computing session-level scores requires the query-to-session aggregation methods described in Chapter 4. Second, Chapter 5 addresses the dual of that relationship – the relationship between metric-based user models and user behaviour – and describes our proposed method for measuring the accuracy of a user model through the lens of the C/W/L structure (see the box with label "Metric Scores & Satisfaction" in Figure 1.2). The impression models described in Chapter 3 allow for the measurement of user model accuracy via logged behaviours that do not contain impression or eye-fixation sequences, such as click logs. Finally, Chapter 5 considers the question of whether the metrics with accurate user models also tend to be the metrics that have a relationship with user satisfaction.

[9] The material in Chapter 5 (except Sections 5.5.4 and 5.6.2) is based on the following published paper:

- Alfan F. Wicaksono and Alistair Moffat. Metrics, User Models, and Satisfaction. In *Proc. WSDM*, pages 654–662, 2020.

Chapter 2
Background

This chapter introduces the main concepts behind information retrieval evaluation. Section 2.1 describes methodologies and fundamental evaluation metrics for assessing the quality of IR systems. Section 2.1 also argues that the use of recall as a metric for search effectiveness has some problems, including that it is difficult to associate recall with the criteria of user satisfaction. Hence, rather than using recall, or striving to estimate recall, it is more useful to incorporate an accurate user model into the precision-based metric, assuring the connection between metrics and what search users have experienced.

It is thus necessary to consider both system and user contexts in the assessment of search quality, including that an effectiveness metric should embody a user model reflecting how the simulated users interact with the result pages. In this case, a key to development of any user-oriented effectiveness metric is the understanding of the behaviour of search engine users from available search interaction logs. Section 2.2 discusses prior research that explored user interaction patterns from observational data.

After describing the user search behaviour, this chapter introduces C/W/L framework as a tool that connects metrics and user models (Section 2.3). In particular, this framework is useful for comparing user models of existing precision-based metrics, and for developing a new user model (and thus, a new metric) through the same lens of user characteristics. Section 2.4 then describes the classification of existing user models and the process of how they can be mapped into C/W/L framework.

The effectiveness metric itself also needs to be evaluated. This meta-evaluation problem involves defining the criteria for a good metric, including user satisfaction, user preference, user model accuracy, robustness, and sensitivity. This thesis focuses on user satisfaction and user model accuracy as two key indicators for good metrics. Section 2.5 discusses the notion of user satisfaction in the context of IR evaluation. Finally, Section 2.6 elaborates existing meta-evaluation approaches, ranging from empirical-based approaches, such as correlation analysis with satisfaction ratings, to axiomatic-based approaches. To under-

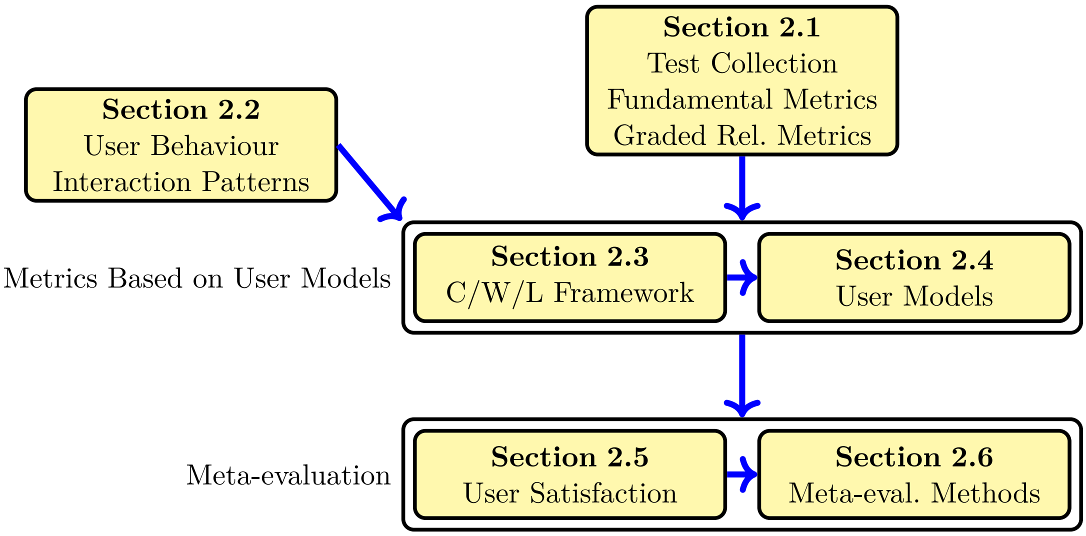


Figure 2.1: Connectedness of the topics in Chapter 2. Yellow rectangles represent sections and their main topics. Several subtopics are also grouped together under common high level topics. Blue arrows represent the reading order that needs to be followed.

stand the structure of this chapter, Figure 2.1 explains the connectedness of the topics that are covered.

2.1 Information Retrieval Evaluation

Evaluation is central to the development of information retrieval systems. However, evaluation of any IR system is a complex task, involving many facets, such as the system being evaluated, evaluation criteria, metrics, instruments, and methodology [188]. There are at least two major paradigms in IR evaluation: (1) user-based evaluation involving a representative sample of search engine users; and (2) test collection-based evaluation, which requires a collection of documents, a set of topics, and relevance judgements. This section focuses on the latter paradigm, exploring some of its requirements, in particular test collection-based (offline) metrics.

2.1.1 The Use of Ranking, Search Success, and Evaluation

The Use of Ranking. The typical interaction between an information retrieval (IR) system and a user can be described as follows: (1) the user submits a *query*, a realisation of an information need, to the system; (2) the system then generates result pages containing a set of items or documents, which is deemed to provide the answers; and (3) the user finally interacts with the result pages to fulfil their information need. Van Rijsbergen [222, p. 6] states that two main objectives of an IR system are "(1) to retrieve all the relevant documents and (2) at the same time retrieving as few of the non-relevant as possible". These objectives are also described by Cleverdon and Keen [51] as the two factors influencing the success of reciprocal actions between the system and the user. Therefore, one crucial challenge is to find a mechanism that is able to organise the retrieved set of documents, so that users receive as much useful information as possible from the retrieved items.

The most widely adopted search result organisation is the *ranked list* presentation, where the retrieved documents are organised in a ranking-style interface and ordered based on their system-estimated likelihood of being relevant. As a result, a document that has the highest estimated relevance score is placed in the top rank position, and thus users expect to see part of the ranking that is most likely to satisfy their information need.

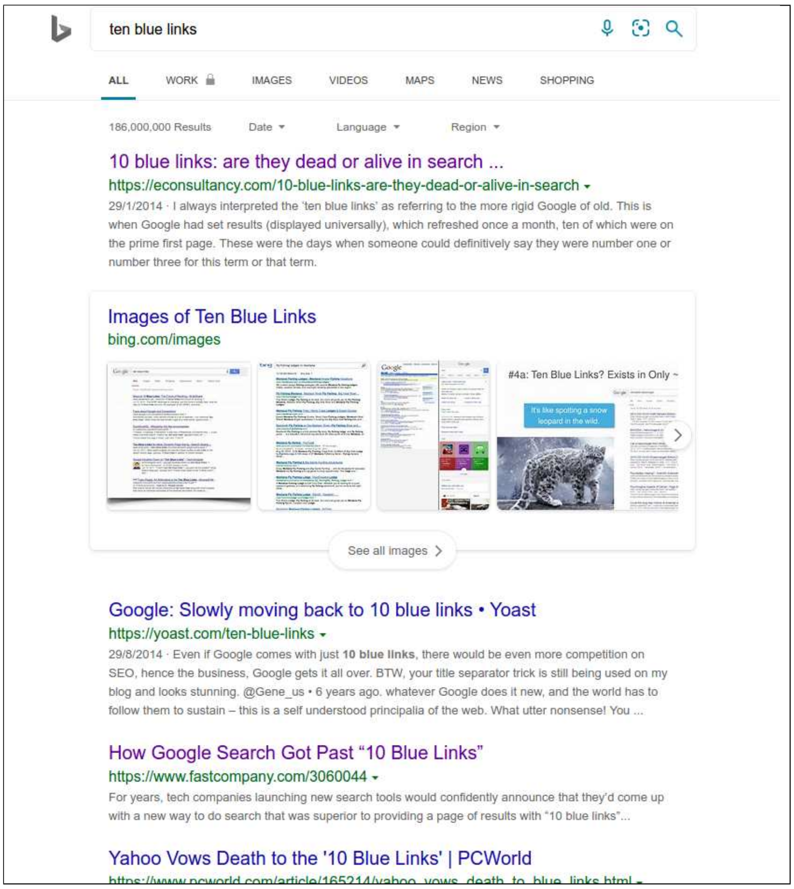
Robertson [170] describes the history of ranked retrieval systems. According to the Robertson's review, the idea of using a probabilistic approach for generating results ranking was proposed by Maron and Kuhns [146] in 1960. Maron and Kuhns [146] describe the concept of ranked retrieval in the abstract of their paper:

> The resulting technique called "Probabilistic Indexing," allows a computing machine, given a request for information, to make a statistical inference and derive a number (called the "relevance number") for each document, which is a measure of the probability that the document will satisfy the given request. The result of a search is an ordered list of those documents which satisfy the request ranked according to their probable relevance.

In the 1960s, Salton also started carrying out historical series of experiments using the SMART system, which generates ranked list of results (see, for example, the experiment results by Lesk and Salton [131]).

The theoretical justification for results ranking presentation is the *probability ranking principle* (PRP), which was envisioned by Cooper [54] in the 1970s, and analysed by Robertson [168] in 1977. This is also recorded in Robertson's book [168, p. 295] as follows:

> If a reference retrieval system's response to each request is a ranking of the documents in the collections in order of decreasing probability of usefulness to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data made available to the system for this purpose, then the overall effectiveness of the system to its users will be the best that is obtainable on the basis of that data.


ten blue links
ALL
WORK
IMAGES
VIDEOS
MAPS
NEWS
SHOPPING
Date
Language
Region
#4a: Ten Blue Links? Exists in Only ~
It's like spotting a snow leopard in the wild.
See all images

186,000,000 Results

10 blue links: are they dead or alive in search ...
https://econsultancy.com/10-blue-links-are-they-dead-or-alive-in-search
29/1/2014 · I always interpreted the 'ten blue links' as referring to the more rigid Google of old. This is when Google had set results (displayed universally), which refreshed once a month, ten of which were on the prime first page. These were the days when someone could definitively say they were number one or number three for this term or that term.

Images of Ten Blue Links
bing.com/images

Google: Slowly moving back to 10 blue links • Yoast
https://yoast.com/ten-blue-links
29/8/2014 · Even if Google comes with just **10 blue links**, there would be even more competition on SEO, hence the business, Google gets it all over. BTW, your title separator trick is still being used on my blog and looks stunning. @Gene_us • 6 years ago. whatever Google does it new, and the world has to follow them to sustain – this is a self understood principalia of the web. What utter nonsense! You ...

How Google Search Got Past "10 Blue Links"
https://www.fastcompany.com/3060044
For years, tech companies launching new search tools would confidently announce that they'd come up with a new way to do search that was superior to providing a page of results with "10 blue links"...

Yahoo Vows Death to the '10 Blue Links' | PCWorld
https://www.pcworld.com/article/165214/yahoo_vows_death_to_blue_links.html

Figure 2.2: A SERP containing a ranked list of links and summaries, plus vertical results (images), generated by `Bing.com` on 2020-09-24, for query "ten blue links".

The main concern with PRP is that it assumes that the relevance of a particular item is not affected by that of other items in the same ranked list. This assumption is at odds with what users perceive. For instance, consider a case where two identical useful documents are located in the first and third rank positions. Users would probably regard the third document as being *not useful* since they have already seen the same document at rank position 1 [68]. If we expect diversity in our search results, PRP has no longer provided an optimal result [230]. Much past research has addressed diversity and novelty in the ranked list of results [3, 40, 48, 166], and propose a counterpart of PRP, such as *PRP for interactive information retrieval* (IIR-PRP) [67], and a more *dynamic* version of IR [193].

However, the use of variants of PRP results basically still makes use of the ranked list of documents as the main presentation method, but with a *re-ranking* mechanism, which is responsive to the user feedback. Indeed, apart from being simple, the main advantages of the ranking-based search results presentation are: (1) its evaluation instruments have been well-developed; (2) it is adopted by major IR evaluation conferences, such as Text REtrieval Conference (TREC), Cross-Language Evaluation Forum (CLEF), and NII-NACSIS Test Collection for IR systems (NTCIR) [225]; (3) major commercial search engines, such as `Google.com`, `Bing.com`, `Yandex.ru`, and `Yahoo.com` still employ the ranked list style of presentation. For commercial search engines, a search engine results page, SERP, was traditionally a ranking with "ten blue links" to relevant items. However, the development of search interface recently has provided SERPs with more complex vertical results, such as images, maps, videos, and wiki-boxes [231]. Figure 2.2 shows a snapshot of SERP initiated from `Bing.com` using the query "ten blue links".

As an alternative to the *ranked list* style, some authors propose the use of *cluster*-based presentation, where a set of retrieved documents is grouped into clusters based on topical proximity [81, 99]. The justification of this presentation style is the *cluster hypothesis*, which states that "closely associated documents tend to be relevant to the same requests" [99, 222]. Hearst and Pedersen [81] propose *Scatter/Gather*, a cluster-based SERP browsing mechanism, which in turn provides empirical evidence that supports the cluster hypothesis. In the Scatter/Gather paradigm, first the user poses a query; second, the system then responds with the top-n documents and clusters them into several topically-coherent groups; finally the user inspects a subset of retrieved documents by selecting a particular cluster that interests them. Each cluster of retrieved documents is associated with a descriptive textual summary consisting of a set of topical words characterising its information. Hearst and Pedersen [81] carried out a user study to evaluate the effectiveness of Scatter/Gather, and the results suggested that in most cases the participants successfully chose the cluster with the largest number of relevant documents, suggesting that the clustering paradigm offered by the Scatter/Gather is beneficial to users.

Leuski and Allan [132] combined the *ranked list* and the *clustering* paradigms by positioning a *spring-embedded visualisation* between two ranked lists (left and right ranked lists) on the screen. In the visualisation, a document is represented as a single colored circle, indicating the proximity to the known highly scored (and slightly scored) documents, and any two similar documents should be close to each other. Hence, this visualisation allows users to determine the clusters of documents that potentially provide useful information, or to visually decide what other documents are topically similar to a particular document. If a particular document in a ranked list is selected, the corresponding point in the *spring-embedded visualisation* is then highlighted. To evaluate the effectiveness of this visualisation, Leuski and Allan [132] conducted user studies to compare the use of the *spring-embedded visualisation* with that of the traditional ranking style and found that, in general, users were satisfied with their visualisation and clustering mechanism.

Research into search interfaces has developed rapidly. Teevan et al. [209] find that users tend to perfom *orienteering* (a behaviour whereby users follow a series of small steps to find the target information instead of jumping directly using a keyword search) even though they merely search for a specific information, such as phone numbers or office addresses. Teevan et al. [209] further suggest to incorporate this phenomenon into the design of future search interfaces. Cutrell et al. [60] designed a search interface, *Phlat*, that aims at improving the quality of a personal search system. One of its characteristics is that it allows users to tag their personal content with textual metadata, so that they can easily return to their items in the future. Arguello et al. [15] propose an approach for combining search results from different verticals (webpages, images, videos, news, and so on) in a ranked list style of presentation. They formulise the task as a block-ranking problem, whereby each block is defined as a group of items that should be placed together in the ranking. Wang et al. [231] propose a method to render different vertical search results in an optimal way. They define the task as an optimisation problem, so that search results and their corresponding presentations maximise the user satisfaction score. In contrast to Arguello et al. [15], the proposal of Wang et al. [231] does not restrict search results from being presented in a ranked list style. Moreover, they could also be presented in multiple columns on the same page. Overall, these research projects are applicable to the search engines that still rely on the ranked list style of presentation.

The Success of IR Systems. A typical IR system returns a ranked list of documents sorted by system-estimated relevance (according to the user's information need) with likely best one placed in the top of the ranking. The user then interacts with the ranking in any of a wide variety of ways. For example, the user might scan all results before clicking at a particular link; or the user might abandon the SERP after examining all snippets from rank 1 to 5 since none of them seems to be relevant. Cleverdon and Keen [51] list five factors that contribute to the success of a search activity:

1. The presence of relevant documents,
2. The absence of non-relevant documents,
3. The response time between the query submission and the search results being generated,
4. The presentation of the search results, and

5. The effort being made by the user.

Sanderson [184] added further factors to the list, such as the quality and representativeness of the query being submitted, the context of the query, and the type of information need. All of these factors should be considered when measuring the utility of a SERP.

Ranking Evaluation. Suppose two IR systems return two different ranked lists of documents for the same query. One of the critical questions to ask is: "which ranking is better?", or in a broader sense, "which system is more effective?" In a production environment, the difference in quality between the two rankings can be inferred from implicit feedback generated by users, such as clickthroughs and query reformulations [111, 118, 167]. This evaluation mechanism is often referred to as *A/B testing*, which is a form of controlled experiment testing a causal relationship between system changes and their effects on the behaviour of users [124]. Note that A/B testing is widely employed for major commercial search engines [124]. Another family of online evaluation method is the *interleaving* approach [45], where two rankings initiated from the same query are interleaved into a single ranked list using a certain strategy [111, 167]. The clickthrough information observed from the combined ranking is then used to decide which system provides better rankings.

The relative quality of two rankings can also be assessed in an offline fashion using a *test collection*, which typically consists of a collection of documents, a set of queries, and a set of relevance judgements. With this evaluation paradigm, each item in the ranking is assigned a *relevance* score (often in a binary scale, where 0 is for a non-relevant document, and 1 for a relevant one), transforming a ranked list of items into a relevance vector. For example, consider two SERPs of length five with the following relevance vectors:

$$\overrightarrow{r_1} = \langle 0, 0, 0, 1, 1 \rangle \,, \qquad \overrightarrow{r_2} = \langle 1, 1, 1, 1, 1 \rangle \,,$$

where the first SERP contains relevant items at rank positions 4 and 5, while the second one is full of relevant items. By examining at the two vectors, one can easily conclude that the second SERP might have a better quality than the first one. However, as the vector becomes more complex, an *effectiveness metric* becomes necessary to the assessment of a ranking by transforming its relevance vector into an *effectiveness score*, which is deemed to reflect what users have experienced when interacting with the SERP. A key property of any effectiveness metric is that it should depend on part of the ranking that has been inspected by the user [188]. However, some traditional metrics, such as *average precision* and *recall*, fail this expectation.

2.1.2 User-Based and Test Collection-Based Evaluation

The field of IR has a long history of using an experimentation-based approach for conducting an evaluation task. According to Voorhees [225], there are two broad classes of IR evaluation: user-based evaluation and test collection-based (or system) evaluation. The latter is based on the Cranfield evaluation paradigm [50]. These evaluation paradigms complement each other. User-based evaluation can be conducted in several ways, ranging from laboratory-based user studies, which require a direct observation of user behaviours, interviews, and questionnaires, to online experimentation, such as A/B testing [124] and interleaving [45]. Meanwhile, test collection-based evaluation allows researchers to perform offline experimentation without directly receiving (either implicit or explicit) feedback from actual users.

User-Based Evaluation. User satisfaction has always been the goal of any IR system [55], or at least an important concept that cannot be dismissed in IR evaluation [199]. Therefore, user-based evaluation seems to be an appropriate way to evaluate the effectiveness of a particular search system since it can capture many user aspects, such as satisfaction, presentation, and interactions [29, 201, 209]. This evaluation method can be conducted in a laboratory environment, where research participants are asked to complete tasks using a search engine, while at the same time, their search activities, including click-throughs, dwell time, query reformulations, and mouse hovers, are recorded by the system. Other user studies also employ eye-tracking tools to capture sequences of eye fixations on the screen [107, 110]. Researchers might also interview their experimental subjects, or ask them to complete a questionnaire, in order to obtain feedback on the extent to which they feel satisfied at the end of a search session. All types of feedback from participants are then combined to evaluate the quality of the tested search engine.

However, a traditional lab-based user study might not be sufficient as the complexity of the system increases. Online experimentation are needed to address the issue of complexity and scalability of the target system [85, 87]. One of the most popular online evaluation approaches to compare the performance of two systems in a production environment is A/B testing. However, this evaluation method requires large amount of data so that the evaluation results are reliable, and also risks altering the search experience of users to the extent that they may become alienated with the service. Another online experimentation, called *interleaving*, can give reliable comparisons of retrieval algorithms using less data than A/B testing [45]. With interleaving, two rankings – the rankings “A” and “B” – generated by two different systems for the same query are interleaved using a certain rule.

The combined ranked list is then given to users, and the comparison results are obtained by interpreting the resulting clickthrough data. Several interleaving strategies have been proposed [111, 167]. *Balanced interleaving* randomly picks a ranking to fill the first rank slot in the interleaved ranking, and then the two rankings consecutively contribute to the remaining part of the combined ranking with duplicate items being skipped [111]. This interleaving strategy, however, suffers from introducing biased results: when the two rankings are almost identical, it tends to favour one over another. Radlinski et al. [167] propose *Team-Draft interleaving* to remove this bias issue. The idea is to think of the problem as combining two football teams, for which each team has a captain who has the right to select the next best player (that is, a document), and add the player into their team and the interleaved ranking. In each turn, the captain whose team has fewer members than the other should pick their player first. However, when both teams have the same number of players, a random selection is applied.

Test Collection-Based Evaluation. User-based evaluation method can be expensive and time-consuming. In particular, the group of participants should ideally be a representative sample drawn from the actual users of the IR system being evaluated, and that may entail expense and also a decreased ability to repeat and reproduce any particular measurement. This can be problematic for some tasks that may need to be repeated as small changes happen in the system.

The main advantage of a test collection-based evaluation, as opposed to the user-based approach, is that it can be repeated multiple times at a lower cost, and thus it is suitable for tasks that involve running multiple experiments, such as tuning a retrieval heuristic. Indeed, the original goal of test collection-based experimentation is to build and optimise heuristics for finding and ranking a set of items for a query [52].

One well-known approach to collection-based evaluation was developed at the Cranfield Aeronautical Laboratories in the 1960s using three main assumptions [50]:

1. The relevance of a document is represented as the topical similarity between the topic, a textual representation of information needs, and the document;

2. All users in the population correspond to the same relevance judgements, meaning that the relevance judgement for a query or topic is agreed by all users in the population; and

3. Each topic or query has a complete list of relevant documents.

This Cranfield paradigm has been widely adopted in major IR evaluation efforts, such as TREC, CLEF, and NTCIR [184, 225]. In practice, the instantiation of this paradigm usually involves three components: (1) a test collection, (2) evaluation metrics, and (3) a mechanism to justify that one method is better than the other, based on the scores generated by the evaluation metrics. Voorhees [225] further note that the former component (that is, a test collection) is divided into three: (1) a set of documents, (2) a set of topic statements, and (3) a set of relevance judgements (or *qrels*). Moreover, the set of documents in the test collection must be from the same domain as the operational setting being evaluated. For example, if the domain of the operational setting is about legal issues, it would not be appropriate to use a collection of documents from medicine.

A set of topics is a collection of queries that will be given to the retrieval system. The system then runs all topics against the set of documents in the collection using its underlying retrieval algorithm. In a TREC evaluation, a topic generally contains an identifier, a title, a description, and a narrative section, expressed in text [225].

A test collection is not complete without a set of relevance judgements that assign relevance scores to all topic-document pairs. TREC evaluations usually employ a binary indicator to assess the relevance of a document with respect to a particular topic (1 for relevant, and 0 for non-relevant). It is also possible to use a more fine-grained relevance score (*graded relevance*) [102, 172]. Furthermore, there are many guidelines for relevance assessment [198, 225]. In case of TREC, Voorhees [225] suggests that the relevance assessors should pretend as if they were making a written report about a particular topic in order to produce a reliable set of relevance judgements. If any part of a document being judged helps the assessors to complete their writing task, the document then is marked as “relevant”, otherwise it is “non-relevant”. In addition, relevance judgements can also be obtained via crowd-worker assessments [238].

Once we have a test collection, we load and index all documents in the collection into our retrieval system, and subsequently run all topics againts all documents using a retrieval algorithm. For each topic, the system returns a ranked list of documents which have been sorted by decreasing order of the score of being relevant, which is known as a *run*. After that, we compute an effectiveness score of the system for a particular query using the provided relevance judgements (qrels) and an evaluation metric. The overall effectiveness score is usually computed by averaging the scores across all topics. However, as is noted by Voorhees [225], a test collection-based effectiveness score cannot be used in isolation. That is, while relative scores can be used as a tool to compare one retrieval system to another (or to contrast different configurations of the same retrieval system) on the same test collection, they should not be regarded as having meaning in isolation as absolute quantities. Moreover, any test collection experimentation should also adhere to the common rules for experimental design [208].

A test collection and evaluation metric, together, can be thought of as a *simulation* of users as if they were interacting with search result pages in an operational setting [184]. Hence, the resulting score can be interpreted as the utility derived by the *simulated user* when they are examining the ranking, suggesting a prediction of how well a system will perform relative to another under the same simulated operational setting [184].

Although the assumptions embodied in the test collection-based evaluation provide a simple way to evaluate and compare different retrieval *runs* (under the same test collection), as well as allowing easy repetition of the evaluation process with other retrieval systems, some problems still remain. A series of studies found that the results from a test collection-based evaluation are not in good agreement with those from a user-based evaluation, in the sense that the systems that perform well according to a test collection-based evaluation might not do so under a user-based evaluation [83, 218, 219]. Kelly [117] argue that this phenomenon could be due to the idiosyncratic nature of relevance. For example, a user might consider a particular document as being relevant although the TREC assessor might not think so, and vice versa. In general a test collection-based evaluation, such as TREC-based evaluation, assumes that the notion of relevance is (usually) binary, one dimensional, and static, while the actual nature of relevance itself is very dynamic and situational.

The second problem is that the user behaviour as simulated by some popular metrics, such as average precision, generally may not accurately reflect actual user behaviour. In a TREC evaluation, a retrieval system usually returns a ranked list of 1000 documents for a particular topic (or query), and then a metric is employed to quantify the utility of the ranking into a score. In the case of average precision, simulated users are assumed to alter their browsing behaviour based on the part of the ranking that has not yet been seen, which cannot be done by real users [151]. Kelly [117] further notes that this indicates that some metrics are only appropriate to validate the performance of the system (a system-centric metric: *how good is the system in returning relevant documents?*), but not necessarily the usability of the system (a user-centric metric: *is the system usable?*). Moffat et al. [155] argue that the solutions to that problem are two, removing *recall* as a factor as for average precision, and incorporating user behaviour into search effectiveness metrics.

Voorhees [225] describes other issues potentially raised from the realisation of the Cranfield paradigm (especially in the TREC evaluation setting), such as differences in relevance judgements. Nevertheless, a test collection-based evaluation is stable when it comes to comparing multiple ranked lists of documents generated by different retrieval systems. Voorhees [225] compares several system rankings generated using different sets of relevance judgements. By repeating this experiment multiple times in different configurations (by varying four factors: topics, systems, metrics, and groups of assessors), the correlation coefficients across the generated rankings are high for all configurations, suggesting that this evaluation paradigm is stable. Zobel [247] and Sanderson and Zobel [185] demonstrated that significant differences in effectiveness metric scores observed on a test collection via the use of statistical tests (such as t-test and Wilcoxon) are reliable, meaning that the same results will likely continue to occur in other settings. Over the years, test collection-based evaluation has been popular among IR researchers, and contributing much to the development of modern search system, including Web search engine [52].

2.1.3 Search Task Classification

It has been widely recognised that an *information need* is what drives the user to perform search [31, 94, 157, 222]. The user translates their information need into a query, and then submits the query to a search system. Search results generated by the system in response to the query are intended to fulfil the user's information need. However, the notion of information need is rarely defined and still not clear [30, 190]. Borlund and Pharo [30] note that one way to operationalise the concept of information need is via the notion of *search task*, which is defined by Wildemuth et al. [235, p. 1134] as "goal-directed activities carried out using search systems."

In particular, categorising search tasks is a key aspect to the evaluation of search engine system. For example, different type of tasks leads to different effectiveness metrics [57] and different information-seeking behaviour [119, 155]. Kelly et al. [119] note that search tasks can be categorised according to type (such as, navigational and exploratory) and according to properties (such as, complexity).

Broder [31] introduces a taxonomy that groups Web search tasks into three classes. The first class is *navigational*, whereby the user wishes to find a particular site that they have in mind. For example, the user enters the query "unimelb library" to target The University of Melbourne library website[1]. Rose and Levinson [173] note some motivations for navigational queries, including that typing natural language queries via a search service is more convenient than directly typing the URL. Further, this task type is still connected

[1] `https://library.unimelb.edu.au/`

with the *known-item* search task in earlier TREC evaluation [116]. The "home page finding task" in the TREC 2001 Web Track is also a navigational task [79]. The second class is *informational*, whereby the user's intent is to seek for information on one or more web pages, for example, with the goal to gain knowledge about something. The third class is *transactional*, whereby the user wishes to reach a site with the goal to perform further interactions or transactions, such as online shopping or downloading resources. Broder [31] further used a set of queries drawn from `AltaVista.com` at that time, and found that the proportions of navigational, informational, and transactional queries are, respectively, 20%, 48%, and 30%. Rose and Levinson [173] provide a refinement of Broder's taxonomy by replacing the notion of "transactional" with a more general term, "resource", and by providing subcategories for both informational and resource queries.

Kelly et al. [119] argue that Broder's taxonomy is essential in the context of Web search at that time, but is less useful in the context of interactive IR because the categories are too broad. To develop search tasks, Kelly et al. [119] further used six types of cognitive processes from Anderson and Krathwohl's Taxonomy of Learning Objectives [14]. Moffat et al. [155] adapt three levels of this taxonomy to understand the relationship between task complexity and two quantities: the expected number of relevant documents and the expected number of queries. The first level is *Remember*, which is the lowest level in the hierarchy and involves fact-finding questions, such as "When was Gerard Salton born?" The second level is *Understand*, which involves "constructing meaning from oral, written, and graphic messages through interpreting, exemplifying, classifying, summarizing, inferring, comparing, and explaining" [119]. The third level is *Analyse*, which involves "breaking material into constituent parts, determining how the parts relate to one another and to an overall structure or purpose through differentiating, organizing, and attributing" [119]. Chen et al. [46] used Broder's taxonomy and two levels from Anderson and Krathwohl's taxonomy (*Remember* and *Understand*) as confounding factors when investigating the relationship between user satisfaction and effectiveness metric scores.

2.1.4 Fundamental Effectiveness Metrics

Saracevic [188] notes five requirements that need to be considered when a system is evaluated, including an IR system: (1) *a system*, that is, the test collection and the process or the retrieval algorithm being evaluated; (2) *criteria*, which is associated with the goal or objective of the system; (3) *metrics*, which quantify effectiveness as a single score, depending on the criteria being used; (4) *measuring instruments*, such as relevance judgements;

and (5) *a methodology* to conduct the evaluation and obtain the measurements. Here, we are particularly interested in *metrics*.

IR system evaluations mostly employ the notion of relevance as their main criterion. This is mainly due the widespread use of the test collection-based evaluation, such as the TREC evaluation framework, which is based on the Cranfield paradigm [225]. The use of relevance as a standard criterion for an IR system was envisioned by Kent et al. [120] in 1955, where the *effectiveness* of the system, a representation of how well the system in fulfilling an information need, can be thought of as being equivalent with the *performance* of the system, which is associated with the ability to find relevant documents. With this perspective, metrics are system-centric rather than user-centric, and are usually referred to as *effectiveness metrics*, *performance metrics* [117], or *system effectiveness metrics* [4]. In addition to relevance, Saracevic [188] describes other criteria that are more user-centric, such as utility, success, completeness, and cost.

Although we can separate the test collection and the metrics, they are closely related. When a new test collection is developed, it is often the case that a new metric is also proposed. The following section presents several traditional effectiveness metrics. Formally, a SERP of length N (with respect to query Q) can be represented using the following *relevance vector*:

$$\overrightarrow{r_Q} = \langle r_1, r_2, r_3, r_4, \ldots, r_N \rangle ,$$

where r_i is the relevance of the document at rank i. Here, N is the number of retrieved items. If the SERP is assumed to be in a *full-depth* form, N is the number of documents in the collection. In the context of many metrics, we assume that the relevance is binary, where $r_i = 1$ if the corresponding document in the ranking is relevant and $r_i = 0$ if otherwise. (Other metrics that can handle graded or continuous relevance are considered in Section 2.1.5.) A metric with a set of parameters $\mathbf{\Theta}$, $M(\overrightarrow{r}; \mathbf{\Theta})$, is defined as a function that takes as input a relevance vector $\overrightarrow{r}$ and returns a real value (often $0 \leq M(\overrightarrow{r}) \leq 1$) quantifying the effectiveness of the ranking for query Q.

Precision and Recall. Kent et al. [120] gives an early description of the combined use of Precision and Recall. At that time, they were employed to evaluate *Boolean search systems*, which return an unordered set of documents [120], providing the answers for the query. If N is the number of retrieved documents; R is the number of relevant documents in the collection; and R' is the number of retrieved documents that are relevant; then

Precision and Recall are computed as follows.

$$\begin{aligned}\text{Precision} &= \frac{R'}{N} && (2.1)\\ \text{Recall} &= \frac{R'}{R}. && (2.2)\end{aligned}$$

It is straightforward to compute Precision and Recall for a ranked list of length N:

$$\begin{aligned}\text{Precision}(\overrightarrow{r}) &= \frac{\sum_{i=1}^{N} r_i}{N}, && (2.3)\\ \text{Recall}(\overrightarrow{r}) &= \frac{\sum_{i=1}^{N} r_i}{R}. && (2.4)\end{aligned}$$

In Equation 2.4, the knowledge of R for each query is assumed to be known. In practice, R is usually not known and should be estimated [225].

There have been many proposals for unifying Precision and Recall [207, 221, 234]. Van Rijsbergen [221] described a way to combine Precision and Recall into a single metric, called the F metric,

$$\text{F}(\overrightarrow{r}) = \left(\alpha \cdot \left(\frac{1}{\text{Precision}(\overrightarrow{r})}\right) + (1-\alpha) \cdot \left(\frac{1}{\text{Recall}(\overrightarrow{r})}\right)\right)^{-1}, \tag{2.5}$$

and it is very common to use $\alpha = 0.5$, leading to the F1 metric:

$$\text{F1}(\overrightarrow{r}) = \frac{2 \cdot \text{Precision}(\overrightarrow{r}) \cdot \text{Recall}(\overrightarrow{r})}{\text{Precision}(\overrightarrow{r}) + \text{Recall}(\overrightarrow{r})}. \tag{2.6}$$

The traditional Precision and Recall serve as a basis for other effectiveness metrics used in the earlier TREC evaluations. Some of them will be introduced in this section.

Precision at K. The user modelled by the traditional Precision metric is assumed to examine all of the documents provided in the ranking. As an alternative, Precision can also be parameterised with an integer K, modelling a user who only examines the first K documents in the ranking. This is the idea of precision at K (Prec@K) computed as:

$$\text{Prec@K}(\overrightarrow{r}; K) = \frac{\sum_{i=1}^{K} r_i}{K}, \tag{2.7}$$

where $K \leq N$ is the parameter of the model. The choice of K usually depends on the characteristics of users and the search result presentation; $K = 10$ is common in the context of Web search [184]. Note that Prec@K will ignore all documents beyond rank K, relevant or not. For example, consider the following relevance vector of size 10:

$$\overrightarrow{r_{Q1}} = \langle 1, 0, 1, 1, 0, 1, 0, 1, 1, 1 \rangle \,.$$

The precision values at ranks 5 and 8 are:

$$\begin{aligned} \mathsf{Prec@K}(\overrightarrow{r_{Q1}}; 5) &= \frac{3}{5} = 0.600\,, \\ \mathsf{Prec@K}(\overrightarrow{r_{Q1}}; 8) &= \frac{5}{8} = 0.625\,. \end{aligned}$$

Average Precision. Average precision (AP) combines recall and precision for the evaluation of a ranked list of documents, and is defined as:

$$\mathsf{AP}(\overrightarrow{r}) = \frac{\sum_{r_i=1} \mathsf{Prec@K}(\overrightarrow{r}; i)}{R} \,. \tag{2.8}$$

With this definition, AP is a top-heavy metric, and has R as its normalisation factor. When only top-K documents are considered, a variant of AP, average precision at K (AP@K), is employed for the assessment of a truncated ranking at depth K:

$$\mathsf{AP@K}(\overrightarrow{r}; K) = \frac{\sum_{i=1}^{K} (\mathsf{Prec@K}(\overrightarrow{r}; i) \cdot r_i)}{R} \,, \tag{2.9}$$

where $K \leq N$, and R is assumed to be known. This version of AP is used by `trec_eval`, an evaluation software employed in the TREC evaluation[2], but with the denominator being replaced by $\hat{R}$, the estimated value of R. Section 2.1.5 will introduce an approach for the estimation of R.

When there are not enough judged documents to accurately estimate R, three alternative normalisation factors have emerged. Baeza-Yates and Ribeiro-Neto [23] replace R in Equation 2.9 with $\sum_{i=1}^{K} r_i$, the number of documents retrieved by the system. Baeza-Yates and Ribeiro-Neto further refer to this metric as the "average precision at seen relevant documents". Hawking et al. [80] introduce another version of AP@K, where R is replaced by K. This score of this metric is thus $0 \leq \mathsf{AP@K}(\overrightarrow{r}; K) \leq \min(R/K, 1)$. Voorhees and Harman [227] employ $\min(K, R)$ as the normalisation factor. This version of AP@K has been used in the TREC-7 Very Large Collection track [227]. As is noted by Craswell and Robertson [58], the latter variant of AP@K has the property that replacing any non-relevant document with a relevant one in the top-K position always increases the score.

[2] `https://trec.nist.gov/trec_eval/` and `https://github.com/usnistgov/trec_eval`

Average precision computes Prec@K at each ranking position where a relevant document is found, and subsequently calculates the average value. There are two notable differences between AP and Prec@K. First, AP depends on R, the total number of relevant documents for a particular query, while Prec@K does not use such information. Frei and Schäuble [66] note that it is not practical to obtain R from the whole document collection, especially when the size of the collection exceeds tens of megabytes. Second, AP assumes that the simulated users must scan all results in the ranking, while users simulated by Prec@K only proceed to depth K.

Hull [93] points out that the choice between AP and Prec@K should be determined according to the user's search goal or the complexity of the search task. For example, if the user is only interested in relevant documents presented in the first page, Prec@K would be more suitable than AP. But, if the user's goal is to find all documents about a particular technology in the collection, AP would be the best one since Recall is important in this case. In contrast, Zobel et al. [248] argue that AP (and thus, all metrics that depend on Recall) lacks connection with the criteria of user satisfaction. They further argue that "high-recall" users (such as, patent, legal, or medical search users) stop inspecting a SERP based on their feeling that no further relevant items will appear. This should depend solely on what they have seen, and thus this is not related to Recall. Long before that, Cooper [55] already noted that unexamined documents should not contribute to what users have experienced (either satisfaction or frustration) at the end of search.

Robertson [169] suggests that the user modelled by AP does not stop at non-relevant documents, and is equally likely to stop at any relevant document position in the ranking (each one with a stopping probability equals to $1/R$). That is, the user first randomly chooses one of the relevant documents in the ranking as their stopping position, and then inspect the ranking from the top through to that chosen position. Robertson [169] further argues that this user model is plausible under a certain condition.

Webber et al. [232] study the predictive power (that is, the extent to which the system ranking generated from a set of experimental topics reliably predict the ranking from the other set of topics) of simple metrics, such as Prec@10, alongside more complex metrics, such as AP. The experiments were carried out using top 75% of TREC 2004 Terabyte and TREC 8 AdHoc Tracks systems based on AP score. The experiment results show that the more complex metrics are as good at predicting Prec@10 as Prec@10 is at predicting itself, suggesting that reporting evaluation results from Prec@10 together with those from the more complex metrics is redundant.

R-Precision. The TREC-2 conference introduced the use of R-precision, which is Prec@K with $K = R$, the total number of relevant document for a particular topic [73]. Aslam et al. [16] point out that AP and R-precision can be used to approximate the area under a line, *recall-precision graph*. In early IR papers, such graphs were widely used to report the effectiveness of IR systems [222].

Reciprocal Rank. Reciprocal rank (RR) was proposed to evaluate IR systems where the goal is to find a single relevant document [184], such as navigational or known-item tasks [116, 226]. Reciprocal rank (RR) at depth K is computed as follows:

$$\mathsf{RR@K}(\vec{r}; K) = \begin{cases} 1/H & i \leq K \\ 0 & i > K\,. \end{cases} \tag{2.10}$$

where $H \leq K$ is the first ranking position at which relevant document appears. For example, consider two further SERPs:

$$\begin{aligned} \overrightarrow{r_{Q2}} &= \langle 0,0,0,1,0,0,0,1,0,1 \rangle\,, \\ \overrightarrow{r_{Q3}} &= \langle 1,0,1,1,0,1,1,1,1,1 \rangle\,. \end{aligned}$$

The RR scores for $\overrightarrow{r_{Q2}}$ and $\overrightarrow{r_{Q3}}$ at depth 10 are:

$$\begin{aligned} \mathsf{RR@K}(\overrightarrow{r_{Q2}}; 10) &= \frac{1}{4} = 0.250\,, \\ \mathsf{RR@K}(\overrightarrow{r_{Q3}}; 10) &= \frac{1}{1} = 1.000\,. \end{aligned}$$

Webber et al. [232] demonstrate that RR is a poor self-predictor, suggesting that evaluation reports from RR are somewhat unreliable, and is thus unnecessary if evaluation results from more complex metrics, such as AP, are also reported.

2.1.5 Relaxations of the Assumptions

The effectiveness metrics, Precision, Recall, Prec@K, average precision, R-precision, and reciprocal rank mostly rely on several assumptions about the set of relevance judgements, as follows [187]:

1. topical similarity – a document is relevant to a particular query if both of them are from the same topic;

2. binary relevance – the document is either relevant or non-relevant with respect to a particular query;
3. independent – the relevance of a particular document is not affected by the information from other documents;
4. static – the relevance of a document does not change over time;
5. consistent – the relevance judgements are consistent across all assessors; and
6. completeness – the judgements for a topic are complete (no missing judgements).

However, as is noted by Kelly [117], "the published research about how users make relevance assessments and the actual metrics that researchers employ to collect relevance assessments are not very aligned." The following issues describe what the actual users think about relevance [44]:

1. graded relevance – users naturally think about documents in a graded relevance system. For example, a document might be *partially* relevant under a particular query;
2. incomplete – unjudged documents in a test collection do exist since it is impractical to judge all documents in the collection [66]; and
3. dependence and diversity – the user might ignore repetition of the same relevant document in a ranking, and prefer a wide range of aspects (that is, diversity) in the presentation of relevance documents.

Gain Mapping Functions. All fundamental metrics that have been described in Section 2.1.4 are devised based on the binary relevance assumption, that is, $r_i \in \{0, 1\}$. In this case, the *gain* scores are just zero (non-relevant) or one (relevant). However, graded relevance judgements are also available for laboratory test collections [198, 224]. To represent different relevance levels, one typical approach is to use ascending integers starting with zero [102, 220]. Let $r_i \in \{0, 1, 2, \ldots, r_{max}\}$ denotes an ordinal relevance label derived from a document at rank i, with the spectrum ranging from non-relevant document ($r_i = 0$) to highly relevant one ($r_i = r_{max}$). A *gain mapping function*, $g(r_i)$, is required to convert r_i into a numeric gain score, with $0 \leq g(r_i) \leq 1$. This function can be regarded as assigning "relevance weights at different relevance levels" [102, p .431].

Several gain mapping functions have been proposed in the past. Lu et al. [141] describes a *binary* mapping function, which maps ordinal relevance relevance label to a binary value:

$$g(r_i) = \begin{cases} 1 & i \geq \theta \\ 0 & i < \theta \,, \end{cases} \tag{2.11}$$

where θ is a threshold value for binarising the ordinal relevance. For example, Thomas et al. [213] used several threshold values ($\theta \in \{1, 2\}$) when computing RR scores using 4-level graded relevance judgements ($r_i \in 0, 1, 2, 3$). Lu et al. [141] also describe a *linear* mapping function that provides equal weights for all relevance grades:

$$g(r_i) = \frac{r_i}{r_{max}} \,. \tag{2.12}$$

For example, Zhang et al. [242] use this linear function to map 5-level usefulness labels into $g(r_i) \in \{0.00, 0.25, 0.50, 0.75, 1.00\}$. Finally, an *exponential* gain mapping function can be used to emphasize highly-relevant documents:

$$g(r_i) = \frac{2^{r_i} - 1}{2^{r_{max}} - 1} \,. \tag{2.13}$$

This function is based on the *unbounded* exponential gain mapping function described by Burges et al. [35]. Recently, Turpin et al. [220] demonstrated that the linear mapping scheme is closer to the perception of user, compared to the exponential mapping scheme.

Graded Relevance Metrics. Järvelin and Kekäläinen [101] address the issue of graded relevance by proposing *discounted cumulative gain* (DCG), where the relevance value of a document is discounted based on its ranking position, and the discount function itself is a monotonically decreasing with the ranking position, suggesting that simulated users value relevant documents in higher ranking positions, compared to those in lower ranks.

$$\mathsf{DCG@K}(\overrightarrow{r}; K, b) = \sum_{i=1}^{b-1} g(r_i) + \sum_{i=b}^{K} \frac{g(r_i)}{\log_b(i)} \,, \tag{2.14}$$

where $1/\log_b(i)$ is the discount function, K is the evaluation depth, and b is the logarithm base. The higher the base, the deeper the simulated user inspects the ranking. For example, when $b = 2$, the user examines at least top-2 items in the ranking.

As an illustration, consider the following SERP of size 5 with 3-level graded relevance,

$r_i \in \{0, 1, 2\}$, where 0 is for non-relevant, 1 for partially relevant, and 2 for fully relevant.

$$\overrightarrow{r_{Q4}} = \langle 1, 0, 2, 1, 0 \rangle \,.$$

If the following gain mapping function is employed: $g(r_i) = r_i / r_{max}$, the resultant *gain vector* is $\langle 0.5, 0.0, 1.0, 0.5, 0.0 \rangle$. The $\mathsf{DCG@K}(\overrightarrow{r_{Q4}}; 5, 2)$ is computed as follows:

$$\begin{aligned} \mathsf{DCG@K}(\overrightarrow{r_{Q4}}; 5, 2) &= 1.0 \times 0.5 + \frac{1}{1.00} \times 0.0 + \frac{1}{1.58} \times 1.0 + \frac{1}{2.00} \times 0.5 + \frac{1}{2.32} \times 0.0 \\ &= 1.383 \,. \end{aligned}$$

Note that the value of $\mathsf{DCG@K}(\overrightarrow{r}; K, b)$ can be greater than 1. In 2008, when addressing an extension of DCG for session-based offline evaluation, Järvelin et al. [103] describe another version of discounted cumulative gain: $\mathsf{DCG@K}(\overrightarrow{r}; K, b) = \sum_{i=1}^{K} g(r_i) / (1 + \log_b i)$.

Burges et al. [35] describe an alternative version of DCG with a different discount computation that avoids the parameter b:

$$\mathsf{DCG@K}(\overrightarrow{r}; K) = \sum_{i=1}^{K} \frac{g(r_i)}{\log_2(i+1)} \,. \tag{2.15}$$

The discount function $1 / \log_2(i+1)$ is also sometimes called the *Microsoft* version.

To limit the value of DCG to 1 for a given depth K, Moffat and Zobel [151] proposed *scaled discounted cumulative gain* (SDCG), where the value of DCG@K of a given ranking is divided by the value of DCG@K when all documents in the ranking are fully relevant:

$$\mathsf{SDCG@K}(\overrightarrow{r}; K) = \left(\sum_{i=1}^{K} \frac{g(r_i)}{\log_2(i+1)} \right) \Big/ \left(\sum_{i=1}^{K} \frac{1}{\log_2(i+1)} \right) \,. \tag{2.16}$$

Järvelin and Kekäläinen [102] propose a second gain-based metric which accounts for normalisation factor and the score of DCG can be limited to 1. Similar to AP, it is now relative to the best that can be attained by a ranking of that length. Their proposed metric is called *normalised discounted cumulative gain* (NDCG). The idea is to compute the ratio between the DCG score of the ranking being evaluated and the DCG score of the corresponding *ideal* ranking. Hence, the score of NDCG is between 0 and 1:

$$\mathsf{NDCG@K}(\overrightarrow{r}; K, b) = \frac{\mathsf{DCG@K}(\overrightarrow{r}; K, b)}{\mathsf{DCG@K}(\overrightarrow{r^*}; K, b)} \,, \tag{2.17}$$

where $\overrightarrow{r^*}$ is the ideal ranking that can be generated given the knowledge of the relevance grades for all documents, whether or not they were included in the original $\overrightarrow{r}$. By assuming that the ranking being evaluated $\overrightarrow{r}$ contains all relevant documents for a particular topic, we can produce the ideal ranking $\overrightarrow{r^*}$ by sorting the vector $\overrightarrow{r}$ in decreasing order. For example, by assuming that there are only one fully relevant document and two partially relevant documents in the collection, the ideal ranking of $\overrightarrow{r_{Q4}}$ (truncated at depth 5) is

$$\overrightarrow{r^*_{Q4}} = \langle 2, 1, 1, 0, 0 \rangle \,.$$

Thus, $\mathsf{DCG@K}(\overrightarrow{r^*_{Q4}}; 5, 2) = 1.816$; and

$$\mathsf{NDCG@K}(\overrightarrow{r_{Q4}}; 5, 2) = \frac{1.383}{1.816} = 0.762 \,.$$

Q-Measure (QM), proposed by Sakai [175, 176], is also an effectiveness metric that can deal with graded relevance. Let $isrel(i)$ be a function that returns one if $r_i > 0$ and zero otherwise; $cg(i) = \sum_{j=1}^{i} g(r_j)$ denotes a cumulative gain from rank zero to rank i; and $cg_I(i) = \sum_{j=1}^{i} g(r^*_j)$ denotes a cumulative gain from rank zero to rank i in an *ideal* ranking. Q-Measure for the evaluation of a full-depth ranking is then defined as follows:

$$\mathsf{QM}(\overrightarrow{r}; \beta) = \frac{1}{R} \cdot \sum_{i=1}^{N} \left\{ isrel(i) \cdot \left(\frac{\beta \cdot cg(i) + \sum_{j=1}^{i} isrel(j)}{\beta \cdot cg_I(i) + i} \right) \right\} \,, \tag{2.18}$$

where $\beta \geq 0$ is the persistence parameter of QM. The larger the value of β, the smaller the discount for relevant documents at lower ranking positions. Sakai [179] further suggests to use $\beta = 1$ or $\beta = 10$. In the case of binary relevance, $cg_I(i) = \min(i, R)$, and thus QM for binary relevance (denoted as QM′) is computed as:

$$\mathsf{QM}'(\overrightarrow{r}; \beta) = \frac{1}{R} \cdot \sum_{i=1}^{N} \left\{ r_i \cdot \left(\frac{(1+\beta) \cdot \sum_{j=1}^{i} r_j}{\beta \cdot \min(i, R) + i} \right) \right\} \,. \tag{2.19}$$

As is also the case for AP, Q-Measure requires the knowledge of Recall. Indeed, binary relevance-based QM′ reduces to AP when $\beta = 0$. Finally, Sakai and Zeng [182] also introduce Q-Measure at K, denoted by $\mathsf{QM@K}(\overrightarrow{r}; \beta, K)$, where the normalisation factor R in Equation 2.18 is replaced by the new term, $\min(K, R)$; and the summation runs from 1 to $K \leq N$.

In 2008, Moffat and Zobel [151] propose a metric, *rank-biased precision* (RBP), which employs a geometric discount function and also deals with graded relevance. In the context of RBP, the characteristic of a user is modelled via the *persistence* parameter, ϕ, representing the likelihood that the user progresses from rank i to rank $i+1$ in the SERP. A user with high ϕ value is said to be "patient" since they are willing to scan many documents in the ranking, while users with low ϕ are impatient, and focus primarily on the documents at top-ranking positions. RBP is computed as follows:

$$\mathsf{RBP}(\overrightarrow{r};\phi) = (1-\phi)\cdot\sum_{i=1}^{N}\Big(g(r_i)\cdot\phi^{i-1}\Big)\,. \tag{2.20}$$

Here, the ranking is assumed to be in a full-depth form of length N. Later, Chapelle et al. [44] and Zhang et al. [244] compared the discount function embodied in RBP and DCG with the observed examination probability estimated from clickthrough logs, and find that the discount function of RBP (with $\phi \approx 0.7$) is closer to the observed examination probability, compared to that of DCG.

The idea to incorporate user persistence into a metric is not new. In 1968, Cooper [53] proposed a similar idea, and stated that "most measures do not take into account a crucial variable: the amount of material relevant to [the user's] query which the user actually needs ... the importance of including user needs as a variable in a performance measure seems to have been largely overlooked". This idea inspired several modern effectiveness metrics, including DCG and RBP.

In 2009, Chapelle et al. [44] introduce *expected reciprocal rank* (ERR) that works for graded relevance. This metric is computed as follows:

$$\mathsf{ERR@K}(\overrightarrow{r};K) = \sum_{i=1}^{K}\frac{1}{i}\cdot\prod_{j=1}^{i-1}\left(1-g(r_j)\right)\cdot g(r_i)\,, \tag{2.21}$$

where K is the evaluation depth. In the original proposal, $g(r_i)$ can also be thought of as the probability that the user is satisfied with the document at rank i [44]. Chapelle et al. further suggest to use the following gain mapping function:

$$g(r_i) = \frac{2^{r_i}-1}{2^{r_{max}}}\,, \qquad r_i \in \{0,\ldots,r_{max}\}\,. \tag{2.22}$$

Yilmaz et al. [240] incorporate click model into a graded relevance metric, *expected browsing utility* (EBU). This metric is defined as follows:

$$\mathsf{EBU@K}(\overrightarrow{r};K,\mathbf{E},\mathbf{C}) = \sum_{i=1}^{K}P(E_i)\cdot P(C_i \mid \mathit{Rel}=r_i)\cdot g(r_i)\,,$$

where $P(C_i \mid \mathit{Rel}=r_i)$ is the probability that the user clicks at rank i given the relevance of the document, and $P(E_i)$ is the probability of the user examining the document at rank i. These two quantities need to be estimated from user observation data, such as click logs.

Unjudged Documents. In the Cranfield paradigm, it is assumed that each topic has a complete list of relevance judgements. In other words, each document in the collection has been judged with respect to a particular query. However, this assumption becomes impractical once the size of the test collection gets very large to the extent of tens of megabytes [66]. The TREC evaluation, one of the evaluation models that adopts the Cranfield paradigm, approximates the true number of relevant documents for a particular query (R) using a *pooling* technique [225]. The organiser of TREC evaluation, NIST[3], provides a set of queries (or topics) and a set of documents to the participants. Each participant then runs their retrieval algorithm for all given topics against all documents in the collection, generating a ranked list of usually (but not always) 1000 documents. Using the submitted runs, NIST then forms a *pool* containing all rankings from all systems initiated from the participants, typically containing the top 100 documents. Once the pool has been formed, all of the documents in it are judged; documents outside the pool are not [73]. In other words, TREC approximates R, the number of relevant documents in the collection for a particular query, with the number of relevant documents appearing in the top 100 of all runs generated by the participating systems when evaluating that query (denoted by $\hat{R}$).

The pooling technique employed by TREC introduces the third category of document: the *unjudged documents*, those outside the pool that never get judged [184]. The presence of such documents can degrade the effectiveness scores for systems that did not contribute to the pools, since the rankings generated by the non-contributing systems can have "highly ranked unjudged documents that are assumed to be not relevant" [225]. Buckley and Voorhees [33] demonstrate that several traditional effectiveness metrics, such as AP, R-precision, and Prec@10, are not robust when the relevance judgements are incomplete.

Several approaches has been proposed to deal with unjudged documents. A simple approach is just to ignore unjudged documents when computing effectiveness metrics [82] or to regard them as being non-relevant [225].

Buckley and Voorhees [33] conjecture that the size of pools could possibly be reduced relative to the size of collections when the number of documents in the collection grows and no new judging is conducted. As the number of unjudged documents grows, there is

[3] See `http://trec.nist.gov`

a need to develop a new effectiveness metric that allows us to evaluate a ranking containing unjudged documents. One such metric is *binary preference* (BPref) [33]. Let $\overrightarrow{r'}$ be a *condensed* ranking constructed by removing unjudged documents in the original ranking $\overrightarrow{r'}$. Then, BPref is computed as follows:

$$\mathsf{BPref}(\overrightarrow{r'}) = \frac{1}{R} \sum_{i:r'_i=1} \left(1 - \frac{min(R, |\{j \mid r'_j = 0 \text{ and } 1 \leq j < i\}|)}{min(R, NR)}\right) , \tag{2.23}$$

where R is the number of identified relevant documents for a particular topic, and NR is the number of non-relevant documents. This metric visits each position, i, where the relevant document appears, and then count the number of non-relevant documents that are ranked higher than i. This serves as a basis for binary preference, that approximates the relevance score of a particular document. When the relevance judgements are complete. Buckley and Voorhees [33] further argue that BPref is consistent with AP and is more robust than R-precision and Prec@10. Sakai [178] shows inter-relationship between BPref, AP, and QM, and further argues that BPref has some drawbacks, including that it is less top-heavy than AP.

A similar idea to BPref can also be applied to AP. Yilmaz and Aslam [239] propose *Induced Average Precision* (indAP) at which unjudged documents are removed from the ranking:

$$\mathsf{indAP}(\overrightarrow{r'}) = \frac{1}{R} \sum_{i:r'_i=1} \left(1 - \frac{|\{j \mid r'_j = 0 \text{ and } 1 \leq j < i\}|}{i}\right) . \tag{2.24}$$

In the presence of unjudged documents, Aslam et al. [17] and Yilmaz and Aslam [239] also show that the *expected precision* at depth K can be computed by uniformly sampling documents at top-K rank positions. The average of expected precision values across all relevant document ranks forms a metric, called *Inferred Average Precision* (infAP) [239].

Büttcher et al. [36] employ a text classification method to determine whether any unjudged document found in the ranking is relevant for a given query or not. Moffat and Zobel [151] suggest reporting of *residuals*, uncertainty scores that represent the imprecision of a weighted-precision metric. Suppose the upper bound (lower bound) value of a particular weighted-precision metric can be computed by assuming that all unjudged documents are fully-relevant (non-relevant); the residual is the difference between the upper and lower bounds values.

Dependence and Diversity. The user might prefer novelty and diversity in the presentation of search results. A retrieval system that promotes novelty is expected to remove all duplicates of relevant documents in the SERP [184]. Meanwhile, the notion of diversity rewards an IR system that is able to present a wide range of aspects of relevant information in the SERP [184]. The TREC Novelty Track [74] and the TREC Interactive Track [161] were organised to address novelty and diversity in search results, as well as spark a new research direction towards IR evaluation that is sensitive to novelty and diversity.

The effectiveness metrics described to this point, such as AP, Prec@K, RBP, and DCG were developed under the assumption that the relevance of a document is not influenced by the any other document. In other words, the assessors are assumed to "forget" all documents that they have already seen when they are assessing a particular document. It has been shown that a simple treatment of duplications when we calculate effectiveness metrics may give unexpected results. Bernstein and Zobel [28] note that "near-duplicate" documents decrease AP score by 20.2% when they are treated as being non-relevant; and increase AP score by 16.0% when we simply ignore them. Hence, a new effectiveness metric that incorporates novelty and diversity is needed.

Clarke et al. [48] propose a metric called α-NDCG, a modification of NDCG; and which rewards diversity and novelty in the ranking. With NDCG, the relevance of a document in a particular rank position is static and atomic, and denoted as r_i. Meanwhile, in the context of α-NDCG, the relevance of a document, r_i, can be thought of as the composition of the relevances across all nuggets that form the document. In this case, the gain mapping function $g(r_i)$ is then replaced with the following relevance function, $rel(i)$ [184]:

$$rel(i) = \sum_{k=1}^{m} J(d_i, k) \cdot (1-\alpha)^{c_{k,i-1}} , \tag{2.25}$$

where m is the number of nuggets (aspects or subtopics) that is relevant to a particular topic; $J(d_i, k) = 1$ if the document d_i contains the k-th nugget or subtopic (according to the assessor), and 0 if otherwise; and

$$c_{k,i-1} = \sum_{j=1}^{i-1} J(d_j, k) ,$$

that represents the number of documents that are ranked higher than d_j and contain the k-th nugget (and hence, the relevance of a particular document depends on the relevances of other documents that have been seen by simulated users). In Equation 2.25, α is an empirical value representing the likelihood that a user is satisfied with the judged relevant document; or alternatively $1 - \alpha$ be the tolerance of a particular user on the duplicates [49]. That is, the simulated user is assumed to be unsatisfied with a relevant document, if it has been previously seen. Note that when $m = 1$ and $\alpha = 0$, α-NDCG reduces to the original DCG.

2.1.6 The Problem of Recall and The Virtue of Precision

The Problem of Recall. As we previously noted, Recall is defined as the ratio between the number of relevant documents retrieved and the total number of relevant documents in the collection (R) for a particular query. This simple metric is, however, problematic; and in general all effectiveness metrics that depend on the knowledge of R, such as AP, R-precision and BPref, share one key shortcoming.

That shortcoming is that it is impractical to know R once the size of a test collection gets large [66]. Furthermore, it is almost impossible for an assessor to judge millions, let alone tens or hundreds of millions, of documents in the collection for a single query. The TREC evaluation conference has been using the *pooling* approach to estimate the knowledge of R for each query in the collection [225]. However, Zobel [247] argues that the pooling approach is not suitable for measuring a high-recall retrieval system since there is still a possibility that many relevant documents in the collection remain undetected. Zobel [247] further conducted an experiment on a TREC data to support the argument about the reliability of the pooling approach in estimating R, and concluded that while the majority of the relevant documents have been successfully identified in the TREC experiments, many more remain unidentified.

Given the relevance vector of a ranking and a reasonable value of K, Prec@K can be exactly computed since it only depends on what was retrieved [188]. As a consequence, the behaviour modelled by Precision reflects how users interact with what is presented by the system, even though they most likely did not look at all results in the ranking. On the other hand, the behaviour of users modelled by Recall is at odds with observed user behaviour. It does not make sense to assume that users clairvoyantly know how many relevant documents were not retrieved by the system. As stated by Saracevic [188], “Recall is a metaphysical metric: how does one know what is missed when one does not know that it is missed?” In addition, Cooper [55, p. 95] notes that “a document which the system user has not been shown in any form ... does that user neither harm nor good.”

The next problem with Recall is that it does not have a clear criteria of user satisfaction.

Saracevic [188] notes that "the assessment of performance and value is related to the question of how to provide a prospective user with useful information." Saracevic and Kantor [189] studies users working with online database search system, and found that the user satisfaction is correlated with Precision, but not with Recall. On the other hand, Su [205] reported the opposite results, concluding that Recall has a higher correlation with system success, compared to Precision. However, the reason why Su [205] finds that Recall is more important than Precision to users is that the experimental participants were doctoral students and academic faculty members who are experts in using a database search system, in particular for the tasks related to dissertation or research proposal; and that the result is an unranked set of documents (Boolean retrieval). In a later study, Su [206] used the data sampled from four Web search engines: `AltaVista.com`, `Lycos.com`, `Infoseek.com`, and `Excite.com`, and employed participants from undergraduate students. The search goals of the participants are related to class assignments, personal interests, travel, and jobs. Su [206] then conclude that Precision is more important to users, compared to Recall.

Buckley and Voorhees [34] note that one of the debates in IR evaluation is about the usefulness of Recall outside specific domains such as *patent search* domain. In another view, the concept of recall is not applicable for navigational queries (or other types of queries whereby users only expect a few relevant documents) since users are surely not interested in all remaining relevant documents once they have found the answer that they expected [151]. For example, when a user looks for the URL of person X's homepage via a Web search engine, they will most likely submit "X homepage" as a query. In the end, users are only interested in the first relevant document (or the first answer) they found in the ranked list of documents, even though the remaining part of the ranking may contain many relevant documents.

Zobel et al. [248] provide thorough argumentations of why Recall is not a plausible metric of the effectiveness of a ranking, even though some of their explanations are still not substantiated. They argue that Recall is not related to several concepts influencing user experience, such as *cardinality* (that is, the number of relevant items inspected so far) and *coverage* (that is, the fraction of questions successfully answered from the ranking being inspected), and thus has no relationship with search satisfaction itself. They further contend that, while Recall may be of interest as a metric for medical, patent, and legal search engines, the user should still have a criterion causing them to stop inspecting the ranking. In this context, this criterion might be best described as a feeling that they have found all relevant documents after they have gone through a number of search activities (likely involving multiple queries, or even other search engines) all failing to provide further relevant items. For example, after finding the last relevant document at rank i, the user then inspects say 100 documents beyond rank i, and then finds that none of them are relevant. It is only then at this point, that the user may have confidence that all relevant items has been examined. That is, the indication that a *total recall* may have been achieved is still based on what have been observed, but not on documents that are not retrieved by the system. Hence, this should not have connection with Recall. However, if Recall is a requirement, it is not the Recall of a query that is important, but Recall of the session.

Precision-Based Effectiveness Metrics. In contrast to Recall, Precision is easier to calculate since it only needs "what was retrieved" (the ranking), and does not need "what was missed" [188]. Moreover, as noted previously, it has been shown that Precision has a correlation with users' satisfaction, which has been the main goal of IR system [5, 189, 206]. This is in agreement with what have been posited by Cooper [55] more than 40 years ago:

> Instead of attempting to estimate recall in spite of all the difficulties what should have been done was to find a way to overcome the deficiencies of the precision metric without bringing a second metric into the picture.

Some efforts to realise Cooper's vision [55] have been made. Moffat et al. [155] describe a special family of *weighted-precision* metrics, whereby a metric from this family is defined as a summation over the multiplication between the gain and the discount function. Average precision (AP), Prec@K, SDCG, DCG, RBP, RR, and ERR are examples from this family of metrics. Further, a weighted-precision metric has received attention since it can be interpreted as having an association with a *user model* [151]. This issue is considered in detail in Section 2.3.

2.2 User Search Behaviour

Knowing that incorporating user behaviour into search effectiveness metrics is critical, understanding user search behaviour is thus one of the essential steps that need to be taken in order to devise effectiveness metrics that reflect user experience. This process involves analysing search interactions either from operational search engines or from lab-based user studies. This section describes research that is concerned with understanding how users interact with SERPs. Section 2.2.1 considers the use of interaction logs as resources for exploring general search interaction patterns, while sub-section 2.2.2 specifically focuses on the use of logs for investigating in what order users scan the SERP (that is, browsing behaviour). Finally, sub-section 2.2.3 discusses several issues related to stopping behaviour.

2.2.1 Interaction Log Study

Interaction logs serve as important resources for improving the effectiveness of SERPs as well as the whole quality of an IR system. In the context of Web search, interaction logs have been used for developing and calibrating search effectiveness metrics [20, 86, 195]; improving the presentation of search results [231]; tuning the ranking algorithms [2, 109]; and increasing the accessibility and user-friendliness of the system either via query suggestion [38] or query auto-completion [127].

Web Search Logs. Silverstein et al. [192] carried out an analysis using a sample of query logs containing around 1 billion search requests collected from `AltaVista.com`, one of the major Web search engines in late 1990s. Silverstein et al. [192] mostly explored query-based patterns in the sessions, such as the composition of terms and operators per query, the association across query terms in the logs, the characteristics of popular queries, and the patterns of how users reformulate their initial queries in the sessions. They find that typical Web search users submit short keywords as their queries; and that users most likely only inspect the first page of the paginated SERPs. Prior to that, Jansen et al. [96] had carried out studies on a smaller interaction log containing 51,473 queries from another Web search engine, `Excite.com`. They also share the same findings in terms of the number of viewed pages in the Web search results.

In the following years, work on query logs [96, 192] was undertaken by Lempel and Moran [130] and Spink et al. [202], covering a larger volume of data as well as a new perspective on using the data. Spink et al. [202] analysed a log of approximately 1 million Web queries from `Excite.com`. Their findings reveal the same characteristics as the earlier work: Web search users typically submit only a few query terms, inspect Web pages, and almost never use additional features for advanced search. Moreover, Spink et al. used a more principled methodology in collecting and grouping a sample of Web queries, gaining more insights about what kind of topics people usually ask through Web search engines. Lempel and Moran [130] used another sample of query logs from `AltaVista.com` to observe the distribution of query popularities and to study the number of requests for individual SERPs, mainly for addressing the task of caching search results. One of their notable findings is that the popularity of topics and the number of Web page requests follow a power-law distribution, except for the most popular topics and the most popular Web pages. In addition to `AltaVista.com` and `Excite.com`, a sample of query logs from other Web search engines, such as `BWIE`, `AlltheWeb.com`, and `Fireball`, had been explored by Cacheda and Viña [37], Hölscher and Strube [88] and Spink et al. [203]. Jansen and Spink [95] give an overview across all of those search engines. They reported characteristics and trends, including session length, the number of terms in the queries, operator usage, and results page viewed. One of the findings is that approximately half of the Web search users submitted more than one query to address a single information need (53% on `AltaVista.com` data collected in 2002), suggesting that Web search evaluation should treat a session as a single unit with respect to a single information need.

Some authors explore temporal aspects of Web query logs, observing how the query- or session-based characteristics change over time [27, 243]. Beitzel et al. [27] carried out temporal and topical analysis on query logs containing billions of Web queries initiated from `AOL` Web searches, reporting query traffic for a specific period, query categories, and the tendency of how different topical categories shift at different times. For example, they find that "personal finance"-related queries are popular between 7 AM and 10 AM, while "music" queries are less popular at the same period of time. In general they observed that some queries with certain topical categories can span both short (a few hours) and long (several weeks or months) periods of time.

Zhang and Moffat [243] performed analysis on MSN interaction logs, containing not only a collection of query terms but also chronologically-ordered clickthrough data, collected during May 2006. While they confirmed some of the findings from past studies, such as the frequent use of short Web queries, they also reported other observations, including that "+1" clickthrough jump is very common suggesting that users tended to click in a sequential manner; that the distribution of clickthrough across ranking positions is top-weighted; that users are reluctant to click "next page"; that sessions with only one query are the most common; that clickthrough volumes peak between 11 AM and 12 AM every day; and that the time between any two consecutive clickthroughs is short. Smucker and Clarke [195] used interaction logs from commercial Web search engines to estimate the distribution of search duration between the initial query and the last click action. This information is then incorporated into a time-based discount function for an offline search effectiveness metric. Recently, Azzopardi et al. [20] employ observation data from interaction logs to meta-evaluate how accurate a user model is, by means of its closeness to observed behaviour.

Domain-Specific Search Logs. While the previous interaction log studies have focused on the Web search domain, several studies have explored interaction logs from domain-specific search engines, such as music and job search services. Recently, Chandar et al. [43] employed a mixed method to develop search effectiveness metrics for music search. This method involves conducting user interviews to gain insights about their expectations, and using interaction logs to verify the information obtained from the interviews. Chandar et al. were particularly interested in understanding characteristics of user behaviour that can be used as sensitive quality metrics. In the context of music search (`Spotify.com`), Chandar et al. found that some behaviours, such as "add to playlist", "save to collection", and "follow playlist", indicate search success, and thus these behaviours should be incorporated into an evaluation metrics for music search engine.

In the context of job search, Jansen et al. [97] carried out one of the earliest studies, but with Web search engine data (`Excite.com`) that had been filtered to extract queries that have relationship with job topics. Their studies suggest that job search users usually submit only one query before they end the session; and that most of the job-related queries contain of three terms. Kudlyak and Faberman [128] used the data from `SnagAJob.com`, an online job search engine, to understand the relationship between the strength of labor market and job search effort (measured in the number of applications). They found evidence for an *income effect*, suggesting that the behaviour of job seekers is influenced by the power of the labor market. For example, people with poor job-finding prospects tended to make more job applications to increase the chance and have a longer job seeking duration. Spina et al. [200] compared the behaviour of users from among job, talent, and Web search engines. Job and talent search data was collected from `Seek.com`, while Web search logs were a sample of `Yandex.ru` interaction logs. They suggest that Web search users might have different SERP examination behaviours than job and talent search users. More recently, Mansouri et al. [144] collected $500,000$ Web queries, based on job issues, from `Parsijoo.ir`, a Persian search engine, and found that job search intensities peak in the beginning of the week, and then slowly decrease until the end of the week.

2.2.2 User Browsing Behaviour

Desktop-Based Search. Klöckner et al. [123] addressed a question: *in what order do users scan the entries in a SERP?* They propose two classes of search strategies: (1) *depth-first* strategy, where users scan the entries one-by-one from top to bottom and immediately decide whether to open the document, and (2) *breadth-first* strategy, where users first examine all entries from top to bottom, before then revisiting the most enticing one and clicking the entry link to open the document. To support their arguments, they recorded eye movements and mouse clicks of 41 subjects who were given 10 minutes to use a Google search engine to obtain a particular information. In this experiment, they found that 65% of subjects used a depth-first strategy, 15% used a fully breadth-first strategy, and the remaining (20%) used a partially breadth-first strategy (*mixed* strategy). The latter strategy refers to users who scan forward a few entries before deciding to revisit and open the previously seen entries.

Teevan et al. [209] observed that when the user searches for specific information, such as an email address or an office number, they often perform a directed search (visiting someone's homepage), instead of a keyword search (submitting the query "scott morrison, email address" into a search engine). Teevan et al. then uncovered two types of search strategies: (1) *orienteering*, and (2) *teleporting*. Users who perform *orienteering* tend to follow a series of small steps to find a particular information, while those who perform *teleporting* tend to follow a direct way to find the information. For example, to find an answer for the question "what is the office number of Professor X?", an *orienteering* user will likely visit Professor X's homepage and search an office number on that page, while a *teleporting* user will go to a search engine website and enter "office number Professor X" into the input box, expecting that the search engine will give the answer at the top of the result page. Teevan et al. [209] argue that *orienteering* helps users to decrease the cognitive burden of search activities by maintaining their sense of location, since they usually know where to find the target, but do not know what the exact target is. Hence, their study provides insights to accommodate this search behaviour into the development of search engine interfaces.

Kim et al. [122] adopted the search strategy proposed by Klöckner et al. [123] in their experiments to see whether they vary as to the size of the screen (1280 × 1024-pixel versus 320 × 480-pixel screen, or large versus small size respectively). The results showed that the screen size does not really affect users' search strategy. However, the *mixed* strategy was the major search strategy observed for both types of screen size, while the proportion of users who applied a fully breadth-first strategy was relatively small.

Another study has been conducted by Aula et al. [18] using an eye-tracker to observe users' viewing behaviours. Based on their observation, they classified the behaviours of users into two major groups: (1) *economic* and (2) *exhaustive*, related to the depth-first and breadth-first strategies respectively. Economic users examine a few documents before deciding an action, such as clicking the entry link or reformulating a new query, while exhaustive users tend to scrutinize many entries before deciding their next action. White and Drucker [233] conducted a log-based study by recording users' interactions when using a commercial Web search for a five-month period. They were particularly interested in studying *search trails*, that is, a sequence of activities or navigations generated by users after they posed a query to the search engine. Search trails proceed until some predefined signs of termination, such as *closing browser window*, *page timeout*, or *returning to homepage*, which indicate that users have already completed their search task. In their analysis, White and Drucker address two extreme types of users: (1) *navigators*, and (2) *explorers*. Navigators refer to a group of users who have consistent pattern of trails. They tend to follow a direct or simple path from query submission to termination, revisit the main Web domain, and sequentially tackle their search task. Explorers tend to frequently branch in their trails, submit many queries in a session, and visit pages from many Web domains. The explorers also appeared to be inconsistent, in the sense that they applied several search strategies when seeking for information to complete a particular task.

Joachims et al. [110] asked several participants to complete both informational and navigational tasks using the Google search engine in a laboratory setting. While the participants were inspecting the SERPs, all search actions were recorded, including clickthroughs and eye-fixations, with the help of an eye-tracker tool. The key findings are that the participants tended to scan down the ranking in a sequential manner; that users viewed more snippets beyond rank i before decided to click on item at rank i; and that the user clicking behaviour is affected by at least two factors, the trust in search engine being used and the overall quality of the SERPs being inspected. Cutrell and Guan [59] carried out similar experiments using an eye-tracking tool, observing behaviours including the distribution of viewed rank positions prior to a click action and mean arrival time at each rank position. They suggest that increasing the length of snippets increases user search performance on informational tasks, but decreases that on navigational tasks. Thomas et al. [211] recorded viewing behaviours using an eye-tracking tool, and found that users were more likely to examine documents at higher rank positions than those at lower rank positions (that is, top-weightedness).

Mobile-Based Search. The narrow screen of mobile devices hampers users' navigation activities [42]. This condition may cause users to lose their global view of the task being tackled and require them to remember the previously viewed content, and thus increase their cognitive load [7, 158]. Hence, mobile-based search users may have different behaviour from Web-based search users.

Jones et al. [112] studied the behaviour of twenty participants while performing search tasks using two different systems, one with large screen (desktop) and the other with small screen (as a simulation of mobile device). The systems recorded a number of browsing actions, such as the number of scroll up/down actions, and the number of backtrack actions (that is, returning to previously seen items). The key findings are that the number of scrolling actions observed from small screen users is higher than that observed from their large screen counterparts; that down scrolling activities are more common than up scrolling activities for small screen users, suggesting that they performed a few backtracking scroll actions; and that path length for small screen users were shorter than that for their large screen counterparts.

Lagun et al. [129] found that, on mobile phones, the second item in the ranking gets more views than the first one, suggesting that the discount function of existing metrics, such as DCG, has to be slightly corrected. They also observe that mobile-based users tend to focus on the center and top half of the screen, while the desktop-based users pay more attention on the top left side.

Ong et al. [160] compared the behaviour between mobile- and desktop-based users by adopting the methodologies from Wu et al. [236], controlling the information scent level (the number of relevant items on a SERP) and pattern (the distribution of a fixed number of relevant items on a SERP) on the SERPs. Some of the differences were that desktop-based users had a longer search duration and viewed into deeper ranking positions, compared to the mobile-based users; and that information scent level affected desktop-based search behaviours; and that mobile-based users more accurately selected relevant documents on all information scent levels.

2.2.3 User Stopping Behaviour

Moffat et al. [155] argue that the probabilities governing a user model should be computed based on the part of the ranking that has been inspected. Under the assumption that users examine the document in a sequential-inspection manner from top to bottom, a good user model should reflect how and when the user stops examining the SERP. Cooper [55] proposed two stopping rules: (1) a *frustration rule*, whereby the user stops scanning the ranking after they have inspected a fixed number of non-relevant documents; and (2) a *satisfaction rule*, whereby the user stops scanning the ranking after they have inspected a fixed number of relevant documents. Cooper [56] also developed a more complex rule, where the user stops after they believe that the effort of inspecting the next item in the current ranking is greater than that of moving to a new ranking by submitting a new query. This rule is similar to the information foraging theory posited Pirolli and Card [164], and later on adopted by Azzopardi et al. [20].

Maxwell et al. [147] investigated three stopping strategies (a fixed-depth strategy and the first two strategies proposed by Cooper [55]), and carried out experiments to see the performance of each strategy using simulations, as an alternative to the actual user studies. The simulations were based on the work by Baskaya et al. [26], with querying strategies proposed by Keskustalo et al. [121]. They found that adaptive strategies such as the *frustration rule* and the *satisfaction rule* outperformed a fixed-depth strategy such as the one embodied in Prec@K user model, suggesting that the stopping behaviour of the user is adaptive to the relevance of the documents inspected. Maxwell et al. [148] extended their previous work, and still concluded that the frustration rule and the satisfaction rule closely reflect the actual stopping behaviour. However, a fixed-depth strategy with a carefully chosen depth value might also provide a close approximation to the actual stopping behaviour.

Several authors have carried out experiments to see whether or not stopping behaviour is predictable [214, 236]. Toms and Freund [214] analysed logs containing 288 search sessions initiated from 96 participants. They found that in the final stage of a search session users were more engaged in revisiting the pages they have previously viewed. They argued that this might be a means to evaluate whether or not the *satisfaction rule* was met. Finally, they also found that users tended to spend considerable amount of time in viewing additional pages in the last stage. This is an indication that users are assessing the *benefits* and *costs* to either continue or stop searching [214]. Wu et al. [236] studied the correlation between the stopping behaviours and two factors: (1) information scent level, the number of relevant documents in the first SERP; and (2) *need for cognition*, a personality trait describing to what extent a user enjoys activities that require cognitive effort and energy. They found that the *need for cognition* and the probability of stopping have a positive, but weak, correlation in a condition where the density of relevant documents in the first SERP is high. Several authors have proposed models for user actions, including stopping actions, using Bayesian decision model [126] and Markov chains [215, 216, 217].

2.3 Metrics and User Models

Incorporating user behaviour into precision-based metrics is a new direction that is worth being explored in the field of IR evaluation. As noted by Saracevic [188], any IR evaluation should consider both system and user factors, including that an effectiveness metric should be tightly coupled with what kind of user behaviour it embodies. Section 2.3.1 discusses the urgency of the interaction between user models and effectiveness metrics.

Note that incorporating user behaviours into the measurement of search effectiveness is not straightforward, as it requires us to operationalise the concept of user behaviour itself. One way to do that is by employing the notion of *continuation probability* [155]. Nevertheless, it is still difficult to compare one precision-based metric with another in the current setting, since all of them were developed under different assumptions and a different understanding on what a good metric should be. Section 2.3.2 introduces the C/W/L framework, a general structure that unites weighted-precision metrics under the same assumptions of how a user interacts with a SERP.

2.3.1 User Model

The evaluation of search engine effectiveness thus must consider issues from broader contexts, not only the *seeking* context (algorithms) but also the *using* context (users) [188]. Therefore, modern IR systems should pay attention not only on how good the algorithm is at generating the results ranking, but also on an evaluation aspect whereby it can be ensured that the results ranking presented to the user are *usable* to them. Moffat et al. [155] argue that incorporating users' search goal and pattern of behaviour into an effectiveness metric is crucial, in the sense that the scores yielded by a particular effectiveness metric should have a straightforward explanation and interpretation that corresponds to the user's search experience.

However, current test collection-based IR evaluations mostly treat "systems" and "topics" as the only two sources of variation, ignoring other aspects, such as behaviour of users (particularly when users interact with a SERP). While this provides the ability to compare the retrieval algorithm of one system with that of the others (due to its sensitivity to the system changes), this makes it difficult to understand what changes happen to the system if it is given to the actual users [155]. Bailey et al. [24] introduce a new test collection that embodies two sources of user variability: (1) query formulation and (2) the expected number of relevant documents for each query. One particular interest is how to incorporate user behaviour into search effectiveness metrics. Note that an effectiveness metric that is sensitive to the behaviour of users is also important in the case where a particular IR system needs to be evaluated upon two different populations of users that in general have different behaviours.

The notion of *user model*, introduced by Moffat and Zobel [151], is an effort to bridge the gap between effectiveness metrics and user variability by making metrics sensitive to the variability of user behaviours. Moffat et al. [155] further explain a user model as a formal description of how a population of users interacts with a SERP, from which three characteristics are identified. One of those characteristics is the conditional probability that a user moves to rank position $i + 1$ in the SERP, given that they have examined the one at rank i under the assumption that the user scans down the ranking one-by-one from top to bottom. This has a special name, called *continuation probability*. Section 2.3.2 will elaborate on continuation probability and the other two hypothetical behaviours.

2.3.2 C/W/L Framework

Different metrics tend to have different interpretations and score units, which makes them difficult to compare, particularly in terms of the user model they embody. Moffat et al. [153] develop a framework that generalises as well as unites the family of weighted-precision metrics. By framing different metrics within the same framework and parameters, it is possible to get comparative value units (such as utility or effort) and compare the underlying user model and characteristics between them. With this framework, weighted-precision metrics can be generalised as follows:

$$M(\overrightarrow{r}) = \sum_{i=1}^{\infty} W_M(i) \cdot g(r_i)\,, \tag{2.26}$$

where $W_M(i)$ is a weighting scheme with $\sum_{i=1}^{\infty} W_M(i) = 1$. Therefore, $M(\overrightarrow{r})$ can generally be interpreted as the expected gain (utility) derived by users per document inspected. Moffat et al. [153] suggest that there are two interpretations of $W_M(i)$ depending on the assumption of how users examine the ranked list of documents. These are: (1) users randomly select a document to be examined according to a probability distribution, or (2) users sequentially scan down the ranking from top to bottom.

Random Examination. $W_M(i)$ can be regarded as the probability that users examine the document at rank i by *random selection* at any time, which means that a sequence of item inspections is a sequence of samples drawn from the probability distribution $W_M(i)$ [153]. Consider RBP with persistence ϕ that has the following specification:

$$W_{\mathsf{RBP}}(i) = (1 - \phi) \cdot \phi^{i-1}\,.$$

For example, RBP with $\phi = 0.8$ has the following weighting vector:

$$\mathbf{W}_{\mathsf{RBP}}(\phi = 0.8) = \langle 0.20, 0.16, 0.13, 0.10, 0.08, 0.07, \ldots \rangle \,;$$

Assuming that users perform random selections, they have a 20% chance of examining the document at rank 1 at any time, 16% chance of examining the document at rank 2, and so on.

Sequential Examination. The most well-developed and widely-used browsing model is the cascade browsing model, where users scan down the ranked list one-by-one, and then stop at some point in the ranking [110]. Based on this examination model, Moffat et al. [153] suggest the second interpretation of $W_M(i)$ via the notion of conditional continuation probability, denoted by $C_M(i)$:

$$C_M(i) = \frac{W_M(i+1)}{W_M(i)}\,, \tag{2.27}$$

which represents the probability that the user continues to examine the document at rank $i+1$ given that they have sequentially examined all documents from rank 1 to rank i. For RBP with persistence ϕ, the continuation probability is simply:

$$C_{\mathsf{RBP}}(i) = \phi\,.$$

Therefore, the notion of continuation probability is a generalisation of the notion of persistence in the context of RBP user model of Moffat and Zobel [151], which allows persistence to be variable and adaptive depending on the part of the ranking that the user has inspected so far before they decide to examine the next document.

Expected Search Length. The *expected search length* (ESL), or the expected number of documents inspected, of any user model is given by the following formulation:

$$\mathsf{ESL} = \sum_{i=1}^{\infty} i \cdot L_M(i) = \frac{1}{W_M(1)} = 1 + \sum_{i=1}^{\infty} \left(\prod_{j=1}^{i} C_M(j) \right) \,. \tag{2.28}$$

The latter equation indicates that the computation of $1/W_M(1)$ should converge. Otherwise, the modelled user never stops inspecting a hypothetical infinite ranking, and thus $W(i) \approx 0$. In this case, truncation at a finite depth is often necessary.

Expected Total Gain (*ETG*). The metrics defined by Equation 2.26 are *expected rate of gain* (*ERG*) metric, denoted by $M_{ERG}(\overrightarrow{r})$, since it measures the average gain (or utility) per document inspected. The second type of metric that can be computed is an *expected total gain* (*ETG*) metric, denoted by $M_{ETG}(\overrightarrow{r})$, that measures the average total volume of relevance derived by the end of a ranking inspection. It is computed as follows:

$$M_{ETG}(\overrightarrow{r}) = \sum_{i=1}^{\infty} \left(L_M(i) \cdot \sum_{j=1}^{i} g(r_j) \right) = \sum_{i=1}^{\infty} \frac{W_M(i)}{W_M(1)} \cdot g(r_i) = \frac{M_{ERG}(\overrightarrow{r})}{W_M(1)} . \tag{2.29}$$

By connecting Equations 2.28 and 2.29, it can be shown that $M_{ETG}(\overrightarrow{r}) = \mathsf{ESL} \cdot M_{ERG}(\overrightarrow{r})$. Therefore, *ERG* metrics yield scores in the units of “relevance (gain) per document”, while *ETG* metrics gives score in the units of “relevance”.

Characteristics of a User Model. To complete a two-way relationship between $C_M(i)$ and $W_M(i)$, Moffat et al. [153] show that $W_M(i)$ can be computed from $C_M(i)$ as follows:

$$W_M(i) = \frac{1}{\sum_{k=1}^{\infty} \prod_{l=1}^{k-1} C_M(l)} \prod_{j=1}^{i-1} C_M(j) . \tag{2.30}$$

In addition, a further characteristic can be derived from $C_M(i)$ and $W_M(i)$: the probability that the document at rank i is the last one examined by the user, denoted by $L_M(i)$:

$$\begin{aligned} L_M(i) &= C_M(1) \cdot C_M(2) \ldots C_M(i-1) \cdot (1 - C_M(i)) \\ &= \frac{W_M(2)}{W_M(1)} \cdot \frac{W_M(3)}{W_M(2)} \cdots \frac{W_M(i)}{W_M(i-1)} \cdot (1 - \frac{W_M(i+1)}{W_M(i)}) \\ &= \frac{1}{W_M(1)} \cdot (W_M(i) - W_M(i+1)) . \end{aligned} \tag{2.31}$$

The three-way relationship between $C_M(i)$, $W_M(i)$, and $L_M(i)$ is completed by the following equation:

$$C_M(i) = \frac{\sum_{j=i+1}^{\infty} L_M(j)}{\sum_{j=i}^{\infty} L_M(j)} . \tag{2.32}$$

This three-way relationship forms a measurement framework, called C/W/L, which unites most of the weighted-precision metrics by framing their underlying user models into three characteristics that reflect how the user behaves when interacting with the ranked list of results. Summarising the descriptions above, the components of C/W/L are:

1. The conditional continuation probability, $C_M(i)$, that reflects the conditional persistence of users at rank i, or the tendency of users to continue to the next item in the ranking.

2. The weight, $W_M(i)$, that indirectly reflects the likelihood of viewing a particular document at any time, assuming that a sequence of random selections happens.

3. The *last* probability, $L_M(i)$, reflecting that the item listed at rank i is the last one inspected by the user.

Note that the aforementioned characteristics are related to each other, meaning that once one of the three characteristics is specified, the other two characteristics can be computed from it. For example, when $C_M(i)$ is specified, both $W_M(i)$ and $L_M(i)$ can be easily determined using the Equations 2.30 and 2.31.

Describing User Models Using the C/W/L Framework. Moffat et al. [153] describe the notion of user model using the C/W/L framework and further argue that an ideal user model possesses the following five properties:

1. The probabilities in regard to $W_M(i)$, $C_M(i)$, and $L_M(i)$ should be computed based on the properties of the part of the ranking that have been examined by the simulated users, without considering properties from the whole ranking.

2. $W_M(i)$ should be non-increasing and non-zero, $W_M(i) \geq W_M(i+1)$ and $W_M(i) > 0$; As a consequence, $C_M(i)$ should be greater than zero.

3. All other factors being equal, $C_M(i)$ should be non-decreasing, that is, $C_M(i) \leq C_M(i+1)$. Moffat et al. [153] provide empirical support from a lab-based user study, suggesting that $C_M(i)$ increases with rank position i.

4. All other factors being equal, $C_M(i)$ decreases, as relevance accumulates.

5. $C_M(i)$ should incorporate T, the anticipated number of relevant documents for undertaking the search (or the search goal), as one of its factors; and all other factors being equal, $C_M(i)$ tends to increase as T increases.

A later study by Azzopardi et al. [20] suggests addition of a sixth property, that $C_M(i)$ should be affected by the anticipated minimum rate of gain; and $C_M(i)$ tends to decrease as the current rate of gain drops below that minimum expectation. De Vries et al. [61] and Moffat and Wicaksono [150] further suggest that $C_M(i)$ decreases if extremely poor documents are encountered.

Goal Sensitivity. A user-oriented metric should be *goal sensitive*, since search is a goal-directed activity [53, 235]. For example, if a user performs a difficult search task that involves more than five useful documents, shallow metrics, such as Prec@K with $K = 1$, might be less useful than deeper metrics. In particular, Cooper [53] and Moffat et al. [155] contend that a metric should take into account the number of relevant documents the user wishes to examine in order to satisfy their information need. This quantity, denoted by T, is one approach to operationalise the concept of user goal. Moffat et al. [155] further demonstrate that T varies with task complexity, and that T is a key factor for predicting observed $C(i)$.

Mapping Into C/W/L. For several metrics, such as average precision (AP), reciprocal rank (RR), and discounted cumulative gain (DCG), the mapping process is not trivial. We now show a process of how an existing metric can be mapped into the C/W/L framework. The majority of existing metrics can be described as follows:

$$\sum_{i=1}^{N} D(i) \cdot g(r_i)\,,$$

where $D(i)$ is a *non-increasing* discount function over rank i. The latter equation is a general picture of metrics, instead of being a conceptual framework, since $D(i)$ clearly has multiple interpretations, and thus different $D(i)$ may yield different scores with different units as well. Nevertheless, $D(i)$ is typically related to either viewing or stopping behaviour [39].

To standardise existing metrics through the lens of the C/W/L framework, the infinite sum of $D(i)$ over i should be computed. That is,

$$s = \lim_{m\to\infty} \sum_{i=1}^{m} D(i)\,.$$

The resultant value s is used to determine $W_M(i)$ (note that $L_M(i)$ and $C_M(i)$ can be computed once $W_M(i)$ has been specified):

- if $s = 1$, then $W_M(i) = D(i)$.
- if s is a positive real number other than 1, then $W_M(i) = D(i)/s$.

- if $s \to \infty$, then truncation at a finite depth K is necessary:

$$W_M(i) = \begin{cases} D(i) / \sum_{j=1}^{K} D(j) & i \le K \\ 0 & i > K\,. \end{cases}$$

Note that these steps cannot be applied to every measure; nevertheless, they provide a process for mapping metrics into the C/W/L framework. Section 2.4 will demonstrate how these steps can be applied to several offline metrics, such as average precision (AP), reciprocal rank (RR), and discounted cumulative gain (DCG).

2.4 Classification of User Models

As suggested by Moffat et al. [154], user models can be generally classified into two broad categories. The first is of *static* or *positional user models*, for which the continuation probability, $C_M(i)$, is defined solely as a function of ranking position i. The second one is *adaptive* or *cascade user models*, for which the continuation probability $C_M(i)$ is affected by not only the current inspected rank i, but also by the document relevance, particularly by the part of the ranking $\overrightarrow{r}$ that has been inspected by the user.

2.4.1 Static User Models

In this category, $C_M(i)$ only depends on the ranking position currently being inspected by the user, without considering factors that depend on the quality of the part of the ranking. This section describes Prec@K, RBP, SDCG, and INSQ as examples from this category.

Precision at K. The user modelled by Prec@K always inspects the first K documents in the ranking before stopping at rank K, in the sense that all documents from rank 1 to K have the same chance of being examined by the user; and that they will not examine the documents beyond rank K:

$$C_{\mathsf{Prec@K}}(i) \;=\; \begin{cases} 1 & 1 \le i < K \\ 0 & \text{otherwise}\,, \end{cases} \tag{2.33}$$

$$W_{\mathsf{Prec@K}}(i) \;=\; \begin{cases} \frac{1}{K} & i \le K \\ 0 & \text{otherwise}\,. \end{cases} \tag{2.34}$$

Rank-Biased Precision. As is described previously, the user modelled by RBP always inspects the documents one by one from the top, with probability of ϕ that they will progress from rank i to rank $i+1$:

$$C_{\mathsf{RBP}}(i) = \phi\,, \tag{2.35}$$

$$W_{\mathsf{RBP}}(i) = (1-\phi)\cdot\phi^{i-1}\,. \tag{2.36}$$

Scaled Discounted Cumulative Gain. As is described in Equation 2.15 (page 29), Burges et al. [35] describe an alternative formulation of discounted cumulative gain with

$$D_{\mathsf{DCG}}(i) = \frac{1}{\log_2(i+1)}\,.$$

Here $D_{\mathsf{DCG}}(i)$ is already a non-increasing function. However, the infinite sum of $D_{\mathsf{DCG}}(i)$ over rank position i is divergent. Therefore, truncation at depth K is necessary to avoid $W(i) \approx 0$. When that is done, what results is as *scaled* discounted cumulative gain at depth K (SDCG@K) with scores bounded between 0 and 1 [153]:

$$W_{\mathsf{SDCG}}(i) = \begin{cases} \frac{1}{\log_2(i+1)\cdot\sum_{j=1}^{K} 1/\log_2(j+1)} & i \le K \\ 0 & i > K\,, \end{cases} \tag{2.37}$$

$$C_{\mathsf{SDCG}}(i) = \begin{cases} \frac{\log_2(i+1)}{\log_2(i+2)} & i < K \\ 0 & i \ge K\,. \end{cases} \tag{2.38}$$

Nevertheless, knowing that $1/W_{\mathsf{SDCG}}(1) = \sum_{j=1}^{K} 1/\log_2(j+1)$, an *expected total gain* version of this specification evaluates to:

$$\sum_{i=1}^{\infty} \frac{W_{\mathsf{SDCG}}(i)}{W_{\mathsf{SDCG}}(1)} \cdot g(r_i) = \sum_{i=1}^{K} \frac{1}{\log_2(i+1)} \cdot g(r_i) = \mathsf{DCG@K}(\overrightarrow{r}; K, b)\,.$$

Therefore, DCG@K can be interpreted as an *expected total gain*, instead of *expected rate of gain*, in the perspective of the C/W/L framework.

Inverse Square. Moffat et al. [152] propose inverse square (INSQ) shas a user model that is sensitive to the user's goal T. Recall that T is the expected number of relevant documents the user wishes to examine in order to satisfy their information need (see five properties described on page 49):

$$C_{\mathsf{INSQ}}(i) = \left(\frac{i+2T-1}{i+2T}\right)^2 , \tag{2.39}$$

$$W_{\mathsf{INSQ}}(i) = \frac{1}{S(2T-1)} \cdot \frac{1}{(i+2T-1)^2} , \tag{2.40}$$

where $S(x) = (\pi^2/6) - (\sum_{j=1}^{x} 1/j^2)$. The INSQ user model differs from RBP in two respects. First, as opposed to RBP, which has a constant $C(i)$, the $C(i)$ of a user modelled by INSQ increases as the user progresses in the ranking. Second, INSQ explicitly specifies the user's goal for undertaking the search via the parameter T, while RBP implicitly encodes this via ϕ. Moffat et al. [155] further note that RBP can be "somewhat" goal sensitive by setting $\phi = 1 - 1/(2 \cdot T)$.

2.4.2 Adaptive User Models

In this category, $C_M(i)$ varies as the modelled users encounter relevance in the part of the ranking seen so far. Note that it can be difficult to define $W_M(i)$ using a closed form since it depends on the relevance vector $\overrightarrow{r}$. This section describes AP, ERR, RR and INST as examples from this category.

Average Precision. Average precision (AP) is defined as:

$$\frac{1}{R}\sum_{i=1}^{\infty} \mathsf{Prec@K}(\overrightarrow{r}; i) \cdot r_i = \sum_{i=1}^{\infty}\left(\frac{1}{R}\sum_{j=1}^{k}\frac{r_j}{k}\right) \cdot r_i ,$$

where R is the number of relevant items in the collection. Recall that r_i is binary in the context of AP. Here it can be seen that $D_{\mathsf{AP}}(i) = (1/R) \cdot \mathsf{Prec@K}(\overrightarrow{r}; i)$ is not a non-increasing function, but a non-increasing version of AP's discount function using algebraic manipulation as follows:

$$\begin{aligned}
\sum_{i=1}^{\infty}\left(\frac{1}{R}\sum_{j=1}^{k}\frac{r_j}{k}\right) \cdot r_i &= \frac{1}{R}\sum_{i=1}^{\infty}\left(\sum_{j=1}^{k}\frac{r_j}{k}\right) \cdot r_i \\
&= \frac{1}{R}\begin{bmatrix} \frac{r_1}{1}r_1 + \\ \frac{r_2}{2}r_1 + \frac{r_2}{2}r_2 + \\ \frac{r_3}{3}r_1 + \frac{r_3}{3}r_2 + \frac{r_3}{3}r_3 + \\ \cdots \end{bmatrix}
\end{aligned}$$

$$= \frac{1}{R}\left(r_1 \sum_{i=1}^{\infty} \frac{r_i}{i} + r_2 \sum_{i=2}^{\infty} \frac{r_i}{i} + r_3 \sum_{i=3}^{\infty} \frac{r_i}{i} + \cdots\right)$$
$$= \sum_{i=1}^{\infty} \left(\frac{1}{R} \sum_{j=i}^{\infty} \frac{r_j}{j}\right) \cdot r_i \, .$$

Hence, the alternative discount function for AP is:

$$D'_{\mathsf{AP}}(i) = \frac{1}{R} \sum_{j=i}^{\infty} \frac{r_j}{j} \, .$$

It can be seen that $D'_{\mathsf{AP}}(i)$ is a non-increasing function, and that $\lim_{m\to\infty} \sum_{i=1}^{m} D'_{\mathsf{AP}}(i) = 1$. Therefore, mapping AP into the C/W/L framework leads to the following specification:

$$W_{\mathsf{AP}}(i) = \frac{1}{R} \sum_{j=i}^{\infty} \frac{r_j}{j} \text{ , and thus } C_{\mathsf{AP}}(i) = \frac{\sum_{j=i+1}^{\infty} r_j/j}{\sum_{j=i}^{\infty} r_j/j} \, . \tag{2.41}$$

This specification implies that the user modelled by AP is clairvoyant, with the decision to continue from rank position i to $(i+1)$ depending on what they will see in the future, suggesting that $C_{\mathsf{AP}}(i)$ is plausible only if it is assumed that the user has the ability to understand the part of the ranking that has *not yet been seen*.

(Expected) Reciprocal Rank. Both reciprocal rank (RR) and expected reciprocal rank (ERR) [44] can be described as follows:

$$\sum_{i=1}^{\infty} \frac{1}{i} \cdot \prod_{j=1}^{i-1} (1 - g(r_j)) \cdot g(r_i) \, .$$

For RR, $g(r_i)$ is binary: either 0 or 1. For ERR, $g(r_i) \in [0, 1]$ so as to handle graded relevance (see Equation 2.22 on page 31).

In the case of RR, $g(r_i)$ can be directly replaced by r_i since r_i is binary. By assuming that the first fully relevant document can be found at rank position H, that is $r_H = 1$ and $r_i = 0$ for $i < H$, it can be shown that:

$$\sum_{i=1}^{\infty} \frac{1}{i} \cdot \prod_{j=1}^{i-1} (1 - r_j) \cdot r_i = \sum_{i=1}^{\infty} \frac{1}{H} \cdot \prod_{j=1}^{i-1} (1 - r_j) \cdot r_i \, .$$

Thus, the discount function can be decribed as follows:

$$D_{\mathsf{RR}}(i) = \frac{1}{H} \cdot \prod_{j=1}^{i-1}(1 - r_j) = \begin{cases} 1/H & i \leq H \\ 0 & i > H\,, \end{cases}$$

It is clear that $D_{\mathsf{RR}}(i)$ is a non-increasing function with $\lim_{m\to\infty} \sum_{i=1}^{m} D_{\mathsf{RR}}(i) = 1$. Therefore, $W_{\mathsf{RR}}(i) = D_{\mathsf{RR}}(i)$, and thus:

$$C_{\mathsf{RR}}(i) = \begin{cases} 1 & i < H \\ 0 & i = H\,. \end{cases} \tag{2.42}$$

With ERR, the situation is more complex. In practice, a ranking is usually finite until depth N, and thus the remaining items from rank position $N+1, N+2, \ldots$ are usually assumed to be non-relevant, that is $r_i = 0$ for $i > N$. In this situation, it can be seen that:

$$\begin{aligned} \sum_{i=1}^{\infty} D_{\mathsf{ERR}}(i) &= \lim_{m\to\infty} \sum_{i=1}^{m} \frac{1}{i} \cdot \prod_{j=1}^{i-1}(1 - g(r_j)) \\ &= \sum_{i=1}^{N} \frac{1}{i} \cdot \prod_{j=1}^{i-1}(1 - g(r_j)) + \lim_{m\to\infty} \sum_{i=N+1}^{m} \frac{1}{i}\,. \end{aligned}$$

It is clear that $D_{\mathsf{ERR}}(i)$ is already non-increasing. However, the rightmost term in the latter equation is an infinite sum of *harmonic series* that is divergent. Therefore, truncation at depth K is needed. One possible interpretation is as an instance of ETG metric with:

$$C_{\mathsf{ERR}}(i) = \begin{cases} \frac{i}{i+1} \cdot (1 - g(r_i)) & i < K \\ 0 & i \geq K\,. \end{cases} \tag{2.43}$$

INST. Moffat et al. [155] propose a user model, INST, that has all five properties of a user model described in Section 2.3 (page 49). In general, INST extends INSQ [152] so that it now becomes *adaptive* to the volumes of relevance that accrue.

$$C_{\mathsf{INST}}(i) = \left(\frac{i + T + T_i - 1}{i + T + T_i}\right)^2, \tag{2.44}$$

where $T_i = T - \sum_{j=1}^{i} g(r_i)$ represents the expected remaining number of relevant documents that the user needs to identify. With this definition, it is clear that $C_{\mathsf{INST}}(i)$ increases with i (“sunk cost” property), has a positive correlation with T, and decreases as T_i decreases (or as the relevant documents are accumulated).

2.4.3 Incorporating Costs into Metrics

Many effectiveness metrics, such as Prec@K, average precision (AP), discounted cumulative gain DCG (and its normalised version [102]), reciprocal rank (RR), rank-biased precision (RBP) [151], expected reciprocal rank (ERR) [44], and INST [155], assume that the modelled user views each document in the ranking at a constant rate; and that once they reach at a particular document, they scan all words in the document, and thus either derive the utility associated with the document, or derive nothing if it is non-relevant. However, some cost-based factors, such as the length of documents, the number of documents that have been inspected, and the time spent during the SERP examination process, have been shown to affect user behaviour [19, 20, 195]. Other cost-based factors, such as the number of issued queries, also have a relationship with user satisfaction [4, 108]. This section summarises research that addresses such issues.

Time-Biased Gain. Smucker and Clarke [195] argue that the discount function $D(i)$ in any weighted-precision metric, $\sum_{i=1}^{\infty} D(i) \cdot g(r_i)$, is a function of time, including the amount of time spent by the user to reach the document at rank i and to start to read it. Smucker and Clarke still employ the cascade browsing model, where the user sequentially visits each document in the ranking, until they stop and reformulate, or commence some other activity. A further development is that a model is incorporated to allow the metric to be sensitive to cases where the user may take more time to examine the longer documents than shorter ones, or the user may abandon the document if its summary seems to be not relevant. Smucker and Clarke describe the resultant time-biased gain (TBG) user model, whose metric can be defined as follows:

$$M_{\mathsf{TBG}}(\overrightarrow{r}) = \frac{1}{Z} \sum_{i=1}^{\infty} D_{\mathsf{TBG}}(T(i)) \cdot g(r_i)\,, \tag{2.45}$$

where $D_{\mathsf{TBG}}(t)$ is the discount function working as a response to time; $T(i)$ is the expected time required by the user to arrive at rank i and start reading the corresponding document, with $T(1) = 0$ seconds; $g(r_i)$ is the gain mapping function; and Z is a normalisation factor.

Current search engine interfaces typically provide summaries or snippets, in addition to the links to the full documents, which allow users to pre-judge the quality of the corresponding document before deciding to click and see the full content of it. The user often skips viewing the full document if its summary is not promising. Thus, Smucker and Clarke [195] define $T(i)$ as an accumulation of the total time spent to read the summaries, $T_S(j)$, and the documents, $T_D(j)$, where $1 \leq j < i$. By assuming that summaries tend to be equal in length, the user most likely spends the same time to read each of the summaries in the ranking, meaning that $T_S(i) = T_S$, where T_S is simply the time spent to read any summary. However, a similar assumption does not hold for documents, as they vary in terms of the total number of words. This leads to the definition of $T_D(i) = a \cdot l_i + b$, where l_i is the length of document at rank i measured in words, with a and b being trainable parameters. Moreover, based on the observation that not all users click on relevant documents, Smucker and Clarke [195] add a factor $P(C = 1 \mid \mathit{Rel})$ factor that represents the probability that the user clicks on a particular document given its relevance value *Rel*. This completes the definition of $T(i)$ as follows:

$$T(i) = \sum_{j=1}^{i-1} T_S + T_D(j) \cdot P(C = 1 \mid \mathit{Rel} = r_j)\,. \tag{2.46}$$

Next, the discount function $D_{\mathsf{TBG}}(t)$ is modelled using a decay exponential function:

$$D_{\mathsf{TBG}}(t) = \exp\left(-t \cdot \frac{\ln 2}{h}\right)\,, \tag{2.47}$$

where h is a trainable parameter denoting the time at which 50% of users in the population stop examining the SERP. Finally, all trainable parameters that are involved in developing the aforementioned metric were estimated using the data collected from a user study. For example, based on their data, $h = 224$ seconds, $T_S = 4.4$ seconds, and $T_D(i) = 0.018 \cdot l_i + 7.8$ seconds .

As noted by Smucker and Clarke [195], the ideal condition is when the collection is full of zero-length relevant documents. Suppose t_x is the expected time to read the summary and the content of a zero-length relevant document. Thus,

$$\begin{aligned}
\sum_{i=1}^{\infty} D_{\mathsf{TBG}}(i) &= \sum_{i=1}^{\infty} \exp\left(-T(i) \cdot \frac{\ln 2}{h}\right) \\
&\leq \sum_{i=1}^{\infty} \exp\left(-t_x \cdot (i-1) \cdot \frac{\ln 2}{h}\right) \\
&= \sum_{i=0}^{\infty} \exp\left(-t_x \cdot i \cdot \frac{\ln 2}{h}\right) \\
&= \frac{1}{1 - \exp\left(-t_x \cdot \frac{\ln 2}{h}\right)}\,.
\end{aligned}$$

According to the calibration values stated by Smucker and Clarke [195], the latter term evaluates to 34.9. Therefore, $D_{\mathsf{TBG}}(i)$ is a non-increasing function, and the infinite sum of $D_{\mathsf{TBG}}(i)$ over rank position i converges to a positive real number ≤ 34.9. This leads to the C/W/L specification of TBG via the following $C_{\mathsf{TBG}}(i)$:

$$C_{\mathsf{TBG}}(i) = \frac{\exp\left(T(i) \cdot \frac{\ln 2}{h}\right)}{\exp\left(T(i+1) \cdot \frac{\ln 2}{h}\right)} \,. \tag{2.48}$$

With this specification, the expected rate of gain version of the metric represents the normalised version of TBG, while the expected total gain version represents the TBG score without normalisation. Note that, when the expected time to read an item (the summary and content) is constant across all ranking position i, TBG reduces to RBP with

$$\phi = \exp\left(-\frac{\ln 2}{h}\right) = 2^{-\frac{1}{h}} \,.$$

A major problem raised from TBG is that its user stopping model relies solely on the total number of words read by the user. As a result, it does not handle cases with different task complexity. Time-biased gain assumes that all readers exhibit the same behaviour, and do so in response to all queries.

U-Measure. Instead of defining *cost* in terms of the amount of time spent, Sakai and Dou [181] propose a metric, U-measure (UM), whose weight function decays with the amount of text read by the user. This metric can also be computed using Equation 2.45 in page 56, but with the following discount function:

$$D_{\mathsf{UM}}(i) = \max\left(0,\, 1 - \frac{pos(i)}{L}\right) \,, \tag{2.49}$$

where $pos(i)$ is the amount of text (measured in characters) read by the user to reach the document at rank i, with $pos(1) = 0$; and L is the largest number of characters that the user may have to read in a session. Based on observation on `Bing.com` data collected in 2012, Sakai and Dou [181] suggest to use $L = 132{,}000$ characters. As demonstrated by Azzopardi et al. [21], UM can be mapped into the C/W/L framework via

$$C_{\mathsf{UM}}(i) = D_{\mathsf{UM}}(i+1)/D_{\mathsf{UM}}(i) \,. \tag{2.50}$$

Bejeweled Player Model. Zhang et al. [241] develop a user model, bejeweled player model (BPM), arguing that the simulated user will not stop inspecting the SERP unless one of the following conditions has been met: (1) the total gain accumulated so far exceeds

T; (2) or the total number of documents inspected exceeds K (that is, the anticipated cost, or the number of documents the user wishes to see). Note that the first condition is similar to that for INST, while the second one is similar to that for Prec@K. Azzopardi et al. [21] show that BPM can be mapped into the C/W/L structure using the following specification:

$$C_{\mathsf{BPM}}(i) = \begin{cases} 1 & \sum_{j=1}^{i} g(r_j) < T \text{ and } i < K \\ 0 & \text{otherwise}\,. \end{cases} \tag{2.51}$$

Zhang et al. [241] further propose two versions of BPM: static BPM (S-BPM) and dynamic BPM (D-BPM). For the static version, T and K are fixed. For the dynamic version, both T and K change as the user encounters relevance. When the user inspects a relevant document, the user's desire (T) and their willingness to inspect more documents (K) increase; and when they examines a non-relevant document, both T and K decrease.

Information Foraging-Based Metric. Azzopardi et al. [20] propose a new user model based on information foraging theory (IFT) by exploiting the C/W/L structure [164]. The idea behind the IFT-based user model is that, similar to the natural foraging behaviour of typical people, the user will keep observing the current information patch when the rate of gain experienced from the current patch can still be tolerated, but when they feel that the rate of gain is decreasing, and think that other patches would most likely contain more useful information, the user will likely leave the current patch and move to a new information patch. In this case, an information patch can be thought as a SERP where the user examines the ranked list of items to satisfy their information needs. Azzopardi et al. [20] argue that the user's information foraging behaviour depends on two factors: (1) the anticipated number of relevant documents (goal-sensitive), and (2) the expected minimum rate of gain (rate-sensitive). Both factors are directly modelled via conditional continuation probability functions $C_{\mathsf{IFT\text{-}C1}}(i)$ and $C_{\mathsf{IFT\text{-}C2}}(i)$, respectively:

$$C_{\mathsf{IFT\text{-}C1}}(i) = 1 - \left(1 + b_1 \cdot e^{(T-\gamma_i)R_1}\right)^{-1}, \tag{2.52}$$

$$C_{\mathsf{IFT\text{-}C2}}(i) = \left(1 + b_2 \cdot e^{(A-\frac{\gamma_i}{\kappa_i})R_2}\right)^{-1}, \tag{2.53}$$

where T is the user's desire, γ_i is the total number of relevant documents accumulated so far, A is the anticipated rate of gain, κ_i is the time spent so far to reach the document at rank i, and the remaining variables b_1, b_2, R_1, R_2 are empirical parameters. With these definitions, both IFT-C1 and IFT-C2 are adaptive, since their $C(i)$ functions are affected by the document relevance. Finally, the user behaviour encoding the two factors is modelled as follows:

$$C_{\mathsf{IFT}}(i) = C_{\mathsf{IFT\text{-}C1}}(i) \cdot C_{\mathsf{IFT\text{-}C2}}(i)\,. \tag{2.54}$$

Therefore, IFT is goal-sensitive, rate-sensitive, and adaptive. Further, Azzopardi et al. [20] use $b_1 = b_2 = 0.25$, $R_1 = R_2 = 10$, $T = 0.2$, and $A = 0.1$ to model typical Web search users.

To measure the accuracy of the proposed user model, Azzopardi et al. [20] use three criteria: (1) the likelihood of stopping rank, (2) the estimated utility experienced by the user, and (3) the estimated total time spent on the SERP. For data, they collected click-through logs from a major web search engine containing 1,000 common head queries with a set of relevance judgements. The first criteria assumes that the last click rank is the stopping rank, which is not always true in reality. Moreover, in their evaluation process, they did not group the queries based on the task complexity, such as those proposed by Broder [31]. It is widely known that viewing patterns of typical Web search users are heavily top-weighted, meaning that in the majority of users only look at the top links, click one of them, and then stop. While the IFT approach is another important development, their use of (only) head queries means it remains unclear whether more patient users and more extensive information needs can be accurately explained.

Summary. To conclude this section, Table 2.1 (page 61) shows the categorisation of metrics against several properties that have been proposed for defining $C(i)$. Five properties are taken from Moffat et al. [155]; one property is related to the notion of "rate of gain" used for defining IFT [20]; and others involve effort-based factors, such as the number of characters read [181] and the amount of time spent [195].

2.5 User Satisfaction

User satisfaction, which is tightly coupled with the contentment experienced by human beings when a specified need has been fulfilled [117], is an important goal of any search activity [55]. This section explores the concept of user satisfaction in the context of IR evaluation and discusses factors that affect this concept.

Property for $C(i)$	AP	SDCG@K Prec@K	RBP	ERR@K RR@K	INSQ	TBG UM	INST	BPM	IFT
Properties from Moffat et al. [155]									
$C(i)$ is based on the part of the ranking that has been inspected.	✗	✓	✓	✓	✓	✓	✓	✓	✓
$C(i)$ does not depend on R.	✗	✓	✓	✓	✓	✓	✓	✓	✓
$C(i)$ is explicitly affected by T.	✗	✗	✗	✗	✓	✗	✓	✓	✓
$C(i)$ is non-decreasing with i.	✗	✗	✓	✗	✓	✗	✓	✗	✗
$C(i)$ is positively correlated with $T - \sum_{j=1}^{i} r_j$.	✗	✗	✗	✗	✗	✗	✓	✗	✓
"Rate of gain" [20]									
$C(i)$ decreases as the rate of gain, $\sum_{j=1}^{i} r_j/i$, decreases.	✗	✗	✗	✓	✗	✗	✗	✗	✓
Other properties									
$C(i) > 0$ for $i \geq 1$.	✗	✗	✓	✗	✓	✓	✓	✗	✓
$C(i)$ is affected by the amount of time spent so far, or by the number of characters scanned so far [181, 195].	✗	✗	✗	✗	✗	✓	✗	✗	✗

Table 2.1: Categorisation of metrics against existing properties for $C(i)$. Reciprocal rank (RR) and ERR are assumed to be evaluated over a ranking of depth K.

2.5.1 The Concept of User Satisfaction for IR Evaluation

Spärck Jones [199] argue that the concept of user satisfaction is central to the evaluation of an IR system, and thus this concept cannot be simply dismissed in any evaluation experiment. Su [204] further argues that considering the concept of user satisfaction has a key advantage: it takes into account users' subjective assessment in evaluating many aspects of IR systems. Both arguments (by Spärck Jones [199] and Su [204]) are in agreement with the spirit of integrating evaluation processes at system and user levels, posited by Saracevic [188].

While some authors contend that the concept of user satisfaction should be considered in any IR evaluation, others have also argued that a measurement that depends on this concept alone may not be reliable [84, 138, 197]. Kelly [117] notes that the concept of satisfaction is internal to the user, and thus it is not directly observable. Al-Maskari and Sanderson [4] further argue that user satisfaction has ambiguous definitions; and that it is difficult to develop instruments for assessing satisfaction. Liu et al. [140] show that users vary greatly in terms of providing satisfaction ratings for the same SERP. Based on the approach of Viswanathan [223], Kelly [117] suggests that satisfaction should be assessed using multiple items, such as the quality of SERPs, system response time, or system preference.

Soergel [197] argues that the ultimate goal of any IR system is not to make the user happy (subjective satisfaction), but to make the user successful (improved performance). Soergel [197] further observes that users could still be satisfied with non-relevant documents in the SERPs. This is called the *user-distraction* phenomenon. Recently, Liu et al. [138] found a clear evidence of this phenomenon. Similarly, Hildreth [84] also argues that user satisfaction can be easily influenced by non-performance factors, and hence there is no clear relationship between user satisfaction and actual search effectiveness. Hildreth further demonstrates that user satisfaction has weak correlation with *user performance* (the number of relevant items found by the user), but strong relationship with *ease of use* and *system usefulness*. This problem has also been investigated in recent study by Liu et al. [138] who conclude that user satisfaction and other criteria, such as search success (relevance, credibility, and readability on each landing page) [159], should be used together in any IR evaluation.

Notwithstanding these concerns, the concept of user satisfaction has been widely used in the field of IR evaluation [75]; and recent work has shown that system effectiveness has a significant relationship with user satisfaction either via the general notion of relevance

[5, 91, 108] or the notion of usefulness [145]. Al-Maskari and Sanderson [4] investigated four factors that potentially influence user satisfaction: (1) system effectiveness, which measures the objective of the IR system via the traditional precision or average precision; (2) user effectiveness, which can be assessed by the number of relevant documents examined by users or the amount of time to complete the task; (3) user effort, such as the number of queries or the ranking positions of relevant items; and (4) user characteristics, such as experience with the topics and systems. They found that user satisfaction is positively correlated with system effectiveness, but with a weak relationship; that user satisfaction is strongly and positively correlated with user effectiveness, suggesting that incorporating user behaviour into the measurement of search effectiveness is essential; that user satisfaction is negatively correlated with user effort (and this is also in agreement with the recent study by Jiang et al. [108]), suggesting that users are less satisfied when they are obliged to exert themselves; and that no significant relationship was observed between user satisfaction and user characteristics. In conclusion, all the aforementioned studies support the argument that it is possible to measure user satisfaction in an operational setting.

2.5.2 User Feedback for Predicting Satisfaction

Self-reported users satisfaction ratings, however, cannot be observed at scale. In contrast, implicit feedback, such as clicks and query reformulations, can be collected real time from operational systems. A number of studies have proposed various surrogates for self-reported satisfaction using online metrics or implicit feedback. Fox et al. [65] demonstrate that clicks, reading time, and exit type (the way in which user stop reading the result page, such as closing the browser window, or submitting a new query) are the three best factors for predicting satisfaction at the level of individual result page. Similar implicit online metrics again becomes important factors for predicting session-level satisfaction. For example, the time taken to inspect a particular SERP is negatively correlated with session satisfaction.

Feild et al. [63] address the problem of searcher's frustration prediction, and found that features based on query logs are important to this problem, including search duration, the number of unique queries, and information from scrolling actions. Ageev et al. [1] show that session success is linked to the user's effort. They found that successful users issue more queries, view more SERPs, scan SERPs to deeper rank positions, analyse SERP faster, perform click actions faster, and use advanced query syntax more frequently compared to those who are not success. Guo et al. [71] investigate factors that affect search engine switching, and propose a machine learning model to predict the switching behaviour. They found that dissatisfaction and coverage are two main reasons for switching with identical query. In other work, Guo et al. [72] employ information derived from cursor movements before and after click actions to improve the performance of session success prediction. These includes the speed and the coordinates of mouse cursor. In addition to cursor movements, other behaviours are found to be correlated with session success, such as the total number of queries in a session (Pearson's $r = -0.52$), the total number of clicks in a session ($r = -0.36$), the total number of clicks divided by the number of SERP views in a session ($r = 0.36$), and search duration in a session ($r = -0.47$).

Hassan et al. [77] utilise a combination of DCG and Markov model to classify a search goal as either successful or not. A search goal is defined as an action sequence performed by users to address a single information need, and consists of several types of actions, such as query and click actions. In a follow-up work, Hassan [76] proposes a semi-supervised approach to predict whether a search goal leads to a successful goal or not. Hassan et al. [78] show that combining reformulation and click information gives a better performance for query satisfaction prediction, than using each of them individually.

Wang et al. [229] extend the work of Hassan et al. [78] by modelling action-level (query or click) satisfaction to predict the overall session satisfaction. Jiang et al. [108] combine utility metrics (such as, click dwelling time) and effort-based metrics (such as, the number of queries in a session) to predict user-reported 5-point session satisfaction. Liu et al. [140] propose a methodology to extract frequent cursor subsequences from mouse-based action logs, and then use those subsequences to predict session satisfaction. Liu et al. [136] investigate factors that can characterise the difficulty of a particular search tasks, and found that several variables, such as number of viewed items in the first SERP, dwell time on the first SERP, and number of saved documents per query, are significant factors.

2.6 Meta-Evaluation

Meta-evaluation aims at assessing the quality of an effectiveness metric. Cooper [55] argues that user satisfaction is a staple metric of system performance. Based on this argument, some authors have suggested that a good metric should produce scores that strongly correlated with user satisfaction [5, 91, 104, 106, 137, 139, 241].

In addition to satisfaction, user performance and preference are also two key aspects for meta-evaluation. Several studies have investigated the correlation between metrics and user performance, such as the number of relevant documents found by the user [83, 219]. It is also desirable to investigate the extent to which metrics predict user preferences [182, 186].

Other authors contend that a metric should have an accurate user model, reflecting actual user behaviour [20, 39]. There are also comparison-based meta-evaluation approaches, such as the notion of *sensitivity*, which measures how sensitive a metric is to small changes in the ranking [167]. As opposed to the empirical-based approach, axiomatic-based meta-evaluation employs mathematical techniques to decide whether an effectiveness metric satisfies a predefined set of stipulated properties. This section describes existing meta-evaluation approaches for offline search effectiveness metrics, including their classification and the connection between them.

2.6.1 Meta-Evaluation Based on User Satisfaction

A good metric is the one that has a strong positive correlation with user satisfaction. There have been at least two ways of evaluating how well a metric predicts the user satisfaction:

1. Direct computation of correlation coefficients, such as Pearson r, Spearman ρ, or Kendall τ, between metric scores and satisfaction ratings given by users, which can be carried out in both query- and session-levels; and

2. Evaluation of whether or not offline measurements correlate with online metrics that are deemed to reflect user satisfaction.

The first approach is common and adopted by the recent developments of effectiveness metrics for both query- and session-levels [5, 46, 91, 104, 106, 137, 139, 241]. While this meta-evaluation approach is simple (given a set of user satisfaction ratings), it has two drawbacks. First, user-reported satisfaction ratings cannot be obtained at scale. Second, Hufnagel [92] argues that user satisfaction ratings only reflect individual performance, rather than the overall performance of a system. Hufnagel then found that there is a possibility that the user discounts the performance of the system even though it actually performs well.

The second approach provides an alternative in the absence of user satisfaction ratings. Previous work has shown that user satisfaction can be estimated using online indicators, such as the number of clicks in a session and the reciprocal of clicked ranks [46, 167]; and that a sequence of user actions can be used to predict search successfulness [77]; and that there is a relationship between implicit metrics (for example, clickthrough patterns, the total time spent on a SERP, and the patterns of how the user ends a query or a session) and users' explicit satisfaction ratings [65]. Chapelle et al. [44] used clickthrough metrics to evaluate their proposed metric, expected reciprocal rank (ERR). They argue that when the score of a metric is highly correlated with that of clickthrough metrics, it is an indication that the proposed metric captures user satisfaction. In their proposal, they used five click metrics:

1. UCTR – A binary variable indicating whether or not a click was observed in a session;
2. Max RR – The maximum reciprocal clicked ranking position. This metric returns zero if no clicks were observed;
3. Mean RR – The mean reciprocal clicked ranking position. This metric returns zero if no clicks were observed;
4. SS – A similar metric to UCTR, ignoring cases in which the click happened at ranking positions where low-quality documents appeared;
5. PLC – Precision at lowest click. The ratio between the number of clicks and the deepest clicked rank.

2.6.2 Meta-Evaluation Based on User Performance

A number of authors have investigated the relationship between scores generated by effectiveness metrics and user performance. Hersh et al. [83] show that the performance of an instance recall task, measured by the proportion of instances for a particular topic that are identified within a set of documents saved by the user, lacks relationship with AP. Turpin and Hersh [218] compare user performance on baseline and improved systems based on AP, and demonstrate that users did not benefit from extra relevant documents appearing on top-10 rank positions when performing a question-answering task.

Turpin and Scholer [219] used five search systems with five different AP levels (from 55% to 95%) and asked participants to solve a precision-based task (finding a relevant item) and a recall-based task (finding as many relevant documents as possible in five minutes) using those five systems. User performance for the former and the latter tasks are measured by, respectively, the time taken to spot the first relevant document, and the number of relevant documents identified in a five minute period. The results suggest that user performance has a weak relationship with system performance when measured by AP on these two tasks. For the former task, they also show that the relationship between user performance and Prec@K ($K \in \{2, 3, 4, 10\}$) is weak.

Allan et al. [8] investigate the relationship between user effectiveness (measured by time on task, proportion of the correct facets found by users) and system effectiveness (measured by BPref). In general they find that retrieval system effectiveness is aligned with user effectiveness. However, for several specific cases, the relationships are not significant. For example, when BPref improves from 70% to 90%, there is no evidence that the time taken by the task decreases; and when BPref increases from 60% to 70%, there is no significant difference in the proportion of correct facets.

Al-Maskari et al. [6] carry out user studies using two systems with different AP scores (bad system with $\mathsf{AP} = 0.05$ and good system with $\mathsf{AP} = 0.20$), and observe user performance when using these two systems. Five user performance factors are employed: time to find the first relevant document; number of saved relevant documents; number of queries; user satisfaction; and easiness. When users used the bad system, they spent more time to save the first relevant document, saved fewer relevant documents, and issued more queries in the session. With a good system, the same users were more satisfied (as indicated by user-reported 4-point satisfaction ratings), and found that the tasks are easier (as indicated by user-reported 4-point easiness ratings). Al-Maskari et al. also observe that Prec@200 has a stronger relationship with each of the five user performance factors than AP; and that AP has the weakest correlation with satisfaction ($r \approx 0.12$) compared to Prec@K with $K \in 10, 20, 50, 200$ (all with $r \approx 0.30$). This once again suggests that AP lacks correlation with user performance.

Smucker and Jethani [196] construct two different ranked lists with different levels of precision, one with a uniform Precision level of 0.6 (good ranking) and the other with a uniform Precision level of 0.3 (bad ranking), and then observe the changes of user performances on these two rankings. When inspecting the SERP with lower Precision, users tended to spend more time viewing result snippets, and were less likely to click on non-relevant documents. When examining the SERP with higher Precision, the maximum ranks that were viewed by the same users were likely to be shallower. Smucker and Jethani finally conclude that Precision and human performance have a strong relationship, and suggest that when a metric effectively includes a model of user behaviour, the metric correlates better with human performance. Indeed, they argue that the mismatch between existing offline metrics and user performance is due to the assumption that users take a constant time to inspect every document in the ranking. This observation leads to proposal of the metric TBG [195].

2.6.3 Meta-Evaluation Based on User Preference

Thomas and Hawking [210] report preference-based experiments where two rankings with different quality are shown to the same user, side-by-side in the same window. One panel presents top-10 results from Google.com, assumed to be a good set of answers; and the other one displays Google.com's list of results from rank 21 to rank 30, assumed to be a bad set of answers. Even though no effectiveness metric is explicitly used for these rankings, it can be assumed that good rankings have better Prec@10 compared to bad rankings. They found evidence that users tend to prefer the good set of search results over the bad counterpart, both when users were given pre-defined head queries from Google.com (popular queries) and when they were encouraged to use their own topics (natural queries). However, there is no evidence that users preferred the ranking in the left-hand or right-hand panels in either of these two experiments. Similar results were also observed, when the rankings overlapped (Google.com's top-10 results in one panel and results from rank 6 to 15 in the other).

Sanderson et al. [186] investigate the extent to which offline metrics predict user preferences, using sets of topics and subtopics from a diversity task for TREC 2009 Web track. Inspired by the side-by-side presentation method of Thomas and Hawking [210], two sets of top-10 results from a pair of runs are displayed to users (via a crowdsourcing service) who are then asked to indicate which of the two they prefer, if either. They found a significant result in the level of agreement between several diversity measures, such as α-NDCG, and user preferences. Treating each subtopic as a distinct topic, several adhoc metrics (NDCG, RR, ERR, and Prec@10) are also computed and compared with user preferences. The results also suggest that these adhoc metrics are aligned with user preferences with a level of agreement of around 65%.

Sakai and Zeng [182] argue that the work of Sanderson et al. [186] has a weak connection with intentwise graded relevance evaluation. By using a set of topics and intents from NTCIR-9 INTENT-1 task, they compute the agreement level (measured by Kendall's τ) between user preferences and several adhoc measures based on graded relevance. In addition to several adhoc measures (Precision, QM, RBP, NDCG, and ERR), they also introduce two new metrics. The first one is *intentwise rank-biased utility* (iRBU@K), a component of rank biased utility [13], computed as:

$$\mathsf{iRBU@K}(\overrightarrow{r}; \phi, K) = \sum_{i=1}^{K} \left(g(r_i) \cdot \prod_{j=1}^{i} (1 - g(r_j)) \right) \cdot \phi^i \,,$$

where ϕ is the persistence parameter and $g(r_i) = (2^{r_i} - 1)/2^{r_{max}}$ is the gain mapping function used in ERR. The second new metric is a combination between ERR and QM, called *expected blended ratio* [182, p. 597]. The results show that NDCG and iRBU@K have the strongest agreement with user preferences among all adhoc metrics in their experiments. Two metrics, NDCG and iRBU@K ($\phi = 0.99$), predict user preferences with Kendall's $\tau \approx 0.80$. The other metrics, QM, RBP, and expected blended ratio, perform relatively well ($\tau \approx 0.76$). However, ERR has the weakest relationship with user preferences ($\tau \approx 0.60$).

2.6.4 Meta-Evaluation Based on User Model Accuracy

This kind of meta-evaluation focuses on assessing how accurate the user model embedded in a metric is. This generally can be done by computing a *closeness* score, such as squared error or likelihood [20], between the model and actual user behaviour observed from interaction logs. Furthermore, it is also necessary to operationalise the concept of user behaviour, such as via the notion of continuation probability, $C_M(i)$, and last probability, $L_M(i)$. Azzopardi et al. [20] adopted theory from the field of economics and argued that a good metric should reflect the actual user behaviour. Based on the findings by Hassan et al. [77], that user behaviour can be used to predict search successfulness, Azzopardi et al. [20] proposed the following three approaches for meta-evaluation of an effectiveness metric. This method requires clickthrough logs with the corresponding relevance judgements for each query-document pair in each ranking.

1. Stopping rank likelihood – comparing the stopping probability of a user model embodied in a metric with empirical distribution of stopping ranks estimated from clickthrough logs (using mean likelihood). A similar meta-evaluation experiment was also carried out by Carterette [39];

2. Estimating experienced utility – comparing the utility estimated using the proposed user model and the actual utility experienced by the user, by assuming that the user only derived utility from clicked ranking positions (using mean squared error); and

3. Estimating time spent on the SERP page – comparing the expected total time estimated by the proposed user model and the actual time spent observed from the interaction logs.

A major concern is that it is still not clear that a metric that has an accurate user model also correlates with user satisfaction. Use of a range of factors is thus appropriate, anticipating that each will contribute to an overall assessment of any proposed relationship.

2.6.5 Comparison-Based Meta-Evaluation

Other work quantifies the *relative behaviour* of a metric compared to the behaviour of a *reference metric* [39, 151, 167, 177]. The main goal is to understand which one among several metrics is preferable in terms of a specific property or criteria.

Carterette [39] proposed a method to evaluate *robustness* of an effectiveness metric: the extent that the metric produces approximately the same decisions whether they are conditioned on a few versus many topics, or shallow versus deep pools. Radlinski and Craswell [165] compared interleaving-based evaluation [167] with traditional metrics, such as Prec@K, NDCG, and AP, in terms of their *sensitivity* to the small changes in the ranking quality. In terms of *judgement cost*, Moffat and Zobel [151] proposed the notion of *residuals*, the extent of uncertainty due to unjudged documents, to compare judgement effort among weighted-precision metrics.

Sakai [177] used the notion of *discrimination power*, which is defined as the probability that a metric successfully concludes "System A is significantly better than System B (under a certain statistical test)" in a shared task environment (TREC evaluation), to compare the stability of several metrics (see Lu et al. [141] for another definition of discrimination power). Hence, it is difficult to draw conclusions using a metric with a low discrimination power in the experiments. Moreover, the notion of discrimination power has been used in the recent developments of metrics, such as TBG [195] and UM [181]. Buckley and Voorhees [32] used a similar notion when they concluded that AP@1000 is better than Precision.

2.6.6 Axiomatic-Based Meta-Evaluation

The aforementioned meta-evaluation methods are based on empirical experiment, often with the help of statistical methods, to compare one metric with either a reference metric or a gold standard, such as satisfaction ratings. On the other hand, axiomatic-based meta-evaluation requires a set of predefined desirable axiomatic properties. The meta-evaluation process thus involves using any mathematical proofing technique to decide whether or not a particular metric satisfy the properties [10, 11, 12, 64, 149].

Moffat [149] proposed seven possible numerical properties of effectiveness metrics that can be used to compare metrics; and to understand the advantages as well as the drawbacks of metrics through the lens of such properties. For example, one of the seven properties is the *monotonicity* property, which states that if an evaluation depth is extended from K to $(K+1)$, the score should not decrease. Similarly, Ferrante et al. [64] introduced two properties for utility-based metrics: *replacement* and *swap*. The former property states that replacing an item with a more relevant one at the same position in the ranking should not decrease the score, while the latter asserts that swapping a more relevant item in a lower rank position with the less relevant one in a higher position should not decrease the score.

2.7 Summary

We introduced the core concept for the evaluation of information retrieval system, specifically focusing on the importance of incorporating user contexts into offline search effectiveness measures. Section 2.1 provided an overview of test collection-based evaluation paradigm, including its existing effectiveness measures which are mostly defined based on two classic measures, Precision and Recall. We argued that a desirable measure should be computed based on (only) the part of the ranking that has been examined; and that recall-based measures face potential problems, including that they take into account unretrieved documents that have never been inspected by users. We further contended that integrating an accurate user model into precision-based measures would be more useful, rather than focusing on how to accurately estimate recall.

Section 2.2 provided an overview of methodologies for exploring user search behaviours from search interaction logs, gaining insights regarding interaction patterns when users inspect SERPs. Section 2.3 introduced several existing user models through the lens of the C/W/L framework. With this framework, the concept of user behaviour is operationalised via three interrelated hypothetical functions: conditional continuation probability function, weight probability function, and last probability function. Furthermore, the notion of continuation probability, denoted by $C(\cdot)$, is particularly of interest since most of the offline measures can be explained using this function, rather than using the other two functions. Several authors have postulated properties regarding how an ideal $C(\cdot)$ should behave. A classification of existing user models was also introduced in Section 2.4.

Effectiveness metrics need to be assessed. Several aspects for meta-evaluation have been proposed, including user satisfaction. Section 2.5 describes the notion of user satisfaction and its relationship with implicit feedback. Section 2.6 introduced several meta-evaluation strategies for offline effectiveness measures, grouped into six classes, ranging from experimental-based approaches to axiomatic-based ones.

However, a number of issues need to be raised in the development of C/W/L-based effectiveness metrics. First, it is necessary to connect the hypothetical properties regarding $C(\cdot)$ with what is actually observed from the real search interaction logs. Therefore, we develop a methodology for inferring $C(\cdot)$ from interaction logs in Chapter 3, with emphasis on how to infer gaze distribution from clickthrough data and subsequently use the inferred distribution for the computation of the observed $C(\cdot)$. Chapter 4 shows empirical evidence for hypothetical properties of $C(\cdot)$ that have been postulated by Moffat et al. [155]. Second, existing measures mostly focus on scoring a single query with respect to a single information need. However, users often submit more than one query to fulfill a need. Therefore, it is desirable to have a session-based measure that considers a multi-query session as a single scoring unit. In Chapter 4, we introduce a session-based C/W/L framework and develop a new session-based measure based upon it. Finally, we further develop and demonstrate a C/W/L-based meta-evaluation framework in Chapter 5, measuring the relationship between predicted $C(\cdot)$ and observed $C(\cdot)$; and between metric scores and user satisfaction.

Chapter 3

Modelling User Actions

Typical search engine users interact with SERPs via at least two different actions: impressions (views) and clicks. A collection of search actions is thus a key resource to understanding what factors contribute to conditional continuation probability and to calibrating the parameters of user-based search effectiveness metrics. This chapter investigates the relationship between these two kinds of action, particularly the extent to which impressions can be predicted from clicks. The goal of this chapter is to construct tools for inferring observed behaviour from interaction logs, including a framework for inferring impression distributions from click-based logged behaviours and an approach to computation of empirical estimates of continuation probability.

Section 3.1 introduces our motivation and problem statement. Section 3.2 describes the interaction logs used in our analysis and experiments. The estimation of continuation probabilities from interaction logs is critical to the development of an accurate user model. Section 3.3 describes three alternatives for computing empirical conditional continuation probabilities, denoted by $\hat{C}(i)$, from a collection of impression sequences. However, impression data may not always be available from operational search logs. In the absence of impression sequences, it is desirable to be able to infer the impression distributions from clickthrough sequences. This process can be done via an impression model.

The material in Section 3.3 is based on the following published papers:

- Alfan F. Wicaksono and Alistair Moffat. Empirical Evidence for Search Effectiveness Models. In *Proc. CIKM*, pages 1571–1574, 2018.
- Alfan F. Wicaksono. Measuring Job Search Effectiveness. In *Proc. SIGIR*, page 1453, 2019.

The material in Section 3.4 is based on the following published paper:

- Alfan F. Wicaksono and Alistair Moffat. Exploring Interaction Patterns in Job Search. In *Proc. Aust. Doc. Comp. Symp.*, pages 1–8, 2018.

The material in Sections 3.5 and 3.6 is based on the following published paper:

- Alfan F. Wicaksono, Alistair Moffat, and Justin Zobel. Modeling User Actions in Job Search. In *Proc. ECIR*, pages 652–664, 2019.

Section 3.4 explores several interaction patterns in regard to impressions and clicks. The insights gained from that exploration are then employed for the development of impression models in Section 3.5. Finally, Section 3.6 demonstrates how to compute empirical $\hat{C}(i)$ using impression models and shows that the resultant $\hat{C}(i)$ is close to the "true" $\hat{C}(i)$ computed using actual impression sequences.

3.1 Motivation and Research Question

Users' viewing and clicking behaviours may exhibit a variety of search interaction patterns. For example, users may click on the results in the ranking in turn; or they may inspect many results below rank i before deciding to click at rank i; or they may examine a number of results beyond the deepest click rank; or they may not click in a sequential order; or they may perform "one-step jump, and then two-step jump". Modelling of these interaction patterns is a key step for the evaluation of a search engine, and thus can provide guidance for improving its design or effectiveness.

After submitting a query and then obtaining a response from a search engine, users exhibit a series of activities. There are at least two types of activity: viewing the result listed at rank i (that is, an *impression* at rank i) and clicking at rank i. There may be additional types of action in other domains, such as "apply for job" in job search, "download resume" in talent search, "add to wishlist" in product search, and "follow playlist" in music search. A collection of action sequences is then a useful resource for exploration of interaction patterns and for gaining insights about how each of the actions relate to each other.

In particular, impression data is critical to the development of effectiveness metrics based on user model, such as for understanding in what order the user would inspect the ranking, and for investigating factors contributing to conditional continuation probabilities. The first part of this chapter addresses the following research question:

RQ 3.1: How to infer empirical $\hat{C}(i)$ from a collection of impression sequences?

We propose a method for inferring conditional continuation probability from impression sequences, and demonstrate the use of inferred $\hat{C}(i)$ for tuning the parameters of three static user models: SDCG, RBP, and INSQ, and for comparing which among them provides the most accurate $C(i)$. The proposed method will also be used in Chapter 4 for exploring factors that influence $C(i)$.

However, impression sequences may not be observable, while clickthrough sequences almost always are, particularly from commercial search engine logs. In the absence of

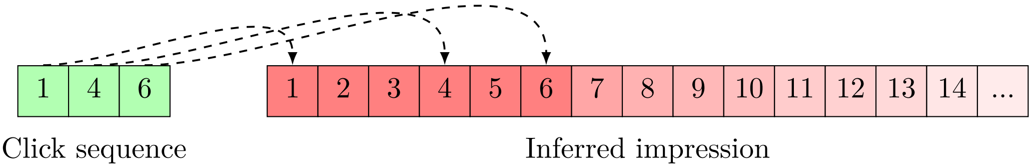


Figure 3.1: Inferring impression distributions from clickthrough sequences. The spectrum of red color on the right-hand side represents the inferred impression probabilities, denoted by $\hat{V}(i \mid u, q)$.

impression sequences, previous work has employed click-based signals, such as the last click rank or the deepest click rank, to mark the end point examined by a user, usually with the assumption that the user inspects the search results in turn from the top. Carterette [39] used the collection of last click ranks over queries to compute a stopping rank distribution across all rank positions. Azzopardi et al. [20, 22] also assumed that the last click rank is the stopping rank, and then demonstrated how to use this assumption for inferring conditional continuation probability from clickthrough logs. Lipani et al. [135] used the deepest click rank, instead of the last click rank, for marking the last point inspected by the user. As is shown by our analysis, below, users often examine results beyond the deepest click rank; and that the last click is not necessarily the last action in the sequence.

The second part of this chapter describes a methodology for inferring the impression distributions from clickthrough actions (that is, an impression model). Let $CR = \langle cr_1, cr_2, \ldots, cr_{|CR|}\rangle$ be a chronologically-ordered sequence of click ranks observed from a particular user and $V(i \mid u, q)$ be the probability that the user u performed an impression action at rank i for query q. An impression model then takes CR as an input and returns an estimate $\hat{V}(i \mid u, q)$ as an output. A possible way to do this is by assuming that the user always inspects all items earlier than the deepest click rank (that is, $\hat{V}(i \mid u, q) = 1$ if $i \leq \max(CR)$) and by imposing probability distributions, $0 \leq \hat{V}(i \mid u, q) \leq 1$, for $i > \max(CR)$. Figure 3.1 illustrates this mechanism with $CR = \langle 1, 4, 6\rangle$.

The approach proposed by Zhang et al. [244] is the only previous work that addresses the development of impression model. They used the average gap between two consecutive clickthroughs to extrapolate $V(i \mid u, q)$ beyond the deepest click rank. However, they did not have resources that would have allowed their models to be validated.

We have access to interaction logs from a popular Australasian job search engine, `Seek.com`. The design of the `Seek.com` user interface (mobile- and desktop-based applications) allows recording of impression sequences from real search users. This impression data is a valuable resource for developing impression models as well as for validating their effectiveness. Hence, the following research question is considered:

RQ 3.2: How to estimate $\hat{V}(i \mid u, q)$ from click logs?

We develop regression-based impression models based on our analysis on a collection of impression sequences. Finally, this new impression model is experimentally compared to previous approaches, including the proposal by Zhang et al. [244], demonstrating that the new model is more accurate than the previous approaches, and is useful for inferring $\hat{C}(i)$ from click logs.

3.2 Action Sequences and Interaction Logs

This section begins by introducing the notion of an *action sequence*, an abstraction that is applied to a sample of interaction logs used in our experiments, and then describe the characteristics and statistics of the experimental interaction logs.

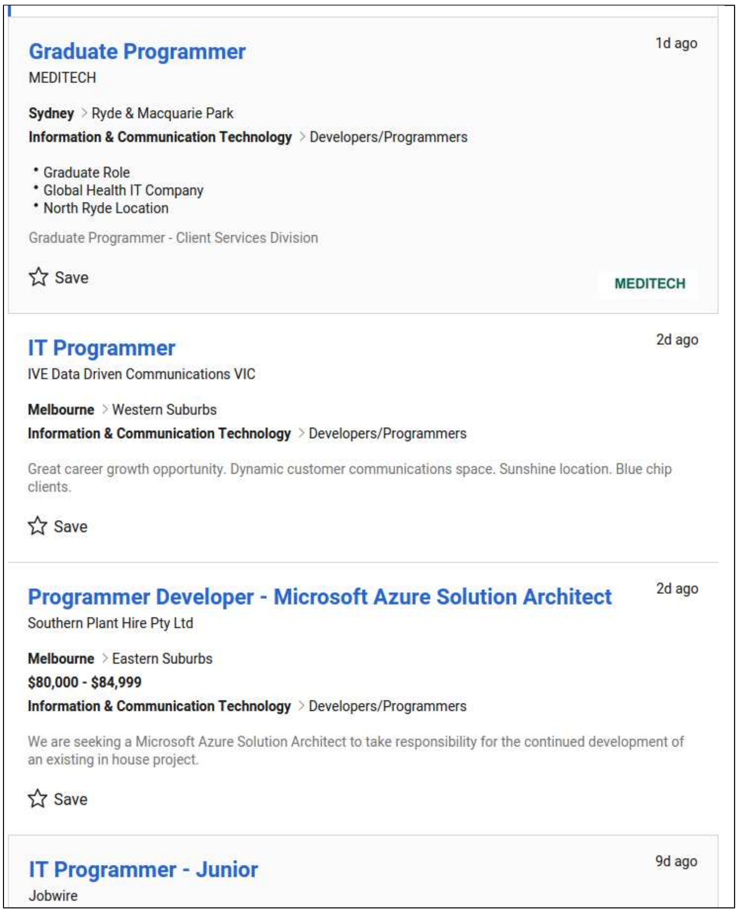
3.2.1 Action Sequences

This study uses a large and rich sample of search interaction logs from a popular job search website, `Seek.com`, as a primary source of observations of user behaviour, providing a basis to develop as well as calibrate a user model for a particular metric. The interaction logs contain a collection of *action sequences*, where an action sequence is an ordered list of post-query actions of a particular user that interacted with a ranking containing a list of search results. Formally, an action sequence is defined as:

$$\mathcal{A} = \langle (a_1, r_1), (a_2, r_2), (a_3, r_3) \ldots \rangle \,,$$

where (a_t, r_t) is an *action* containing two elements: (1) an action type, a_t; and (2) a ranking position at which the action a_t has occured, $r_t \geq 1$.

The behaviour of users is recorded in any of three action types. The first type is *impression* ($a_t =$ "I"), which is recorded when a particular search result is fully visible on screen for 500 milliseconds or more. Note that a sequence of impressions is not the same as a sequence of examinations, since the user might not read an item that is fully visible on screen [245]. With `Seek.com`, a typical screen can show 3 to 4 search results for a general desktop-based website at the same time (see Figure 3.2), and 1 to 2 search results



Graduate Programmer
1d ago
MEDITECH

Sydney > Ryde & Macquarie Park
Information & Communication Technology > Developers/Programmers

- Graduate Role
- Global Health IT Company
- North Ryde Location

Graduate Programmer - Client Services Division

☆ Save
MEDITECH

IT Programmer
2d ago
IVE Data Driven Communications VIC

Melbourne > Western Suburbs
Information & Communication Technology > Developers/Programmers

Great career growth opportunity. Dynamic customer communications space. Sunshine location. Blue chip clients.

☆ Save

Programmer Developer - Microsoft Azure Solution Architect
2d ago
Southern Plant Hire Pty Ltd

Melbourne > Eastern Suburbs
$80,000 - $84,999
Information & Communication Technology > Developers/Programmers

We are seeking a Microsoft Azure Solution Architect to take responsibility for the continued development of an existing in house project.

☆ Save

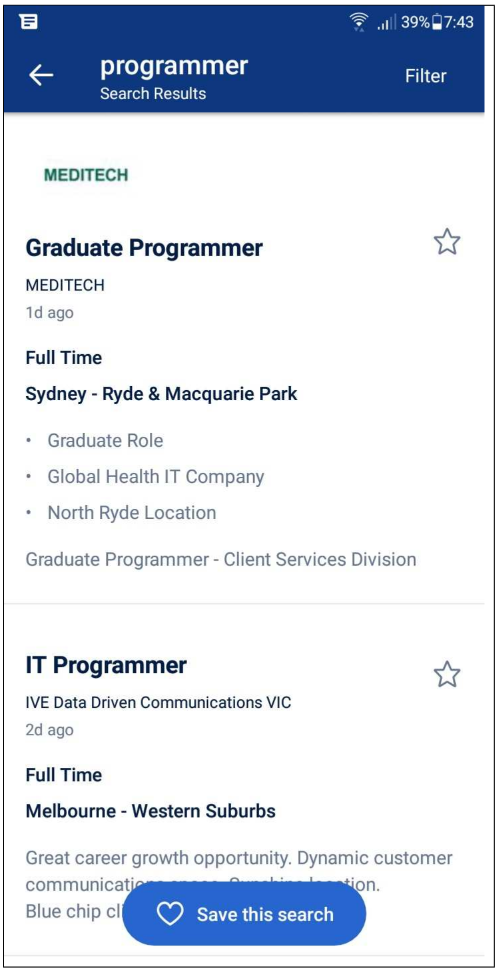
IT Programmer - Junior
9d ago
Jobwire

Figure 3.2: A SERP containing a ranked list of job snippets, generated by `Seek.com` on 2020-05-27, for the query “programmer”. This screenshot was taken on a desktop-based browser.

for a mobile-based application (see Figure 3.3). By looking at how `Seek.com` designed its job search interface, a large set of impression sequences is then a precious resource for exploring and modelling user behaviour, especially from the perspective of *conditional continuation probability* [155].

The second type of action is *clickthrough* (a_t = “C”), which is associated with an event where the user clicks at a particular search result in the ranking so as to reach the corresponding job details page. The third one is *application* (a_t = “A”), which occurs when the user presses the “apply for this job” button in the details page with the expectation that they may lodge an application sometime in the future.

In the context of Web search, *clickthrough* data has served as a valuable feature for predicting relevance [167] and provides a reasonably good surrogate for user satisfaction [46]; while, in the context of job search, *clickthroughs* and *applications* together serve as useful



programmer
Search Results

MEDITECH

Graduate Programmer
MEDITECH
1d ago

Full Time
Sydney - Ryde & Macquarie Park

- Graduate Role
- Global Health IT Company
- North Ryde Location

Graduate Programmer - Client Services Division

IT Programmer
IVE Data Driven Communications VIC
2d ago

Full Time
Melbourne - Western Suburbs

Great career growth opportunity. Dynamic customer
communicati ion.
Blue chip cl

Figure 3.3: A SERP containing a ranked list of job snippets, generated by `Seek.com` on 2020-05-27, for the query “programmer”. This screenshot was taken on an Android-based `Seek.com` application.

proxies for measuring the quality of SERPs [134, 174, 183]. In addition to action sequences, our interaction logs contain other useful information, such as *user id*, *query terms*, *channel* (browser-based web application or mobile application), and *device type* (desktop, phone, and tablet).

To give an illustration of an action sequence, consider the following example:

$$\begin{aligned}\mathcal{A}_1 \quad = \quad & \langle(\text{“I”},1),(\text{“I”},2),(\text{“I”},3),(\text{“I”},4),(\text{“I”},3),(\text{“C”},3),\\ & (\text{“A”},3),(\text{“I”},5),(\text{“I”},6),(\text{“I”},7),(\text{“C”},7),\\ & (\text{“I”},6),(\text{“I”},5),(\text{“I”},7),(\text{“I”},8),(\text{“C”},8),(\text{“A”},8)\rangle\,.\end{aligned}$$

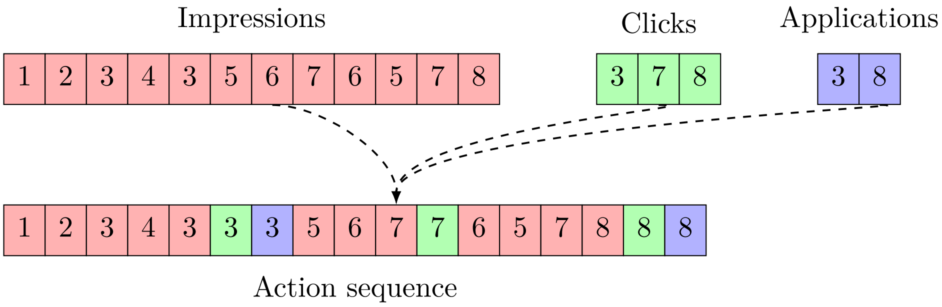


Figure 3.4: An action sequence as the interleaving of impression, click, and application sub-sequences.

In this action sequence $\mathcal{A}$ there are three clicks, at ranks 3, 7, and 8, as well as two job application actions, at rank 3 and 8; and the user has viewed the results from rank 1 to rank 4, and then gone backward to rank 3 before clicking it. The user finally ended the query by viewing, clicking, and applying for the job listed at rank 8. Each action sequence can also be thought of as the combination of three types of sequence: *impression sequence*, *clickthrough sequence*, and *application sequence*, as illustrated in Figure 3.4.

3.2.2 Interaction Logs

Table 3.1 provides a summary of the two search interaction logs used in this study. The desktop browser-based SERP is partitioned into several pages, where each consists of 20 results. Meanwhile, the mobile-based SERPs have no pagination, allowing users to continuously scroll through results pages. Note that a pagination is associated with an interface design, at which users need to click the "next" button usually located at the bottom of any page in order to shift their attentions to the next result page. That is, moving to the next page needs a clicking effort. Figure 3.5 (page 81) shows a page navigation tool generated by the `Seek.com` browser-based Web application. Although the mobile-based application does not have paginations, it has small page-boundary marks. However, mobile-based application users can still smoothly scroll down to the next page.

We argue that these interaction logs, which capture footprints of user behaviour "from the wild", have potential use in IR evaluation. This study focuses on using the data: (1) to understand the variability of user behaviours from the perspective of conditional continuation probability; and (2) to develop and calibrate a user model. The proposed user model will then be incorporated into an effectiveness metric that calculates the usefulness of search engine results pages. Note that the sample of interaction logs used in our ex-

	iOS/Android	browser
Users	$5,003$	$5,107$
Queries, or Action Sequences	$74,475$	$54,341$
SERP size	unlimited	paginated, 20

Table 3.1: Dataset used in this study, consisting of representative samples drawn from `Seek.com` search interaction logs for a 2-month period (30 July 2018 to 23 September 2018), with two modalities, iOS/Android-based vs desktop browser-based queries. Note that SERPs containing "paid items" are excluded.

periments do not contain any personally identifiable information (PII). Two potentially sensitive fields, query terms and document content, are also not considered in this study.

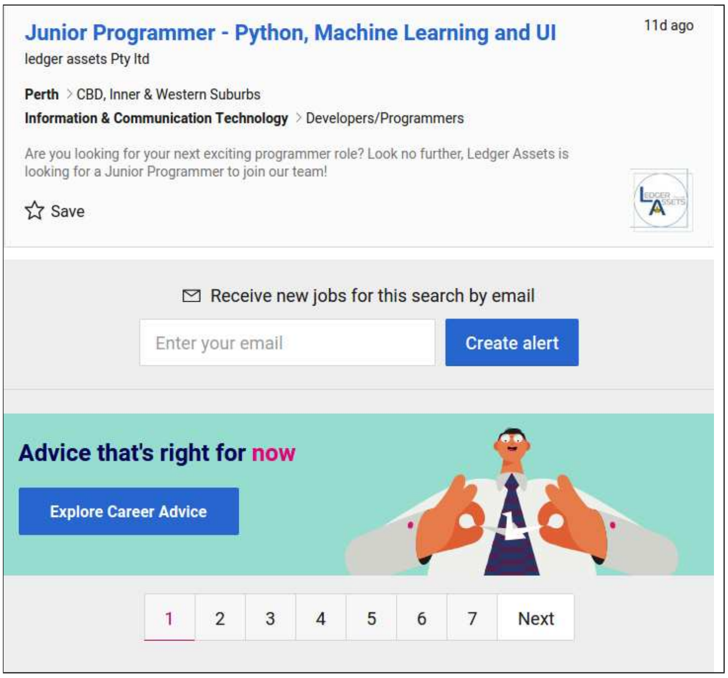
3.3 Inferring Continuation Probability

Recall that the concept of user behaviour can be operationalised by any of three interrelated characteristics: continuation probability, $C(i)$; weight probability function, $W(i)$; and last probability, $L(i)$. The continuation probability is of particular interest because of its use in the development of effectiveness metrics [20, 22, 155]. A computation of empirical $\hat{C}(i)$ from observation data is key to modelling $C(i)$ itself, since it is critical for investigating factors that potentially affect predicted $C(i)$ (such as the work carried out by Moffat et al. [153]) and for measuring the accuracy of predicted $C(i)$. Here we address **RQ 3.1** (page 74), and propose three operational definitions of *continuation* and develop a method for computing empirical $\hat{C}(i)$ from impression sequences recorded from real search users.

3.3.1 Computing Empirical $C(i)$

Notation. The interaction logs described in Table 3.1 contains a collection of user IDs $U = \{u_1, u_2, \ldots, u_{|U|}\}$, where each user ID corresponds to a set of impression sequences $\mathcal{P}(u_i) = \{P_1, P_2, \ldots, P_{|\mathcal{P}(u_i)|}\}$, the lists of search results they examined when conducting search activities.

The Empirical Estimate of $C(\cdot)$. An empirical estimation of the conditional continuation probability at rank i, denoted by $\hat{C}(i)$, is computed by accumulating a numerator, $N(i, P)$, and a denominator, $D(i, P)$, on a per-impression sequence basis. Two options



Junior Programmer - Python, Machine Learning and UI

11d ago

ledger assets Pty ltd

Perth > CBD, Inner & Western Suburbs

Information & Communication Technology > Developers/Programmers

Are you looking for your next exciting programmer role? Look no further, Ledger Assets is looking for a Junior Programmer to join our team!

Receive new jobs for this search by email

Enter your email

Advice that's right for now

Figure 3.5: Page navigation buttons at the bottom of a browser-based search result page generated by `Seek.com` on 2020-06-01, for the query “programmer”.

are then considered for aggregating the per-sequence numerator and denominator over all impression sequences:

1. a micro-average value across users and queries,

$$\hat{C}(i) = \frac{\sum_{u \in U} \sum_{P \in \mathcal{P}(u)} N(i, P)}{\sum_{u \in U} \sum_{P \in \mathcal{P}(u)} D(i, P)}\,, \tag{3.1}$$

2. or a macro-average value across users,

$$\hat{C}(i) = \frac{1}{|U'(i)|} \sum_{u \in U'(i)} \frac{\sum_{P \in \mathcal{P}(u)} N(i, P)}{\sum_{P \in \mathcal{P}(u)} D(i, P)}\,, \tag{3.2}$$

   where $U'(i)$ is a collection of users who inspected rank i at least once, or $U'(i) = \{u \in U \mid \sum_{P \in \mathcal{P}(u)} D(i, P) > 0\}$.

We also propose three alternatives for defining a *continuation*, resulting in three rules for computing the numerator and denominator, as shown in Table 3.2. Rule “L” states that all impressions in the sequence are continuation, except the final one. The second alternative, rule “M” records all occurrences of the maximum rank as being non-continuations. Finally,

Rule	$N(i,P)$ and $D(i,P)$
L	$N_L(i,P) = \sum_{k=1}^{n(P)} \mathcal{I}(p_k = i \wedge k < n(P))$; $D_L(i,P) = \sum_{k=1}^{n(P)} \mathcal{I}(p_k = i)$.
M	$N_M(i,P) = \sum_{k=1}^{n(P)} \mathcal{I}(p_k = i \wedge p_k < \max_{1\leq j\leq n(P)} p_j)$; $D_M(i,P) = \sum_{k=1}^{n(P)} \mathcal{I}(p_k = i)$.
G	$N_G(i,P) = \sum_{k=1}^{n(P)} \mathcal{I}(p_k = i \wedge p_k < \max_{k<j\leq n(P)} p_j)$; $D_G(i,P) = \sum_{k=1}^{n(P)} \mathcal{I}(p_k = i)$.

Table 3.2: Three rules for the operational definition of *continuation* in the impression sequence $P = \langle p_1, p_2, \ldots, p_{n(P)}\rangle$, with $\mathcal{I}(\mathit{expr})$ being an indicator function that returns 1 if *expr* is true and 0 if not.

rule "G", a combination of the first two, assigns continuation to any impression in the sequence that is succeeded by one at a higher ranking position.

For instance, consider $P_1 = \langle 1, 2, 1, 4, 5, 6, 1, 3, 4, 6, 5\rangle$. Rule "L" assigns non-continuation to the last instance of "impression at rank 5" in the sequence P_1, resulting in:

$$\begin{aligned}
N_L(1,P_1) = D_L(1,P_1) &= 3\,,\\
N_L(2,P_1) = D_L(2,P_1) &= 1\,,\\
N_L(3,P_1) = D_L(3,P_1) &= 1\,,\\
N_L(4,P_1) = D_L(4,P_1) &= 2\,,\\
N_L(5,P_1) = 1 \text{ and } D_L(5,P_1) &= 2\,,\\
N_L(6,P_1) = D_L(6,P_1) &= 2\,.
\end{aligned}$$

If rule "M" was applied, all instances of "impression at rank 6" are non-continuations. Hence, rule "M" results in:

$$\begin{aligned}
N_M(1,P_1) = D_M(1,P_1) &= 3\,,\\
N_M(2,P_1) = D_M(2,P_1) &= 1\,,\\
N_M(3,P_1) = D_M(3,P_1) &= 1\,,\\
N_M(4,P_1) = D_M(4,P_1) &= 2\,,\\
N_M(5,P_1) = D_M(5,P_1) &= 2\,,\\
N_M(6,P_1) = 0 \text{ and } D_M(6,P_1) &= 2\,.
\end{aligned}$$

i	$N(.), D(.)$	Sequence											Total
		1	2	1	4	5	6	1	3	4	6	5	
1	$N(.)$	+1		+1				+1					3
	$D(.)$	+1		+1				+1					3
2	$N(.)$		+1										1
	$D(.)$		+1										1
3	$N(.)$								+1				1
	$D(.)$								+1				1
4	$N(.)$				+1					+1			2
	$D(.)$				+1					+1			2
5	$N(.)$					+1							1
	$D(.)$					+1						+1	2
6	$N(.)$												0
	$D(.)$						+1				+1		2

Table 3.3: Computations of both $N(i, P)$ and $D(i, P)$, accumulated by iterating over impressions (from left to right) in the sequence $P_1 = \langle 1, 2, 1, 4, 5, 6, 1, 3, 4, 6, 5\rangle$ using the rule "G".

Finally, Table 3.3 shows an illustration for the computations of $N(i, P_1)$ and $D(i, P_1)$ if rule "G" was employed.

To give an illustration of how to compute $\hat{C}(\cdot)$, consider the following *dummy* dataset containing two sets of impression sequences from two users u_1 and u_2:

$$\begin{aligned}\mathcal{P}(u_1) &= \{\langle 1,2,1,4,5,6,1,3,4,6,5\rangle, \langle 1,2\rangle, \langle 1,3,5,4\rangle\}\,, \\ \mathcal{P}(u_2) &= \{\langle 1,2,3,4,3,2,1\rangle, \langle 1,3,1,4,2\rangle\}\,.\end{aligned}$$

Note that the first impression sequence from user u_1 is the same instance as the sequence P_1 used in Table 3.3. If rule "G" was employed with the micro-averaging aggregation method, the $\hat{C}(i)$ values for some rank positions are:

$$\begin{aligned}\hat{C}(1) &= \frac{(3+1+1)+(1+2)}{(3+1+1)+(2+2)} = 0.889\,, \\ \hat{C}(2) &= \frac{(1+0+0)+(1+0)}{(1+1+0)+(2+1)} = 0.400\,,\ \text{and} \\ \hat{C}(3) &= \frac{(1+0+1)+(1+1)}{(1+0+1)+(2+1)} = 0.800\,.\end{aligned}$$

However, if the same five impression sequences were to be processed using the same rule, but with macro-averaging aggregation across two users, the results would be:

$$\begin{aligned}
\hat{C}(1) &= \frac{1}{2}\left(\frac{(3+1+1)}{(3+1+1)} + \frac{(1+2)}{(2+2)}\right) = 0.875\,, \\
\hat{C}(2) &= \frac{1}{2}\left(\frac{(1+0+0)}{(1+1+0)} + \frac{(1+0)}{(2+1)}\right) = 0.417\,, \text{ and} \\
\hat{C}(3) &= \frac{1}{2}\left(\frac{(1+0+1)}{(1+0+1)} + \frac{(1+1)}{(2+1)}\right) = 0.833\,.
\end{aligned}$$

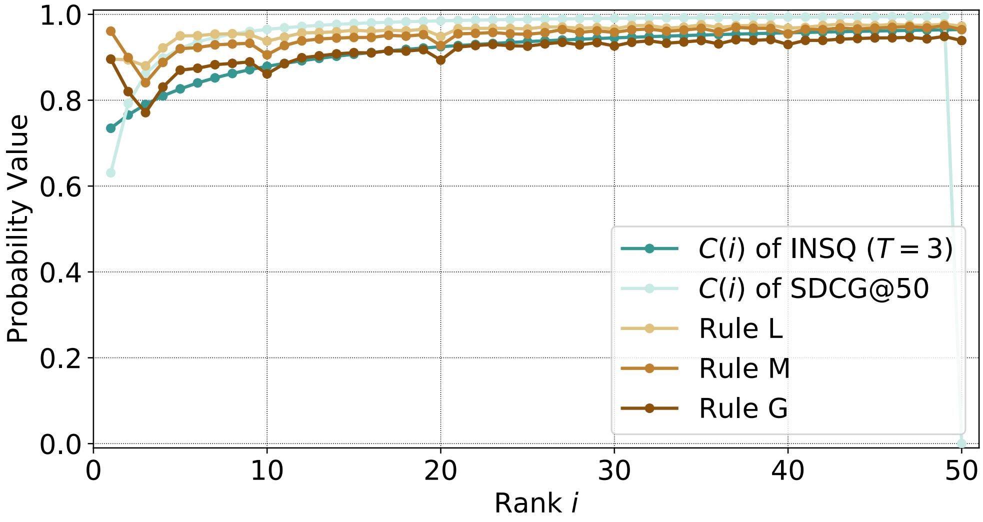
Recall that the dataset used in this study contains 74,475 mobile-based action sequences and 54,341 browser-based action sequences (see Table 3.1 on page 80). This dataset was then employed to compute empirical $\hat{C}(i)$ for $1 \le i \le 50$. Top-50 is particularly interesting, since user behaviours at top two page boundaries can be observed, and the deepest impression rank positions of around 95% of mobile- and browser-based action sequences are less than 50. Figures 3.6 and 3.7 show the plots of empirical $\hat{C}(i)$ functions for iOS/Android and browser-based queries, respectively, across top-50 rank positions. For mobile-based search activities, it can be seen that $\hat{C}(i)$ generally increases with rank i for all rules (Table 3.2) and all aggregation methods. The plots of $\hat{C}(i)$ for the browser-based searchers also reveal a similar pattern, but with obvious significant drops at page boundaries, suggesting that users are disinclined to click the "next page" button at the end of each page. These results demonstrate that the three operational definitions of continuation (rules "L", "M", and "G") are consistent in the sense that they provide the same general behavioural pattern. Note that this result provides empirical corroboration for the "sunk cost" property proposed by Moffat et al. [153, 155].

The broad trend depicted in Figures 3.6 and 3.7 also serves as an empirical evidence for a hypothesis postulated by Moffat et al. [153], which states that the continuation function $C(i)$ is positively correlated with rank position i. Several user models, such as SDCG@K (a scaled version of DCG@K [102]) with $i < K$ and INSQ [152] comply with this hypothesis. While SDCG employs a log-harmonic sequence to model the increase of $C(i)$ with i (for $i < K$):

$$\mathbf{C}_{\mathsf{SDCG}} = \left\{\frac{\log(2)}{\log(3)}, \frac{\log(3)}{\log(4)}, \frac{\log(4)}{\log(5)}, \frac{\log(5)}{\log(6)}, \ldots\right\} = \{0.63, 0.79, 0.86, 0.90, \ldots\}\ ,$$

the definition of INSQ employs a quadratic version of hyper-harmonic sequence ($T = 3$):

$$\mathbf{C}_{\mathsf{INSQ}} = \left\{\left(\frac{6}{7}\right)^2, \left(\frac{7}{8}\right)^2, \left(\frac{8}{9}\right)^2, \left(\frac{9}{10}\right)^2, \ldots\right\} = \{0.73, 0.77, 0.79, 0.81, \ldots\}\ .$$



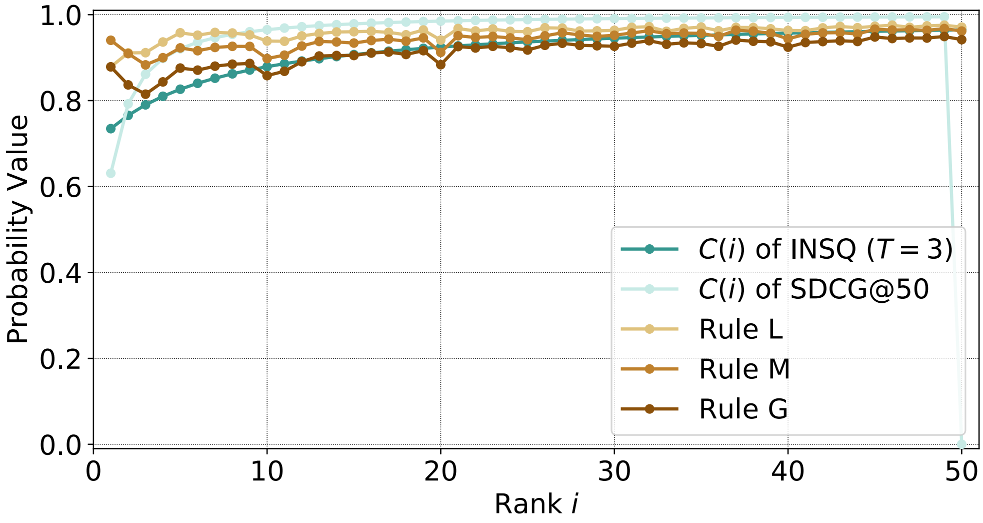
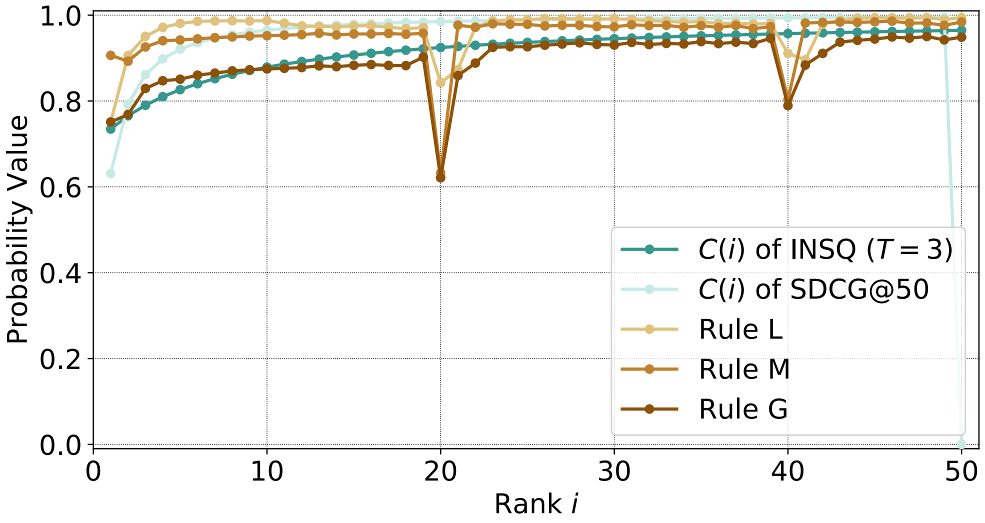


Figure 3.6: Observed $\hat{C}(i)$ for iOS/Android-based queries across top-50 rank positions, computed using three different rules, and then micro- (top) and macro-averaged (bottom) from the `Seek.com` impression sequences. The plots of $C(i)$ for two static user models, SDCG@50 and INSQ with $T = 3$, are also shown for reference.



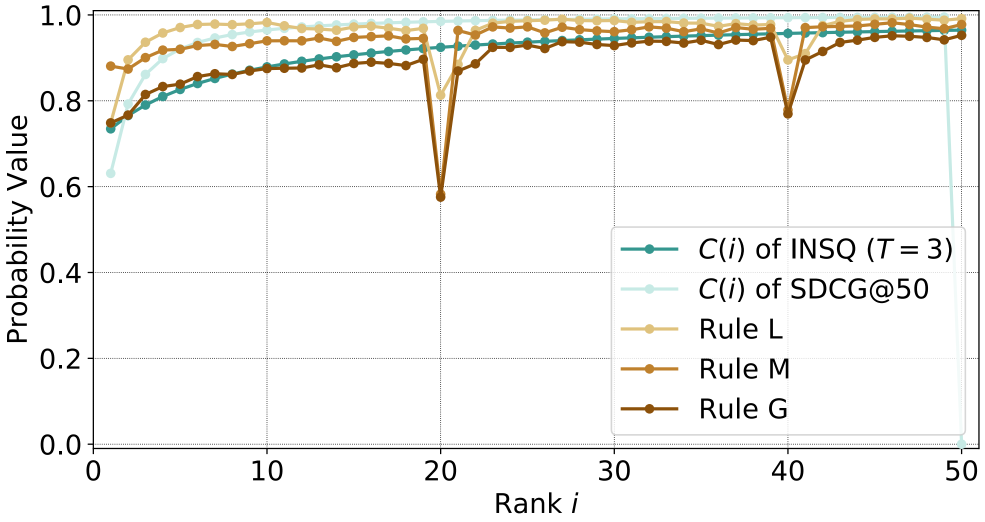


Figure 3.7: Observed $\hat{C}(i)$ for desktop browser-based queries across top-50 rank positions, computed using three different rules, and then micro- (top) and macro-averaged (bottom) from the `Seek.com` impression sequences. $C(i)$ plots for two static user models, SDCG@50 and INSQ with $T = 3$, are also shown for reference. Recall that $C(i)$ relates to progressing past rank i, and that the spikes at ranks 20 and 40 relate to page boundaries. For example, $C(20)$ is the probability of the user shifting from rank 20 (the last result on page 1) to rank 21 (the first result on page 2).

At the beginning, it is clear that $C_{\mathsf{INSQ}}(1) > C_{\mathsf{SDCG}}(1)$. However, starting from rank position 2, the $C(i)$ of SDCG tends to increase with i faster than that of INSQ.

3.3.2 Predicted $C(i)$ Versus Empirical $\hat{C}(i)$

To measure the closeness of predicted $C(i)$ for a particular user model and empirical $\hat{C}(i)$ computed from the data, we propose to use *weighted mean squared error* (WMSE(., .)):

$$\mathsf{WMSE}(\hat{\mathbf{C}}, \mathbf{C}) = \sum_{i=1}^{N} w_i \cdot \left[C(i) - \hat{C}(i)\right]^2 , \tag{3.3}$$

where $w_i = f_i / (\sum_j f_j)$ is the weight associated with the relative frequency of the item at rank position i being inspected by users computed from the observation data, and sums to one, $\sum_i w_i = 1$; and N is the evaluation depth. The weighting is required because $C(i)$ itself is not a distribution, and the errors generated from deeper rank positions, which usually have low empirical support, should contribute less to the overall error score, compared to those from earlier positions.

This study considers three offline metrics whose user models are non-adaptive (INSQ, SDCG, and RBP), and optimise their parameters by minimising WMSE(., .). Both INSQ and SDCG were chosen because one of their key assumptions is consistent with the general trend observed in `Seek.com` data (Figures 3.6 and 3.7), while RBP was considered since previous work demonstrated that its user model is closely aligned with clickthrough-based observed behaviour [44, 244].

Tables 3.4 (page 88), 3.5 (page 89), and 3.6 (page 93) show the best-fit parameters with their corresponding WMSE values using rules "L", "M", and "G", respectively. As can be seen, INSQ provides the best fit among these three static user models, with RBP also closer to the observed $\hat{C}(i)$ than SDCG, with micro- and macro-average methods giving similar estimation results, particularly for rule "G". These experimental results also suggest that users who inspect paginated SERPs tend to be less persistent than those who examine result pages with infinite scrolling.

To conclude, our experiment results validate an important interaction pattern, suggesting that $C(i)$ should increase with rank position i. The next section (Section 3.4) considers insight gained from other interaction patterns, such as impression and click ordering, positional distribution of clickthroughs, impressions prior to clickthroughs, and actions beyond the deepest clickthrough.

Model	iOS/Android		browser	
	parameter	WMSE	parameter	WMSE
SDCG	$k = 51$	0.76×10^{-2}	$k = 51$	0.46×10^{-2}
RBP	$\phi = 0.94$	0.10×10^{-2}	$\phi = 0.94$	0.51×10^{-2}
INSQ	$T = 10.14$	0.04×10^{-2}	$T = 11.52$	0.42×10^{-2}

Model	iOS/Android		browser	
	parameter	WMSE	parameter	WMSE
SDCG	$k = 51$	0.75×10^{-2}	$k = 51$	0.40×10^{-2}
RBP	$\phi = 0.94$	0.07×10^{-2}	$\phi = 0.94$	0.50×10^{-2}
INSQ	$T = 10.70$	0.03×10^{-2}	$T = 9.85$	0.39×10^{-2}

Table 3.4: Best-fit parameters for three static user models computed by minimising $\mathsf{WMSE}(\hat{\mathbf{C}}, \mathbf{C})$ across top-50 rank positions. This computation employs Rule "L" with micro- (top) and macro-averaging (bottom).

3.4 Exploring Interaction Patterns

A second contribution of this chapter is a method for computation of impression distributions from clickthrough sequences. The development of this method requires insights gained from a collection of action sequences. This section explores interaction patterns in regard to impressions and clickthroughs with emphasis on the extent to which the former can be inferred from the latter. This section also examines empirical evidence for several assumptions in the development of existing metrics.

3.4.1 Impression and Clickthrough Orderings

One of the key assumptions underlying many effectiveness metrics is that users inspect the ranking from top to bottom one by one, starting from the item at rank position 1, then examining ranks 2, 3, and so on, until they stop searching. Joachims et al. [110], and Cutrell and Guan [59] validated this *cascade* assumption using user studies based on eye tracking in a lab-experiment setting. This study employs a large sample from commercial interaction logs, as opposed to the lab-based eye tracking data, to re-examine the cascade assumption. Recall that the notion of *impression* in `Seek.com` interaction logs (defined in Section 3.2 on page 76) serves as a primary resource for understanding viewing behaviour.

As an initial step to explore viewing behaviour, it is desirable to compute the distribution of *impression jumps* across all impression sequences in the dataset for both mobile-

Model	iOS/Android		browser	
	parameter	WMSE	parameter	WMSE
SDCG	$k = 51$	1.15×10^{-2}	$k = 51$	1.10×10^{-2}
RBP	$\phi = 0.93$	0.12×10^{-2}	$\phi = 0.94$	0.21×10^{-2}
INSQ	$T = 8.30$	0.08×10^{-2}	$T = 10.01$	0.18×10^{-2}

Model	iOS/Android		browser	
	parameter	WMSE	parameter	WMSE
SDCG	$k = 51$	1.09×10^{-2}	$k = 51$	0.99×10^{-2}
RBP	$\phi = 0.93$	0.05×10^{-2}	$\phi = 0.92$	0.26×10^{-2}
INSQ	$T = 8.03$	0.04×10^{-2}	$T = 7.34$	0.22×10^{-2}

Table 3.5: Best-fit parameters for three static user models computed by minimising $\mathsf{WMSE}(\hat{\mathbf{C}}, \mathbf{C})$ across top-50 rank positions. This computation employs Rule "M" with micro- (top) and macro-averaging (bottom).

and browser-based queries. Given an impression sequence $P = \langle p_1, p_2, \ldots, p_{n(P)} \rangle$, an impression jump is defined as $p_{k+1} - p_k$, with $1 \leq k < n(P)$. For example, consider the impression sequence $P_2 = \langle 1, 2, 3, 5, 4, 2, 4, 7, 6, 5 \rangle$. Table 3.7 (page 93) shows the computation of the distribution of impression jumps for P_2.

Figure 3.8 (page 90) shows this distribution inferred from `Seek.com` data, with the y-axis rendered in a logarithmic scale. The version with linear y-axis is depicted in Figure 3.9 (page 91). Each bar is divided into three components. First, "previously seen" is for an impression jump to a rank that had been seen before. The jumps $\langle 4, 2 \rangle$, $\langle 2, 4 \rangle$, and $\langle 6, 5 \rangle$ extracted from the previous example impression sequence $P_2 = \langle 1, 2, 3, 5, \underline{\mathbf{4}, \mathbf{2}, \mathbf{4}}, 7, \underline{\mathbf{6}, \mathbf{5}} \rangle$ are two instances from this category since ranks 4 and 5 had been inspected previously. Second, "non-sequential new" represents an impression jump to a rank that had not been examined, and is not one greater than the previous maximum rank inspected. In the example sequence P_2, the jumps $\langle 3, 5 \rangle$, $\langle 5, 4 \rangle$, $\langle 4, 7 \rangle$, and $\langle 7, 6 \rangle$ are from this category. Finally, "sequential new" is a movement to a rank position that follows the previous maximum rank inspected. For this category, the jumps $\langle 1, 2 \rangle$ and $\langle 2, 3 \rangle$ are the examples from P_2.

In general, "+1" impression jumps dominate the distribution, suggesting that users tend to scan down the ranking in a sequential manner. This justification is also reinforced by the fact that the "sequential new" cases are the majority in the "+1" bar. That is, when an item is inspected for the first time, it is most likely the next one in impression sequence that has not been examined previously. The fact that "–1" jumps are the second

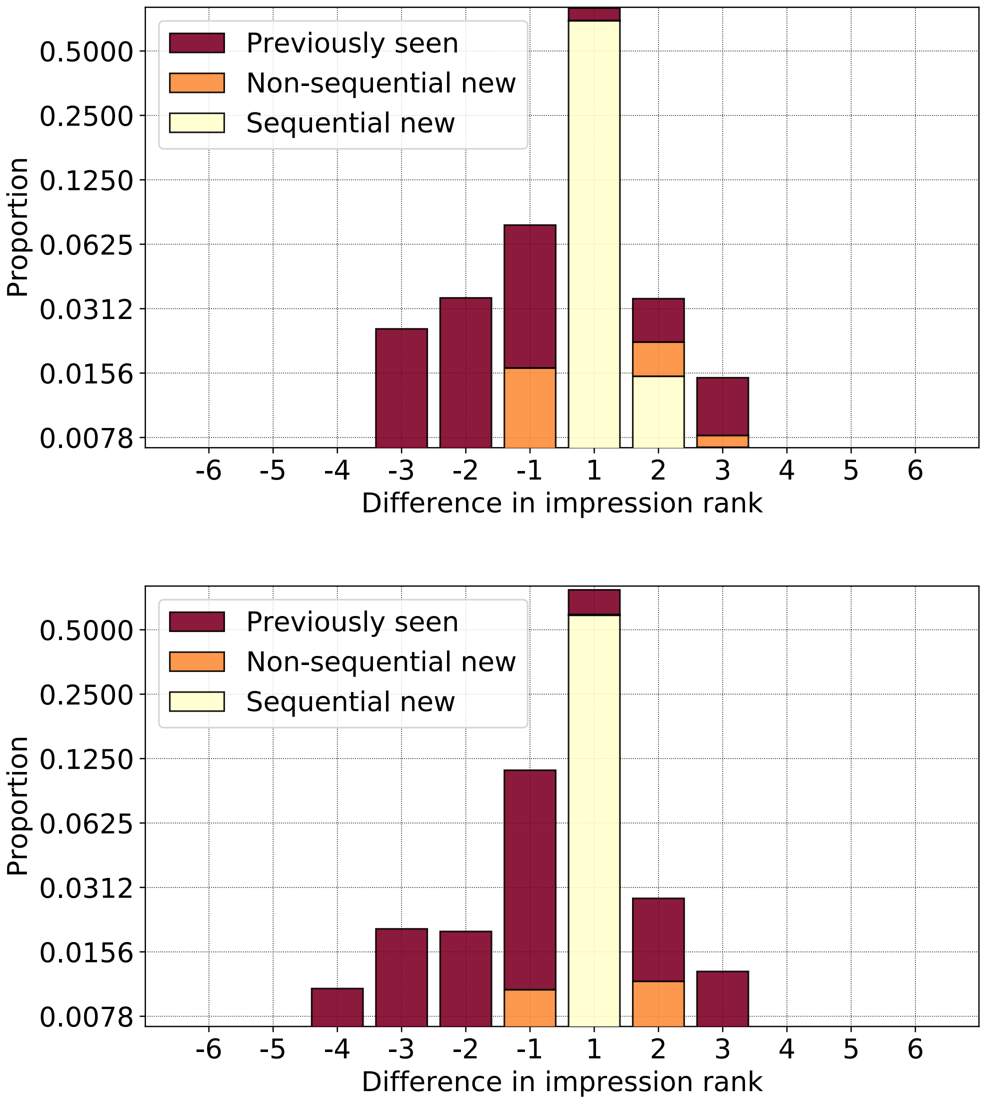


Figure 3.8: Distribution of impression jumps, with the y-axis rendered in a logarithmic scale. The top pane is for iOS/Android-based queries, and bottom pane for browser-based queries.

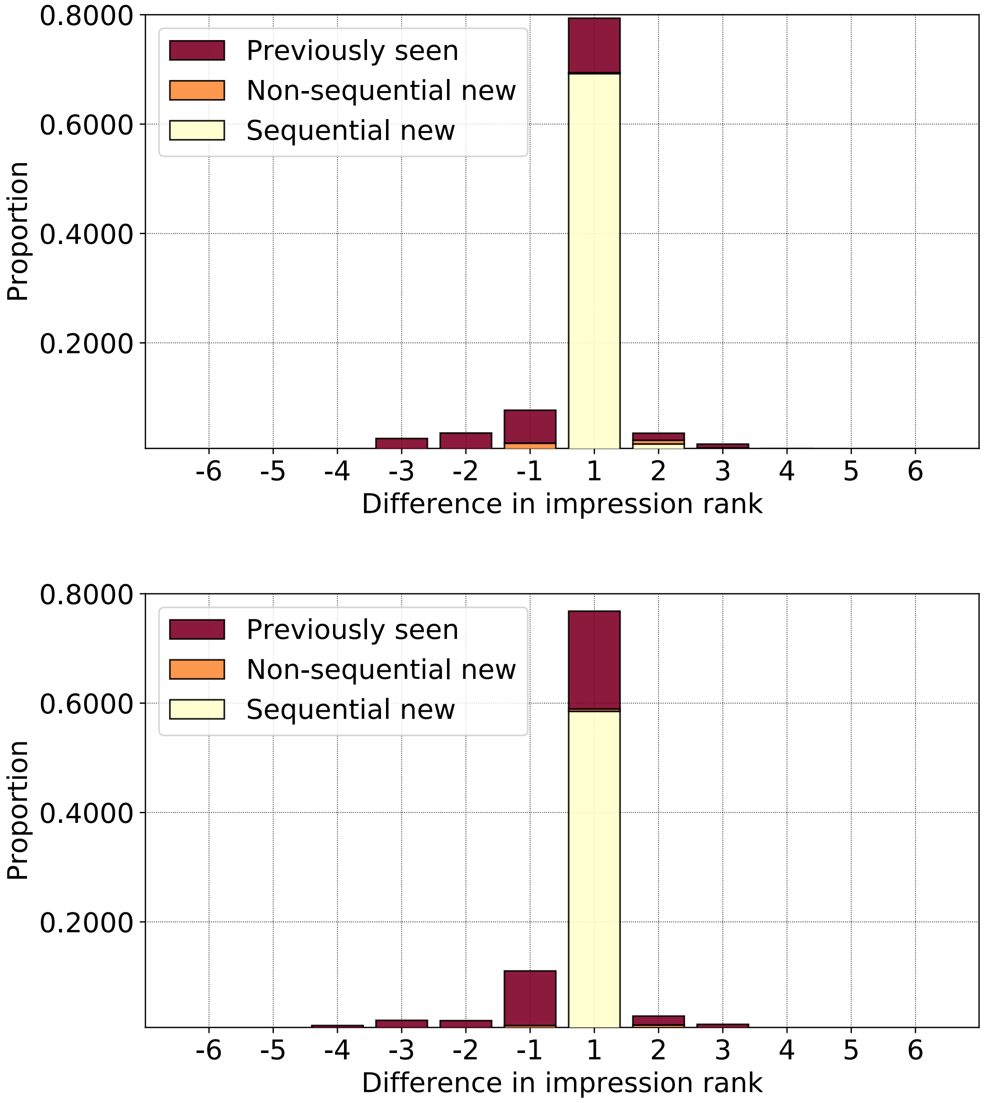


Figure 3.9: The same distribution of impression jumps as described in Figure 3.8, but with the y-axis rendered in a linear scale. The top pane is for iOS/Android-based queries, and bottom pane for browser-based queries.

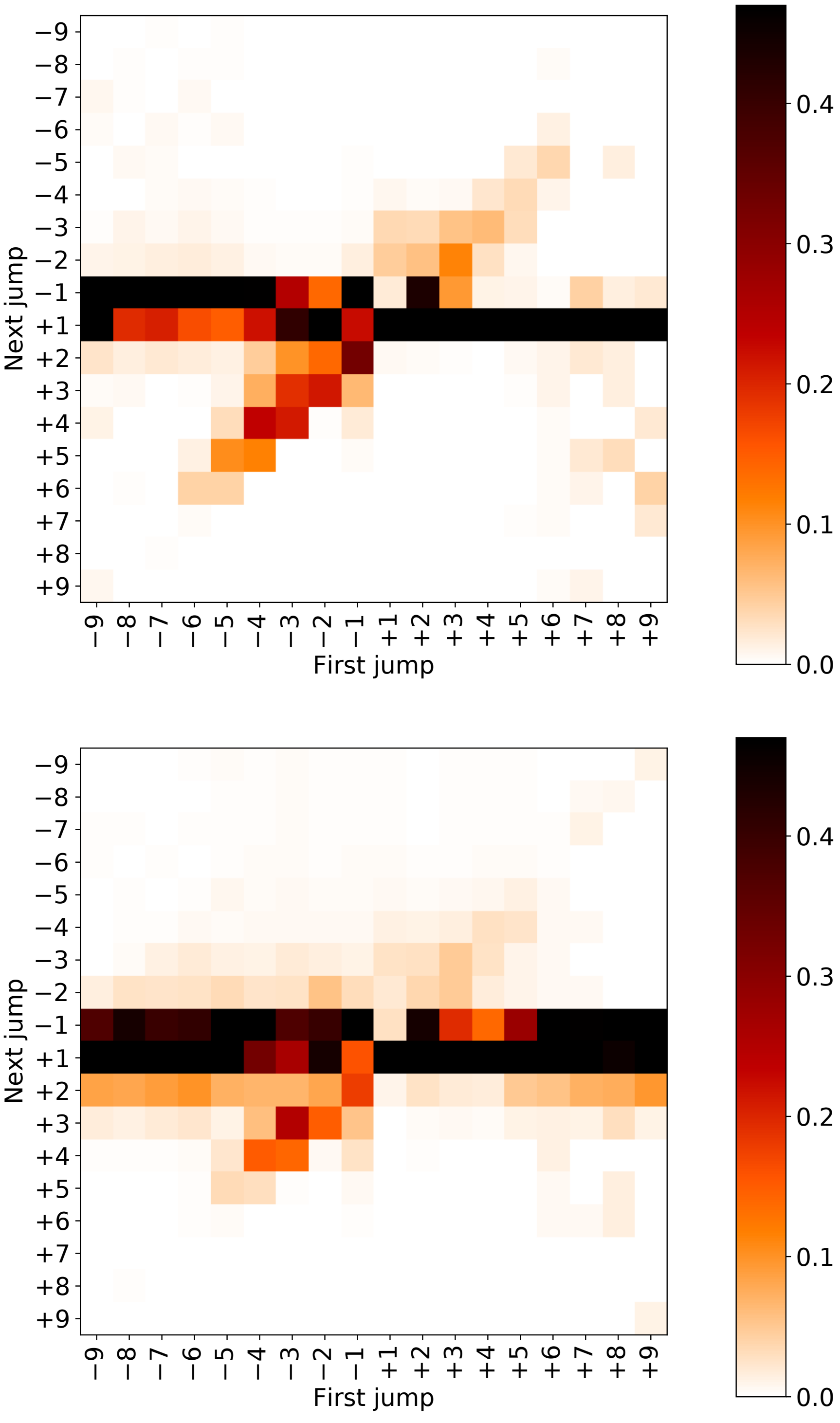


Figure 3.10: Probability of the next jump (y-axis) conditioned on the first jump (x-axis). Note that each column adds up to 1. The top pane is for iOS/Android-based queries, and bottom pane for browser-based queries. This graph is drawn based on the proposal of Thomas et al. [211], but derived using a different data (`Seek.com`).

Model	iOS/Android		browser	
	parameter	WMSE	parameter	WMSE
SDCG	$k = 51$	1.03×10^{-2}	$k = 51$	0.80×10^{-2}
RBP	$\phi = 0.88$	0.23×10^{-2}	$\phi = 0.85$	0.39×10^{-2}
INSQ	$T = 4.36$	0.13×10^{-2}	$T = 3.29$	0.18×10^{-2}

Model	iOS/Android		browser	
	parameter	WMSE	parameter	WMSE
SDCG	$k = 51$	0.94×10^{-2}	$k = 51$	0.89×10^{-2}
RBP	$\phi = 0.88$	0.14×10^{-2}	$\phi = 0.85$	0.44×10^{-2}
INSQ	$T = 4.47$	0.07×10^{-2}	$T = 3.18$	0.21×10^{-2}

Table 3.6: Best-fit parameters for three static user models computed by minimising $\mathsf{WMSE}(\hat{\mathbf{C}}, \mathbf{C})$ across top-50 rank positions. This computation employs Rule "G" with micro- (top) and macro-averaging (bottom).

jump $(p_{k+1} - p_k)$	$\langle p_k, p_{k+1} \rangle$'s	freq.
-2	$\langle 4, 2 \rangle$	1
-1	$\langle 5, 4 \rangle, \langle 7, 6 \rangle, \langle 6, 5 \rangle$	3
$+1$	$\langle 1, 2 \rangle, \langle 2, 3 \rangle$	2
$+2$	$\langle 3, 5 \rangle, \langle 2, 4 \rangle$	2
$+3$	$\langle 4, 7 \rangle$	1

Table 3.7: Frequency distribution of impression jumps for the impression sequence $P_2 = \langle 1, 2, 3, 5, 4, 2, 4, 7, 6, 5 \rangle$.

most common following "+1" jumps indicates that users also tend to perform "one step backwards, and two step forwards". This pattern of behaviour was first noted by Thomas et al. [211] in the context of Web search, and appears to also occur in job search.

Investigating adjacent pairs of impression jumps give insights about three consecutive impressions. For example, the impression sequence

$$\langle 1, 3, 2, 3, 4, 5, 3, 5, 6, 7, 5, 2, 1 \rangle$$

would reduce to the "1-jump" sequence

$$\langle +2, -1, +1, +1, +1, -2, +2, +1, +1, -2, -3, -1 \rangle \,,$$

and the latter can be further processed to generate the list of adjacent pairs of jumps

$$\langle (+2, -1), (-1, +1), (+1, +1), (+1, +1), \ldots, (-2, -3), (-3, -1) \rangle \, .$$

Each cell in the Figure 3.10 (page 92) represents the probability of the second jump in each pair conditioned on the first one. It can be seen that "+1" is the most common next jump regardless of what magnitude and which direction (+ or −) occurred at the first jump. This effect is stronger when the direction of the first jump is +. A "+2" jump that is followed by the "–1" jump is also prevalent regardless of the modality of the query (mobile or browser). A jump in one direction that is followed by one in the other direction is quite common, an effect suggested by the fact that the level of shading around the diagonal is noticeable. Thomas et al. [211] also demonstrate the same result. Note that this effect is stronger for mobile-based queries. The other noticeable difference between mobile- and browser-based queries is that a positive jump that is followed by the "–1" jump is less common in mobile-based queries, suggesting that scrolling-down actions are more common than scrolling-up actions for mobile-based queries. Jones et al. [112] also suggest the same finding for search activities using a small screen device.

This study examined the extent to which the clickthrough actions are positionally ordered in the collection of action sequences. The queries are first stratified based on the number of distinct clickthroughs in the sequences, and then compute Kendall's τ for each clickthrough subsequence against its sequence of step numbers. The details of the latter process are described as follows. First, the intermediate sequence $\langle (t, r_t) \mid (\text{“C”}, r_t) \in \mathcal{A} \rangle$ is formed. Second, Kendall's τ coefficient is then used to measure the relationship between t and r_t, with repeated clicks being removed. To give an illustration, consider again the example action sequence $\mathcal{A}_1$ described in Section 3.2. In this example, there are three click actions at ranks 3, 7, and 8, which are also associated with step numbers, respectively, 6, 11, and 16. Thus, the corresponding ready-to-compute intermediate sequence is $\langle (6, 3), (11, 7), (16, 8) \rangle$, with $\tau = 1.0$.

The result of this Kendall's τ analysis is shown in Table 3.8, with users mostly performing clickthrough actions in increasing order. As can be seen, mobile-based users have a stronger tendency to do this than browser-based users. Figure 3.11, which shows the distribution of clickthrough jumps, portrays the same pattern but from a different perspective. Positive clickthrough jumps, which correspond to concords when calculating Kendall's τ, obviously dominate the distribution, suggesting that users tend to click the results from top to bottom.

Clicks	iOS/Android		browser	
	mean τ	$\tau > 0$	mean τ	$\tau > 0$
2	0.77	88.4%	0.70	84.8%
3	0.78	86.2%	0.72	83.7%
4	0.83	93.4%	0.75	90.7%
5	0.85	94.4%	0.79	94.7%

Table 3.8: Mean value of Kendall's τ for clickthrough sequences as a function of the number of clickthroughs. All paired differences between iOS/Android- and browser-based users are significant, with $p < 0.05$ using a t-test for all cases.

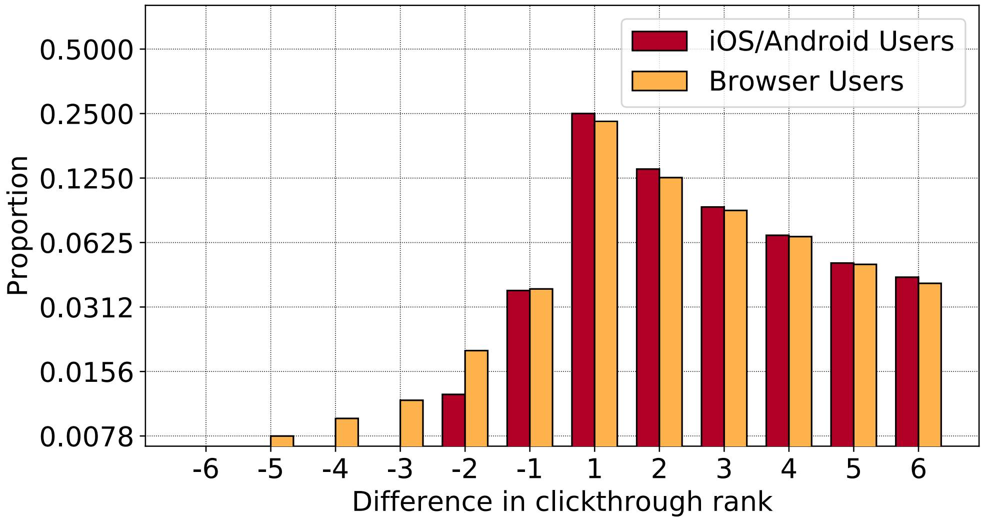


Figure 3.11: Distribution of clickthrough jumps, with y-axis rendered in a logarithmic scale. This graph only includes jumps $\langle -6, -5, \ldots, -1, +1, \ldots, +5, +6 \rangle$.

It is also valuable to investigate how click actions are distributed across rank positions. Figure 3.12 (page 96) shows graphs describing positional distributions of clickthroughs stratified by the total number of clicks in the SERPs. The cell at index location (i, n) represents an estimated conditional probability that users clicked on the result listed at rank position i (vertical axis), given a prior condition that they clicked on a total of n distinct items (horizontal axis). In general clickthrough actions are top-heavy, and results listed on a higher ranking position are more likely to be clicked than those on a lower position, regardless of the final number of clickthroughs.

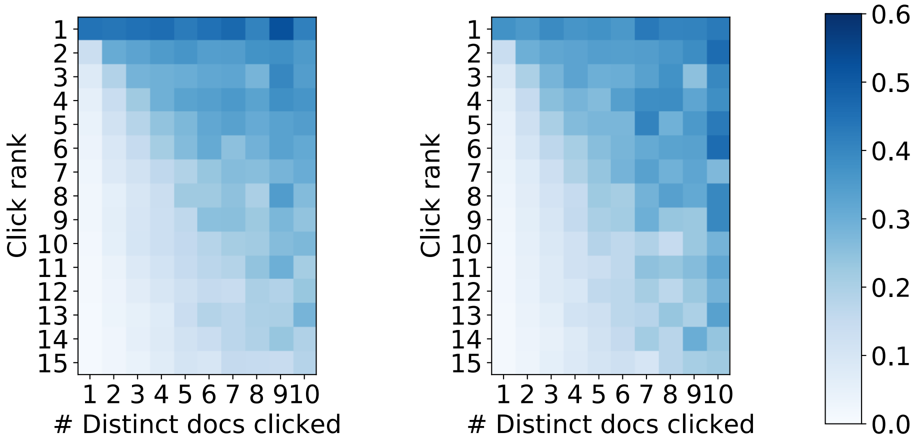


Figure 3.12: Positional distribution of clickthroughs as a function of the number of clickthroughs in the action sequence for both mobile- (left) and browser-based (right) queries.

3.4.2 A Prelude to Clickthroughs

It is also interesting to understand user viewing actions before they decide to click at a particular rank. This is especially useful for gaining insights when building a model for inferring impression distributions from clickthrough sequences. Figure 3.13 (page 97) shows the mean number of distinct impressions prior to and including rank position r_t, and beyond the rank position r_t, before clicks at rank r_t. It can be seen that users tended to examine all results before and including rank r_t prior to a click action at rank r_t, and they also inspected a number of result beyond r_t before the click happened. This outcome again reinforces past observation suggesting that users progress via two steps forward and one step back [211]. Further, the mean number of inspected items beyond r_t for browser-based queries is higher than that for mobile-based queries. This might be due to the fact that a desktop screen is taller than mobile screen (see definition of impression in Section 3.2.1).

The assumption that users always inspect all items before rank r_t prior to a click action at rank r_t has been employed for approximating user viewing behaviour [135, 244]. Lipani et al. [135] employ this assumption to estimate the fraction of user attentions across ranking positions, and then use this statistic to measure how close a predicted session-based user model to the observed behaviour. Zhang et al. [244] use the same assumption to infer impression distributions from clickthrough sequences, but also compute the mean length of gap between any two consecutive clicks to predict the distribution beyond the deepest clickthrough position.

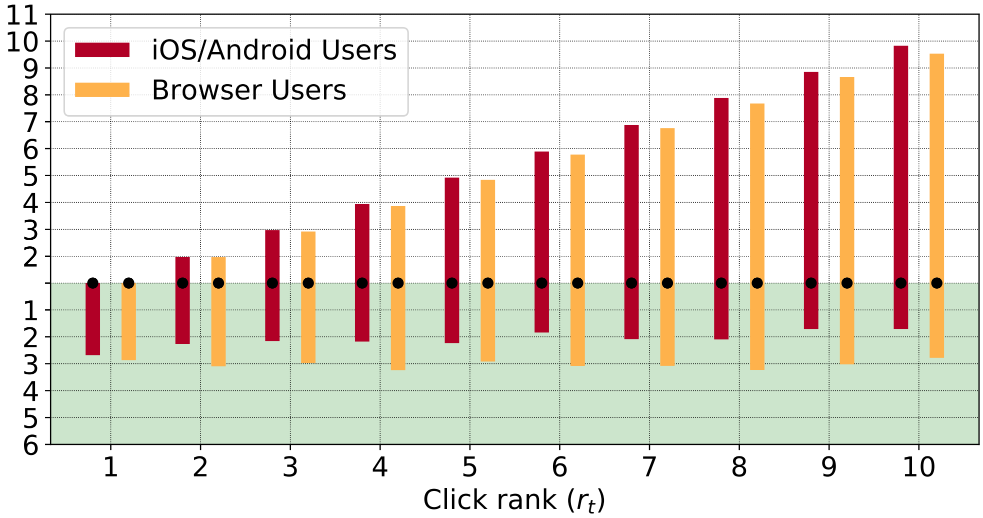


Figure 3.13: Mean number of distinct results inspected (y-axis) prior to and including rank r_t (unshaded area), and beyond rank r_t (green shaded area), with r_t being the rank position of a subsequent click. For example, this graph shows that mobile-based users inspected on average 4.9 items below and including rank 5, and 2.2 items at ranks deeper than 5, before they clicked at rank 5.

3.4.3 Last and Deepest Clickthroughs

It is also useful to be able to predict *stopping ranks* from clickthrough sequences since this information provides a critical resource for meta-evaluation of a metric, specifically for calculating the closeness between predicted behaviour underlying a metric and observed behaviour. Ideally stopping ranks can be accurately observed from impression information. Unfortunately, impressions are not always observable in real search interaction logs, but clickthrough information often is. When impression sequences are not observable, stopping ranks have been approximated using at least two signals: the last clicked ranks and the deepest clicked ranks. The former had been employed by Smucker and Clarke [195] to estimate the total time spent by user to scan down the SERP, by calculating the time interval between the query and the last click observed. It had also been used to quantify the extent to which a particular user stopping model matches observed stopping behaviour, either by visually comparing both predicted and observed stopping ranks distributions [39]; or by calculating the likelihood [20]. Recently Lipani et al. [135] use the deepest click to mark the last inspection point in the SERP, but also with the assumption that users inspect all results below the deepest clicked point.

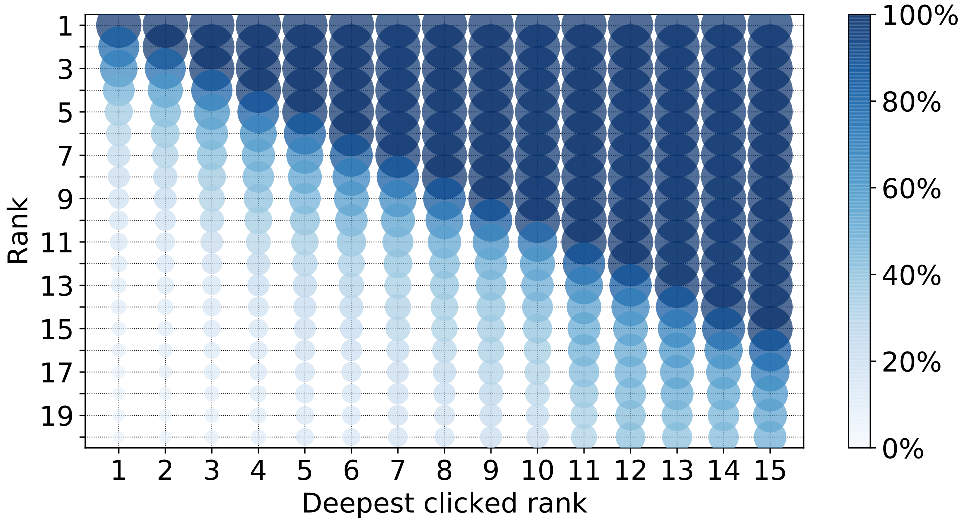
	iOS/Android	browser
The last click in $\mathcal{A}$ at index lc is…		
…the deepest click in $\mathcal{A}$	94.8%	92.4%
…the last action in $\mathcal{A}$	16.2%	21.3%
…the deepest action in $\mathcal{A}$	13.2%	15.3%
The deepest click in $\mathcal{A}$ is…		
…the deepest action in $\mathcal{A}$	14.6%	16.7%

Table 3.9: Statistics regarding the deepest and last clickthroughs, with $lc = \max\{t \mid a_t = \text{“C”}\}$ being the last click index in the sequence $\mathcal{A}$. All paired differences are significant ($p < 0.05$, two sided z-test).

Here we explore statistics in regard to the last click and the deepest click in the action sequences, to verify whether observational data supports existing stopping rank approximation methods, such as those employed by Azzopardi et al. [20] and Lipani et al. [135]. Table 3.9 shows these statistics. The observation that the last click is usually the deepest click in the sequence reinforces the supposition that users perform click actions from top to bottom, as noted previously. However, the two presumptions, that the last clicked document corresponds to the last impression and that the deepest clicked document is associated with the deepest click rank, are at odds with our observational results. Users examined results beyond the deepest clicked rank and after the last click action. Figure 3.14 shows the percentage of action sequences in which each rank position is viewed, stratified by the rank position of the deepest click action. This suggests that users tend to inspect all results below the deepest click rank and a number of documents beyond it.

Our observation further suggests that around 40% to 50% of last-click actions (50% for mobile-based users, and 40% for browser-based ones) were followed by impressions to new results being inspected for the first time. Another perspective in regard to this issue can also be seen in Figure 3.15 (page 100). It is obvious that users still re-examined results that have been previously viewed after the last click action, yet the deepest impression most likely occurred for the first time after the last click.

By adhering to the assumption that users inspect the SERP sequentially from top to bottom, the deepest impression rank can be thought of as an estimate for the actual stopping rank. A critical question that then arises is to what extent the deepest click rank differ from the deepest impression rank. Let di be the deepest impression rank, and dc be the deepest click rank. The difference between the deepest impression rank and the deepest click rank is defined as $\mathit{diff} = di - dc$. Note that diff is always greater than or equal to zero (that is, $\mathit{diff} \in \{0, 1, 2, 3, \dots\}$) since a click action is always preceded by an



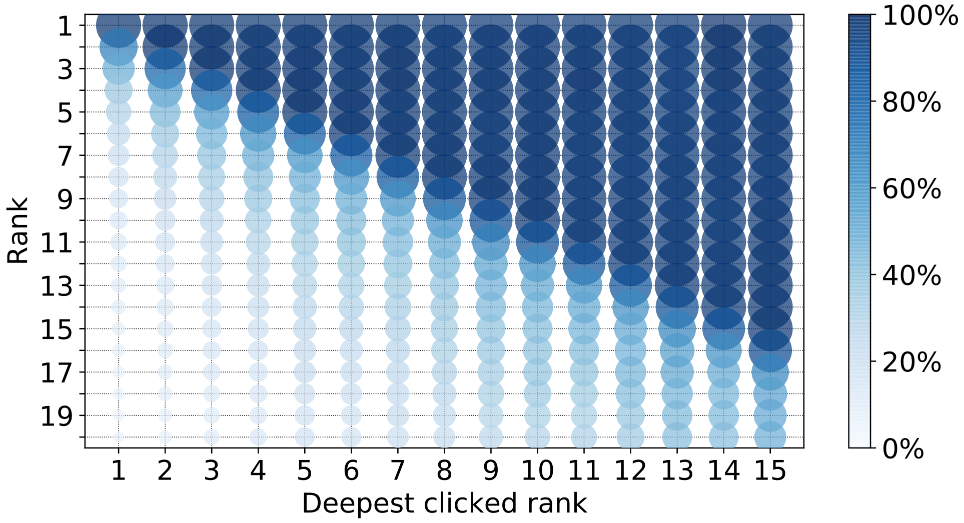
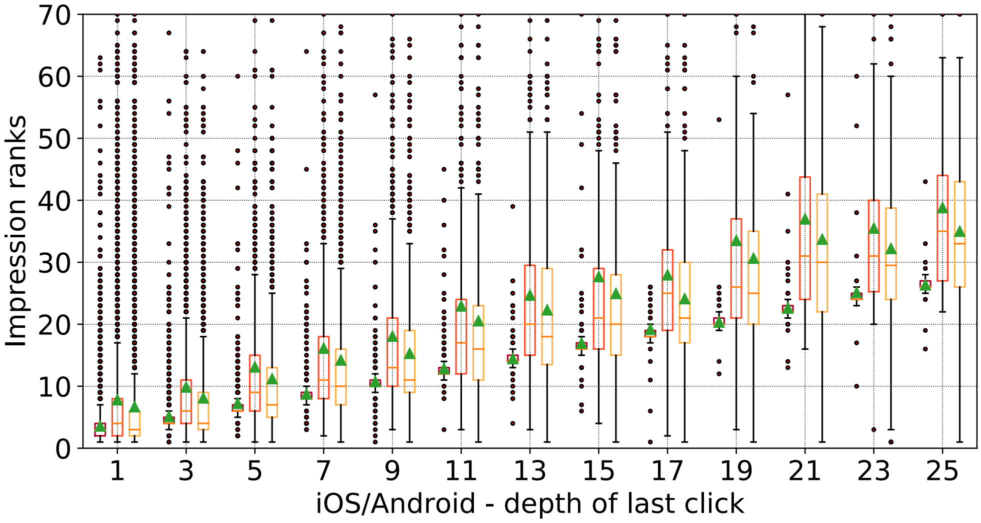


Figure 3.14: Percentage of action sequences in which an impression at a particular rank position is observed (y-axis), stratified by the rank position of the deepest click action (x-axis). The upper graph is for iOS/Android-based queries, and lower for browser-based ones.



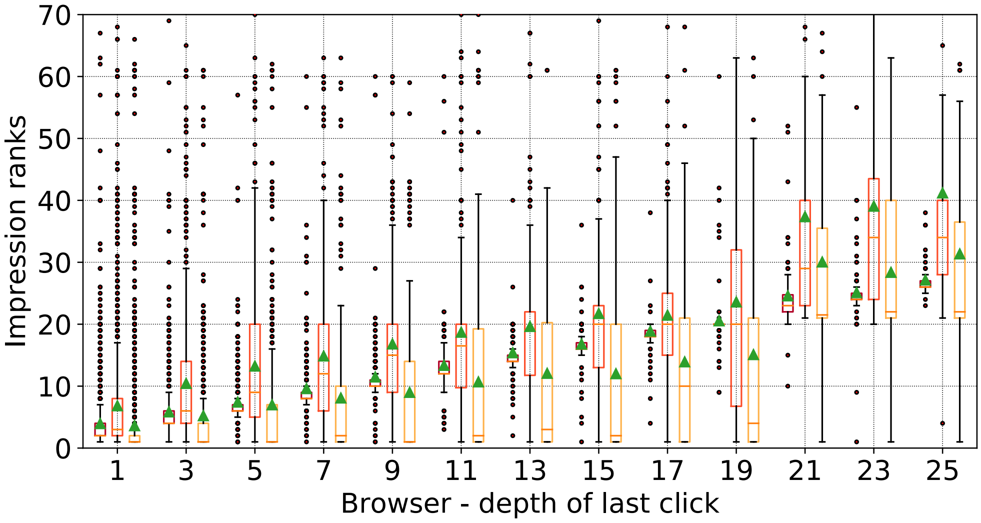


Figure 3.15: Distribution of the deepest impression rank prior to the last click (left box-whisker element in each group of three); distribution of the deepest impression rank after the last click action (middle box-whisker element in each group of three); and distribution of the last impression after the last click (right box-whisker element in each group of three), with distributions being stratified by the rank position of the last click. the green triangle and black dots represent, respectively, the mean value of each distribution and the outliers. Odd depths (only) are shown, with mobile-based queries above, and browser-based ones below; and only action sequences in which impressions occur both before and after the last click action are included.

Deepest click	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
iOS/Android	5.9	5.6	6.4	7.0	7.4	7.5	8.1	7.4	8.1	8.0	10.3	9.9	10.0	10.0	10.9
browser	4.5	5.5	6.4	6.5	7.2	6.8	7.4	7.9	7.3	7.3	7.5	8.7	7.7	7.4	8.1

Table 3.10: Mean number of distinct results inspected beyond the deepest clickthrough rank position, stratified by the rank position of the deepest clickthrough.

impression at the same rank position. Figure 3.16 (page 102) shows the distribution of *diff* for $diff = 0, 1, \ldots, 15$. If no clickthroughs are observed, the deepest click rank is set to zero. In this condition, we find that users examined one or more results beyond the deepest click rank position around 93% to 94% of the time for both mobile- and browser-based queries; and that the expected *diff* for both mobile- and browser-based queries are, respectively, 8.8 and 7.9. This implies that mobile-based users (with infinite scrolling) tended to inspect more results beyond the deepest click rank, compared to browser-based users (with pagination). This is also consistent with what had been observed by Jones et al. [112], where scrolling-down activities are more common than scrolling-up activities for small screen users.

The distribution of *diff* described in Figure 3.16 (page 102) suggests that the deepest click rank cannot be used as a surrogate for the deepest impression rank ($\hat{di} = dc$, with $\hat{di}$ being the estimated deepest impression rank). An *adjustment* process should be applied to the deepest click rank in order to obtain an approximation [244]. Suppose **diff** is a random variable that follows the probability distribution described in Figure 3.16; and **di** is also a random variable that is associated with *di*. Then, a simple approximate solution is to compute the expected value of **di**:

$$\hat{di} = \mathbb{E}[\mathbf{di}] = dc + \mathbb{E}[\mathbf{diff}]\,, \tag{3.4}$$

where $\mathbb{E}[\mathbf{diff}] = 8.8$ ($\mathbb{E}[\mathbf{diff}] = 7.9$) for mobile-based (browser-based) queries. However, Equation 3.4 might be too naïve for at least one reason: *diff* is independent of *dc*. We argue that it is possible to develop a more accurate approach by exploiting the relationship between *dc* and *diff*.

As shown in Table 3.10, a further observation is that the number of distinct items inspected beyond the deepest click rank tended to increase with *dc* itself. This might indirectly imply that *diff* is positively correlated with *dc*. Section 3.5 discusses this issue in detail, and introduces an accurate approach for inferring impression distributions from clickthrough sequences.

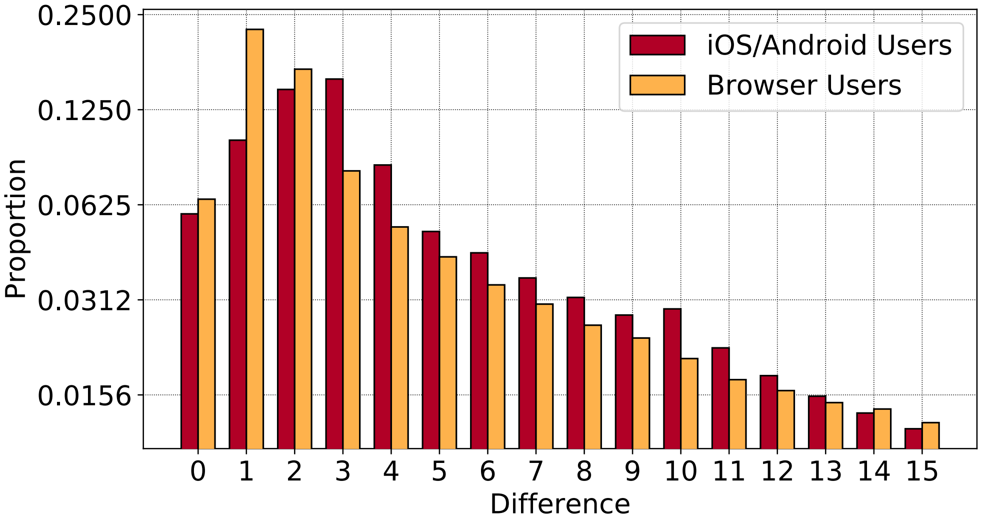


Figure 3.16: Distribution of *diff*, the difference between the deepest impression rank and the deepest clickthrough rank, with the y-axis rendered in a logarithmic scale, and for *diff* ≤ 15. Action sequences that have no impressions are not included.

3.5 Predicting Impression Distributions

We argue that clickthroughs are not directly representative of impressions, particularly for computing empirical $\hat{C}(i)$. However, further analysis suggests that impressions can be predicted from clickthroughs. This section addresses **RQ 3.2** (page 76), introducing an impression model framework, that is, a mechanism for the prediction of impression distributions from clickthroughs. Interaction patterns described in Section 3.4 are incorporated into the development of this model.

3.5.1 Can Clickthroughs Directly Substitute for Impressions?

Section 3.3 introduced our methodology for computing an empirical value $\hat{C}(i)$ using impression sequences and described the use of $\hat{C}(i)$ for calibrating C/W/L-based user models. Here, the same process is applied using clickthroughs and the micro-averaging aggregation method, for the top 50 results. The goal of this section is to see whether clickthrough sequences can be used as a direct surrogate for impression sequences.

Figure 3.17 (page 104) shows the resulting empirical $\hat{C}(i)$ computed using only clickthrough sequences across the top 50 ranking positions, with two reference plots of $C(i)$ for SDCG and INSQ. Consider again the $\hat{C}(i)$ curve estimated using impression sequences

Model	Impression		Click	
	iOS/Android	browser	iOS/Android	browser
RBP	$\phi = 0.88$	$\phi = 0.85$	$\phi = 0.46$	$\phi = 0.46$
INSQ	$T = 4.36$	$T = 3.29$	$T = 1.86 \times 10^{-18}$	$T = 1.14 \times 10^{-18}$

Table 3.11: Best-fit parameters for RBP and INSQ user models computed using impressions and clickthroughs by minimising $\mathit{WMSE}(\hat{\mathbf{C}}, \mathbf{C})$ across top-50 rank positions. Rule "G" was employed for computing $\hat{C}(i)$.

depicted in Figures 3.6 and 3.7 (page 85). Conditional continuation probabilities computed using clickthrough sequences are consistently lower than those estimated using impression sequences for the top 50 rankings.

Table 3.11 further compares the best-fit parameters for the offline metrics RBP and INSQ when they are optimised using impressions and clicks. The resultant best-fit parameter values derived from impressions can also be seen in Table 3.6 (page 93). In general, the clickthrough-based parameter fitting process tends to underestimate the persistence parameter of RBP, ϕ, and the expected volume-of-relevance parameter of INSQ, T.

Recently Liu et al. [139] employed a hierarchical linear regression analysis to find the best-fit SERP-level weights, with query-level satisfaction ratings being the response variable. Liu et al. [139] further show that the best-fit weights are very close to those computed using RBP with $\phi = 0.80$. In other words, RBP with $\phi \approx 0.80$ is expected to produce a query score that correlates reasonably well with query-level satisfaction ratings. Further analysis demonstrates that, when impression sequences are employed, the best-fit values of ϕ are, respectively, 0.85 and 0.88, for browser- and mobile-based queries (see Table 3.6). However, when clickthrough sequences are employed, the optimal persistence parameter for RBP is $\phi = 0.46$ for both modalities, which is notably different to 0.80. Note that the expected viewing depth for the RBP user model with $\phi = 0.80$ is $1/(1-0.80) = 5.0$ items, while with $\phi = 0.46$ it is 1.85 items. This underestimation problem suggests that clickthroughs are not a direct surrogate for impressions. Thomas et al. [211] make the same argument, suggesting that impression sequences from eye-tracking experiments reveal more complex viewing behaviours than clickthrough sequences might suggest, and that the absence of a click at rank i does not imply that the user did not register an impression at rank i. Good abandonment is an example of this phenomenon when users are satisfied by reading the list of snippets in the SERP, without clicking at any of them [133].

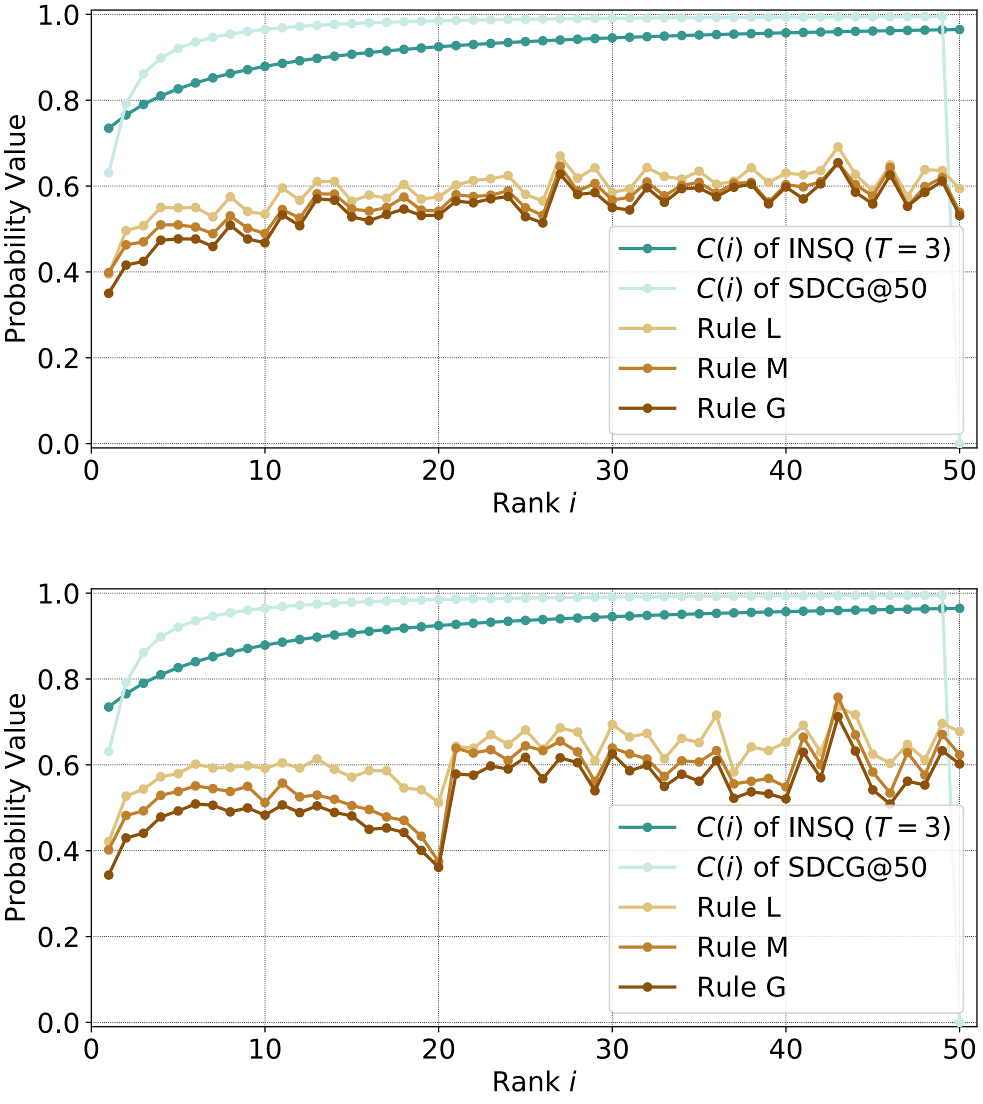


Figure 3.17: Observed $\hat{C}(i)$ for mobile- (top) and browser-based (bottom) queries across the top-50 rank positions, computed using only clickthrough information, and then micro-averaged from the `Seek.com` impression sequences. Predicted $C(i)$ plots for two static user models, SDCG@50 and INSQ with $T = 3$, are shown for reference.

3.5.2 Impression Model

Although impressions cannot be directly replaced by clicks, we argue that impressions can, to some extent, be inferred from clickthrough information. This, however, requires three assumptions:

1. the user scans down the ranked list of results from the top;
2. reads all results at ranks 1 to n, before they click at rank n; and
3. may also examine a number of results deeper than rank n before or after they click at rank n.

Empirical evidence for these assumptions have been provided in Section 3.4 (see Figure 3.8 on page 90, Figure 3.13 on page 97, and Figure 3.16 on page 102). The first assumption had also been validated by a range of eye-tracking experiments [59, 110].

Suppose dc, the deepest click rank, is the only input for the prediction model. Given dc, Equation 3.4 (page 101) provides a simple solution for computing a *point estimate* of the deepest impression rank. The main problem is then to find the best-fit probability distribution that is followed by the random variable **diff**, denoted by $P(\mathbf{diff} = n)$. The expectation of **diff** is simply determined by

$$\mathbb{E}[\mathbf{diff}] = \sum_{i=1}^{\infty} i \cdot P(\mathbf{diff} = i)\,.$$

For a simple model described in Equation 3.4 (page 101), $P(\mathbf{diff} = n)$ depends only on the modality that initiates the queries (that is, mobile or browser), but not on the other potential factors, such as the dc itself.

Table 3.10 (page 101) already suggested that *diff* may depend on click-based factors, including the deepest click rank. A linear regression analysis is employed to identify the contributions of two click-based characteristics to *diff*, the difference between the deepest impression rank and the deepest click rank. Besides the deepest click rank (dc), *the number of clicks*, denoted by nc, is also considered as the second potential factor influencing *diff*. The linear regression model is described as:

$$\mathit{diff} = f(dc, nc; \mathbf{w}) = w_0 + w_1 \cdot dc + w_2 \cdot nc\,, \tag{3.5}$$

where $\mathbf{w} = \{w_0, w_1, w_2\}$ is a set of coefficients that need to be optimised for the linear combination function. Table 3.12 shows the best-fit set of coefficients $\mathbf{w}$. With other factors

Factor	iOS/Android		browser	
	coef.	p	coef.	p
intercept	$w_0 = 7.67$	0.000	$w_0 = 6.40$	0.000
deepest click rank (dc)	$w_1 = 0.17$	0.000	$w_1 = 0.21$	0.000
number of clicks (nc)	$w_2 = -0.61$	0.000	$w_2 = -0.72$	0.000

Table 3.12: Linear regression analysis for computing the effect sizes of two factors, the deepest click rank (dc) and the number of clicks (nc), in modelling *diff*, the difference between the deepest impression rank and the deepest click rank (see Equation 3.5 on page 105). Each of the best-fit coefficients is associated with a p-value.

being equal, a positive w_1 indicates that *diff* tends to increase with dc, while a negative w_2 suggests that *diff* is negatively correlated with nc. Very small p-values for all coefficients suggest that the direction of those relationships is significant, with strong support from the data. Note that this regression analysis did not consider cases with zero clicks (that is, $dc = 0$) since this condition has a different type of interaction pattern.

Instead of a point estimation approach, we propose a more general approach by directly modelling the distribution of impression. Let $DC(u, q)$ be the deepest click rank observed for user u after submitting query q, with $DC(u, q) = 0$ for the zero-click case; and $V(i \mid u, q)$ be the probability that user u registered an impression at rank i with respect to query q. Note that the probability of the user not registering an impression action is denoted as $1 - V(i \mid u, q)$, hence the summation of $V(i \mid u, q)$ over rank i is not necessarily equal to 1. Based on the three assumptions regarding clicking and viewing actions, a general framework for impression models is as follows:

$$\hat{V}(i \mid u, q) = \left\{ \begin{array}{ll} 1 & i \leq DC(u,q) \\ P(\mathbf{diff} \geq (i - DC(u,q)) \mid u) & \text{otherwise}\,, \end{array} \right\} \tag{3.6}$$

where the cumulative distribution $P(\mathbf{diff} \geq n \mid u)$ represents the probability that the user u inspects all results from rank $DC(u, q)$ to rank $DC(u, q) + n$ under the assumption that the user sequentially reads the ranking from top downward. By definition, $P(\mathbf{diff} \geq 0 \mid u) = 1$, meaning that if the deepest click action was observed at rank i for user u, the user must have registered an impression action at rank i. The problem can also be simplified by assuming that the distribution of **diff** does not depend on the user. That is, all users have the same interaction patterns after the deepest click action takes place. This then results in $P(\mathbf{diff} \geq n \mid u) = P(\mathbf{diff} \geq n)$. Thus, the problem is to find a good model approximating $P(\mathbf{diff} \geq n)$.

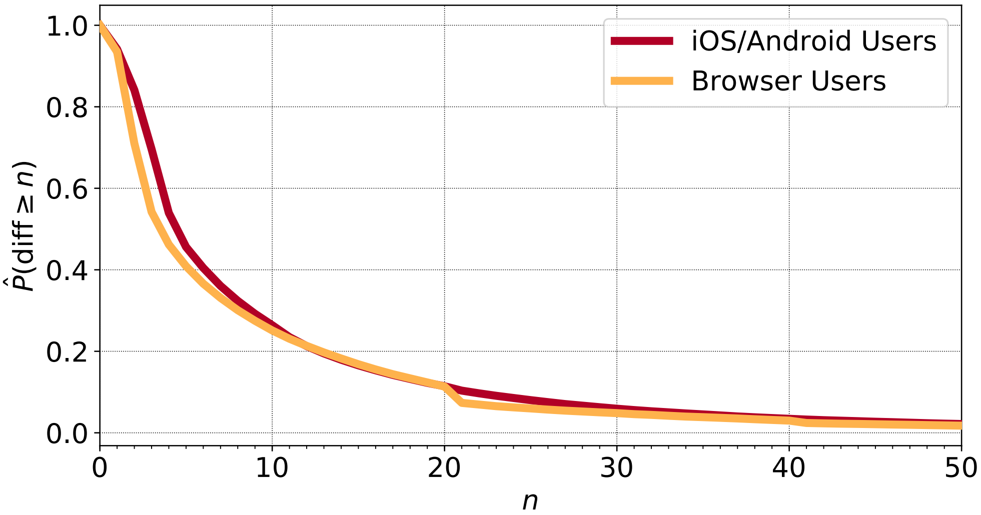


Figure 3.18: Cumulative distribution $\hat{P}(\mathbf{diff} \geq n)$ computed from the `Seek.com` data.

Impression Model 1. We propose a heuristic method by estimating $P(\mathbf{diff} \geq n)$ with a simple mathematical function that provides a similar pattern, with parameters fitted as needed. Figure 3.18 shows the estimated $\hat{P}(\mathbf{diff} \geq n)$ computed from `Seek.com` action sequences across all users and queries (see Table 3.1 on page 80). In this figure, the plot of $\hat{P}(\mathbf{diff} \geq n)$ indicates that an exponential decay function provides a good proxy. This observation suggests that $\hat{P}(\mathbf{diff} \geq n)$ be modelled as follows:

$$\hat{P}(\mathbf{diff} \geq n) = e^{-n/K}\,, \tag{3.7}$$

where $K > 0$ is a tunable variable controlling the decay rate. From a human perspective, K can also be interpreted as the *persistence beyond the deepest click*. The higher the value of K, the more documents that are examined by users beyond the deepest click rank. The curve fitting to the two lines in Figure 3.18 yields the following two constants:

$$K_{mobile} = 7.94\,, \tag{3.8}$$

$$K_{browser} = 7.11\,. \tag{3.9}$$

The fact that K_{mobile} is higher than $K_{browser}$ indicates that mobile-based users tend to be more persistent beyond the deepest click rank, compared to browser-based users. This is also related to Figure 3.16 (page 102) described in Section 3.4.

Impression Model 2. Impression Model 1 assumes that *diff* only depends on the application platform (browser or mobile applications). Both mobile- and browser-based search activities may employ different constant K. However, Table 3.12 (page 106) has provided evidence for the two click-based quantities, the number of clicks nc and the deepest click rank dc, as being two important factors influencing the *diff*. Impression Model 2 incorporates these by considering K not as a constant, but as a linear combination of dc and nc. A fitting process on `Seek.com` data results in two linear models:

$$\begin{aligned} K_{mobile} &= 5.92 + dc \cdot 0.31 - nc \cdot 0.61\,, && (3.10) \\ K_{browser} &= 5.10 + dc \cdot 0.29 - nc \cdot 0.14\,. && (3.11) \end{aligned}$$

Impression Model 2 then approximates $\hat{P}(\textbf{diff} \geq n)$ as follows:

$$\hat{P}(\textbf{diff} \geq n) = e^{-n/g(K)}\,, \tag{3.12}$$

where $g(x)$ is a function that takes $x \in \mathbb{R}$ as an input, and returns zero as x goes to $-\infty$, while approximately acting as an *identity function* for $x > 0$. This criteria is approximately held by the following "softplus" function:

$$g(x) = \ln(1 + e^x)\,.$$

The ZPM Impression Model. In contrast to the approaches described in this section, Zhang et al. [244] contend that the distribution of **diff** should depend on the user u, and thus the parameter tuning process should be on a per-user basis. They extrapolate $P(\textbf{diff} \geq n \mid u)$ using the average length between two consecutive clickthrough rank positions. Formally, they use the *click gap distribution* of a user u, denoted by $P(\textbf{gap} = n \mid u, q)$ with $gap \geq 1$, or the probability that the user u inspects $n - 1$ consecutive results without clicking on any of them after submitting query q. For example, if a user clicks at ranks 1, 4, and 6, the corresponding gaps are 1 $(1 - 0)$, 3 $(4 - 1)$, and 2 $(6 - 4)$. The ZPM Impression Model then defines $P(\textbf{diff} \geq n \mid u)$ as:

$$\hat{P}(\textbf{diff} \geq n \mid u) = \hat{P}(\textbf{gap} \geq n \mid u)\,, \tag{3.13}$$

where $P(\textbf{gap} \geq n \mid u)$ is computed by averaging $P(\textbf{gap} \geq n \mid u, q)$ over all queries submitted by user u; and $P(\textbf{gap} \geq 1 \mid u) = 1$ by definition. Therefore, the ZPM Impression Model requires an additional assumption, that the user always examines rank $DC(u, q) + 1$

after the deepest click action. Zhang et al. [244] also address the problem of smoothing for $P(\mathbf{gap} \geq n \mid u)$ since clickthrough data initiated from a single user is usually sparse:

$$\hat{P}_{smooth}(\mathbf{gap} \geq n \mid u) = \alpha_u \cdot \hat{P}(\mathbf{gap} \geq n \mid u) + (1 - \alpha_u) \cdot \hat{P}(\mathbf{gap} \geq n)\,, \tag{3.14}$$

where $P(\mathbf{gap} \geq n)$ is the global click gap distribution estimated from the whole set of users; and α_u is the smoothing parameter associated with user u, and is calculated as follows:

$$\alpha_u = \frac{CT(u)}{CT(u) + \mu}\,, \tag{3.15}$$

where $CT(u)$ is the total number of clicks recorded for user u, and μ is an empirical value. Zhang et al. [244] further employ $\mu = 5$.

The AWTC Model. Recently, Azzopardi et al. [22] also propose a methodology to compute $\hat{C}(i)$ from clickthrough data, using the following formula:

$$\hat{C}(i) = \frac{\sum_{j=i+1}^{\infty} n(u,j)}{\sum_{j=i}^{\infty} n(u,j)}\,,$$

where $n(u,j)$ is the number of users who clicked at rank j and did not return to the SERP. In this study, $n(u,j)$ is replaced with $q(j)$ (that is, the number queries for which the item listed at rank j was the last one clicked by the user), since a user is associated with many queries in the data.

This approach implies that users inspect the results in turn from rank 1 to rank j, where j is the rank position of the last click in the action sequence. That is, the AWTC method actually embodies an implicit impression model:

$$\hat{V}(i \mid u,q) = \left\{ \begin{array}{ll} 1 & i \leq DC(u,q) \\ 0 & \text{otherwise}\,. \end{array} \right\}$$

Azzopardi et al. [22] originally used the last click rank, instead of the deepest click rank, for marking the stopping rank. However, Table 3.9 (page 98) already suggested that the last click ranks tend to be the deepest click ranks. Lipani et al. [135] employ the same impression model in their work.

3.6 Impression Model Evaluation

This section describes an application of impression models for computing empirical $\hat{C}(i)$, which is critical to the development of user model-based effectiveness metrics. This application also serves as a tool for measuring the accuracy of impression models by quantifying a similarity score between $\hat{C}(i)$ derived from impression models and the gold-standard $\hat{C}(i)$ computed from actual impression sequences. In addition, a method is also proposed for validation of impression models via the distribution of $\hat{V}(i)$ itself, which is computed as the aggregation of individual $\hat{V}(i \mid u, q)$ across all users and queries.

3.6.1 Inferring $C(i)$ from Impression Models

Section 3.3 already described the method for estimating conditional continuation probability from a collection of action sequences. This approach, however, requires impression information in order to yield an accurate estimate. In the absence of impression sequences (that is, only clicks are available), impression models can alternatively be used to infer $\hat{C}(i)$. Recall that directly using clickthrough sequences for computation of $\hat{C}(i)$ is not useful, and leads to an underestimation (see Figure 3.17 on page 104).

From the perspective of the C/W/L framework, the weighting function $W(i)$ is a normalised form of the impression probability $V(i)$, so that $W(i)$ sums to one:

$$W(i) = \frac{V(i)}{\sum_{i=1}^{\infty} V(i)} = W(1) \cdot V(i) \,. \tag{3.16}$$

Therefore, $C(i)$ can theoretically be computed from $V(i)$ using:

$$C(i) = \frac{W(i+1)}{W(i)} = \frac{V(i+1)}{V(i)} \,. \tag{3.17}$$

This relationship serves as a basis for computing an empirical estimate of $C(i)$ from impression models by considering $\hat{V}(i \mid u, q)$ as an *expected count*. Let $N(i, u, q)$ and $D(i, u, q)$ respectively be the per-user-query numerator and denominator contributing to $C(i)$ for user u and query q,

$$\begin{aligned} N(i,u,q) &= \hat{V}(i+1 \mid u,q)\,, \text{ and} && (3.18) \\ D(i,u,q) &= \hat{V}(i \mid u,q)\,. && (3.19) \end{aligned}$$

One way to compute $\hat{C}(i)$ via an impression model is by micro-averaging across all users and queries:

$$\hat{C}(i) = \frac{\sum_{u \in U} \sum_{q \in Q(u)} N(i, u, q)}{\sum_{u \in U} \sum_{q \in Q(u)} D(i, u, q)}\,, \tag{3.20}$$

where U is a set of users and $Q(u)$ is a set of queries observed from user u in the dataset. For example, consider three queries q_1, q_2, q_3 recorded from a mobile-based user u, in which the deepest click rank positions are 1, 7, and 8, respectively. If Impression Model 1 is employed, the values of $\hat{V}(i \mid u, q)$ for $i = 1, 2, \ldots, 10$ are:

$$\begin{aligned}
\hat{V}(i \mid u, q_1) &= \langle 1.00, 0.87, 0.75, 0.65, 0.56, 0.48, 0.42, 0.36, 0.31, 0.27 \rangle\,, \\
\hat{V}(i \mid u, q_2) &= \langle 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 0.87, 0.75, 0.65 \rangle\,, \\
\hat{V}(i \mid u, q_3) &= \langle 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 0.87, 0.75 \rangle\,.
\end{aligned}$$

The estimated values $\hat{C}(i)$ for $i \in \{2, 7, 9\}$ are then computed as follows:

$$\begin{aligned}
\hat{C}(2) &= \frac{0.75 + 1.00 + 1.00}{0.87 + 1.00 + 1.00} = 0.96\,, \\
\hat{C}(7) &= \frac{0.36 + 0.87 + 1.00}{0.42 + 1.00 + 1.00} = 0.92\,, \\
\hat{C}(9) &= \frac{0.27 + 0.65 + 0.75}{0.31 + 0.75 + 0.87} = 0.87\,.
\end{aligned}$$

3.6.2 Model Validation

A held-out dataset containing 100,103 action sequences was drawn from iOS/Android-based `Seek.com` interaction logs during the period between 15 October 2017 and 08 April 2018. As a result, this held-out data is mutually exclusive from the collection of action sequences described in Table 3.1 (page 80). Note that three impression models introduced previously in Section 3.5 assume that the SERPs are not paginated, and hence have infinite scrolling. Therefore, only mobile-based queries are considered because they do not have page boundary effects (see Figure 3.7).

The held-out set of mobile-based action sequences serves as a test set for evaluating the quality of predicted impression distributions in two different ways:

1. by measuring the closeness between the "true" $\hat{C}(i)$ estimated from actual impression sequences and the $\hat{C}(i)$ values computed via impression models; and

2. by computing the similarity between the "true" impression distribution vector $\hat{V}(i)$

resulting from the use of impression sequences and that directly estimated from impression models.

The held-out data is randomly split into ten folds. The similarity score (such as, mean squared error) between predicted and observed behaviours is then computed for each fold. Finally, the average of similarity scores across ten folds is reported. This mechanism also allows the computation of several statistics, such as the Wilcoxon signed-rank test and paired t-test, to see the significance of differences between any two impression models.

Continuation Probability. An empirical estimate of the conditional continuation probability is computed from the held-out set of clickthrough sequences using impression models, with parameters tuned using original action sequences described in Table 3.1. The resulting values are then compared against the corresponding gold-standard $\hat{C}(i)$ derived from the impression sequences in the held-out dataset, using the "weighted-by-frequency" mean squared error (WMSE) as a distance function (see Equation 3.3 on page 87). Figure 3.19 visually shows the plots of $\hat{C}(i)$ estimated using four impression models (Model 1, Model 2, ZPM, and AWTC) for $1 \leq i \leq 50$, with the one observed from actual impression sequences being a reference. The visual observation shows that the $\hat{C}(i)$ values computed using Impression Model 2 are very close to the "true" $\hat{C}(i)$ values from impression sequences. Table 3.13 confirms this relationship by showing detail reports using average WMSE across ten partitions of the held-out data for top-20 and top-50 results. Top-20 is also included because a typical `Seek.com` result page contains 20 items (see Section 3.2.2 on page 79). The $\hat{C}(i)$ computed from Impression Model 2 provides the lowest error among the others, with low p-value indicating its significant superiority. As already noted, computing $\hat{C}(i)$ values directly using click sequences (without the use of the impression model) is not useful.

Empirical Impression Distributions. Suppose $V(i)$ is the probability of the user inspecting rank i, and is directly proportional to the probability weight function associated with rank i, $W(i)$. An important property of $V(i)$ is $V(i) \geq V(i+1)$, which implies that the probability of a user registering an impression at lower rank positions is never greater than the probability of inspecting results at higher rank positions. Note also that $V(i)$ does not necessarily sum to one. The operational computation of $\hat{V}(i)$ is as follows:

$$\hat{V}(i) = \frac{\sum_{u \in U} \sum_{q \in Q(u)} \mathbf{v}(i, u, q)}{\sum_{u \in U} |Q(u)|}\,, \tag{3.21}$$

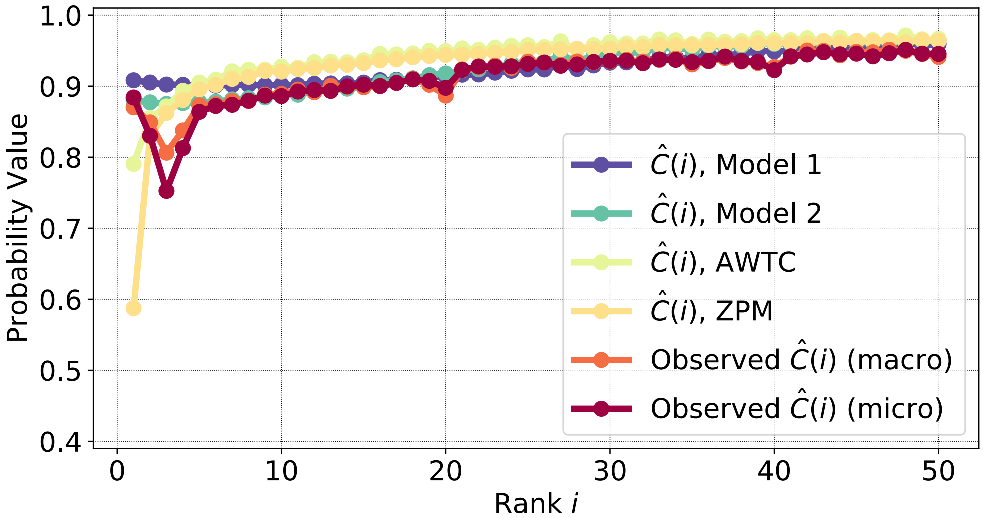


Figure 3.19: Estimated $\hat{C}(i)$ across top 50 items for each query, computed using four impression models (Model 1, Model 2, ZPM, and AWTC) derived from clickthrough sequences of the held-out queries. The gold standard $\hat{C}(i)$ computed using the true impression sequences is included as a reference point.

where $\mathbf{v}(i, u, q)$ is an impression binary indicator when using impression sequences; or a click binary indicator when using click sequences; or an expected count $\hat{V}(i \mid u, q)$ when using impression models. Mean squared error (MSE) is employed to measure the closeness between $\hat{V}(i)$ derived from impression models using held-out click sequences and gold-standard weights computed from the held-out impression sequences. Weighted mean squared errors were not used because the computation of $\hat{V}(i)$ described in Equation 3.21 employs the same denominator (that is, the same weight) across all rank positions. Table 3.14 (page 114) shows the results of this measurement for $n = 20$ and $n = 50$. Under this evaluation process, Impression Model 2 once again outperformed the other models, followed by the ZPM model. Figure 3.20 (page 115) visually shows the estimated $\hat{V}(i)$ computed using these impression models. Nevertheless, inferring $\hat{V}(i)$ using impression models results in predicted distributions that are better than those computed from only clickthrough sequences.

Commercial Web Search. A sample of interaction logs containing around 105,000 Web queries is also available for experimentation. These logs were initiated from users in the U.S. who performed search activities using `Bing.com` on desktop-based browsers, and have been anonymised to hide personally identifiable information. In addition to `Bing.com`

Model	WMSE (top-20)		WMSE (top-50)	
	Micro	Macro	Micro	Macro
Clicks	172.5×10^{-3}	179.1×10^{-3}	169.3×10^{-3}	175.4×10^{-3}
ZPM	5.7×10^{-3}	4.1×10^{-3}	4.5×10^{-3}	3.3×10^{-3}
AWTC	4.1×10^{-3}	2.5×10^{-3}	3.4×10^{-3}	2.1×10^{-3}
Model 1	4.0×10^{-3}	2.5×10^{-3}	3.1×10^{-3}	2.0×10^{-3}
Model 2	2.2×10^{-3}	1.2×10^{-3}	1.8×10^{-3}	1.0×10^{-3}

Table 3.13: Average frequency-based weighted mean squared error (WMSE) between estimated $\hat{C}(i)$ (micro- and macro-averaged method) and the $\hat{C}(i)$ computed using four impression models (Model 1, Model 2, ZPM, and AWTC) running on ten partitions of the held-out data. The $\hat{C}(i)$ directly inferred using clickthrough sequences is also shown as a reference. Lower values are better. Model 2 significantly outperformed the other approaches (Wilcoxon signed-rank test, $p < 0.01$; and paired t-test, $p < 0.01$).

Model	MSE (top-20)	MSE (top-50)
Click distribution	11.53×10^{-2}	4.93×10^{-2}
ZPM	1.41×10^{-2}	0.28×10^{-2}
AWTC	4.88×10^{-2}	2.05×10^{-2}
Model 1	1.19×10^{-2}	0.50×10^{-2}
Model 2	0.37×10^{-2}	0.20×10^{-2}

Table 3.14: Average mean squared error (MSE) between the $\hat{V}(i)$ estimated using impression models and the $\hat{V}(i)$ estimated using actual impression sequences from held-out data. Lower values are better. Model 2 significantly outperformed other approaches (Wilcoxon signed-rank test, $p < 0.01$; and paired t-test, $p < 0.01$).

data, this study also employs two Web-based pre-existing datasets constructed from lab-based user studies: J&A dataset [107] and THUIR3 dataset [139]. The J&A dataset consists of 388 Web search queries collected by Jiang et al. [107] from an eye-tracking experiment. In contrast to the other two Web search datasets, this data contains eye-fixations (that is, impressions) collected using a Tobbi 1750 eye-tracker for a minimum duration of 100 milliseconds [107]. Liu et al. [139] employ the third dataset, THUIR3, to investigate factors affecting query- and session-level user satisfaction. Table 3.15 summarises Web-based search interaction logs used in this study.

In the `Bing.com` data, click sequences are observable, but impression sequences are not. Hence, impression models were applied to infer continuation probabilities from this data. Figure 3.21 shows the empirical $\hat{C}(i)$ computed using Impression Model 2 for top-10 results. However, the parameters are not trained on `Seek.com` data (see Equation 3.10 on page 108) because Web search users may have different behaviour from job search

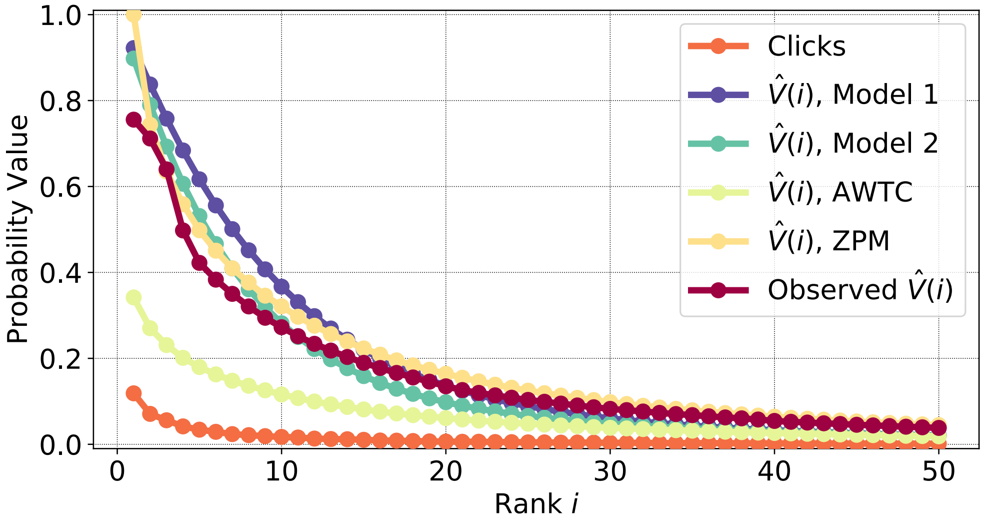


Figure 3.20: Estimated $\hat{V}(i)$ across top 50 items for each query, computed using four impression models (Model 1, Model 2, ZPM, and AWTC) derived from clickthrough sequences of the held-out queries. The gold standard $\hat{V}(i)$ computed using the true impression sequences is included as a reference.

	`Bing.com`	J&A [107]	THUIR3 [139]
Source	Commercial	Lab. experiment	Lab. experiment
Queries	105,000	388	1,548
SERP size	paginated, 10	truncated, 9	paginated, 10
Impressions	No	Yes	No
Clickthroughs	Yes	Yes	Yes
Query-level Ratings	No	No	Yes

Table 3.15: Web-based search interaction logs.

users, particularly in regard to *diff* distribution. Instead, the parameters were tuned on J&A dataset. Although the size of this data is not large, this is particularly useful for tuning impression models in the context of Web search, since it contains impression and clickthrough sequences.

Figure 3.21 reveals that $\hat{C}(i)$ increases with i for $1 \leq i \leq 7$, but then decreases around the page boundary. Note that typical `Bing.com` result pages contain 10 snippets. Further, Table 3.16 shows the best-fit parameters computed by minimising $\mathsf{WMSE}(\hat{\mathbf{C}}, \mathbf{C})$ on `Bing.com` data. In contrast to the results on `Seek.com` job search data (see Table 3.6), these results shows that RBP is more accurate than INSQ in the context of `Bing.com` Web search. The fact that the parameters of both RBP and INSQ for `Bing.com` data are lower

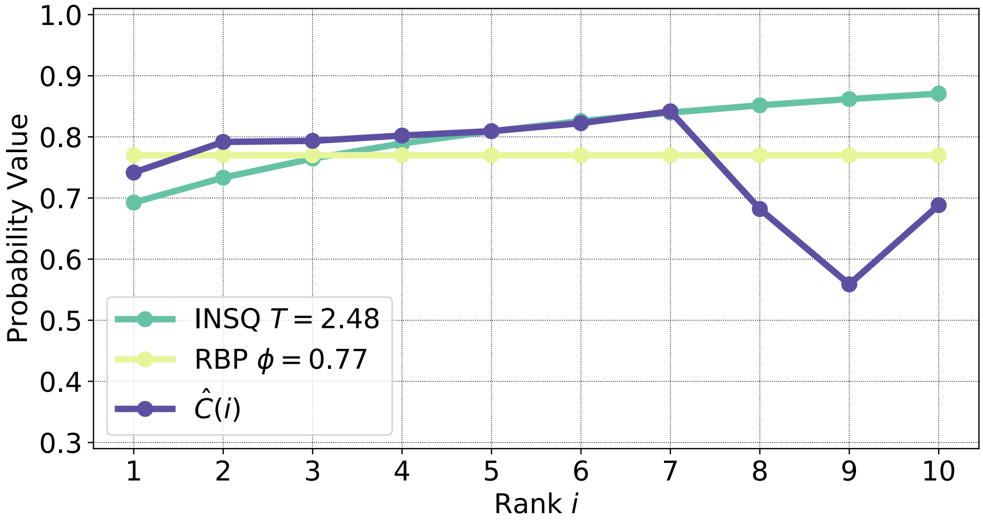


Figure 3.21: Estimated $\hat{C}(i)$ computed from a sample of interaction logs drawn from the Bing.com Web search engine. The $C(i)$ plots of two static user models, RBP and INSQ, are also included with parameters optimised to minimise the weighted mean squared error between $\hat{C}(i)$ and $C(i)$ for top-10 results.

Model	Top-5		Top-10	
	parameter	WMSE	parameter	WMSE
RBP	$\phi = 0.78$	0.35×10^{-2}	$\phi = 0.77$	3.17×10^{-2}
INSQ	$T = 3.08$	0.89×10^{-2}	$T = 2.48$	7.27×10^{-2}

Table 3.16: Best-fit parameters computed from a sample of Bing.com interaction logs.

than those for Seek.com data suggests that Web search users are less persistent than job search users.

It is also desirable to investigate whether the resultant best-fit parameters ($\phi = 0.77$ for RBP and $T = 2.48$ for INSQ) are also supported by evidence from the other Web search datasets. In particular, this study examines whether they also lead to the best correlation coefficients with query-level user satisfaction ratings, knowing that it is critical for a metric score to have a strong relationship with user satisfaction. Further, this study uses another lab-based dataset, namely THUIR3, which contains around 1,500 Web queries with user-generated satisfaction ratings [139]. A grid search is used to find the best-fit parameters for RBP and INSQ, where $\phi = 0, 0.01, 0.02, \ldots, 1.0$ and $T = 0.5, 0.6, \ldots, 5.0$ were tested for, respectively, RBP and INSQ. Figure 3.22 shows that the best correlation coefficients were achieved when $\phi = 0.78$ for RBP and $T = 2.60$ for INSQ. The expected search length (ESL)

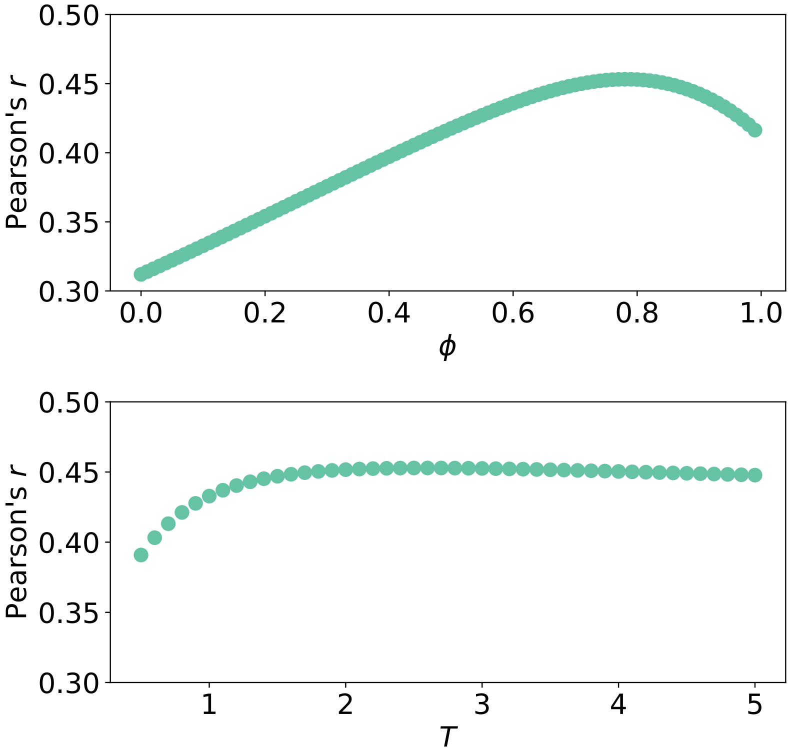


Figure 3.22: Correlation coefficients (Pearson's r) as a function of parameter values of two static user models, RBP (top) and INSQ (bottom), computed using 1,500 Web queries from THUIR3 dataset [139]. The optimal parameters for RBP and INSQ are $\phi = 0.78$ and $T = 2.60$, respectively.

for RBP with $\phi = 0.78$ and INSQ with $T = 2.60$ are, respectively, 4.54 and 5.70. These resultant values are fairly consistent with those computed from `Bing.com` interaction logs for top-10 results: $\phi = 0.77$ for RBP (ESL $= 4.35$) and $T = 2.48$ for INSQ (ESL $= 5.46$).

3.7 Summary

Understanding of observed behaviours is critical to the development of user model-based effectiveness metrics, since it provides guidelines of how the predicted behaviours should be modelled. This chapter described tools for inferring observed behaviour from search interactions logs. These tools include a mechanism for computing continuation probabilities from impression sequences, and an impression model for inferring impression distributions from click-based logged behaviours.

In Section 3.3, we propose three heuristic rules for computing empirical estimates of conditional continuation probabilities from impression sequences. The rule “L” states that all impressions in the sequence are continuation, except the last one. The rule “M” considers all instances of maximum impression rank as being non-continuations. The rule “G” is a combination between rules “L” and “M”, recording a continuation for an impression rank that is still followed by one at a higher ranking position. Our experiment results show that the three proposed rules are consistent in the sense that they all yield the same behavioural pattern of empirical continuation probabilities. That is, the continuation probabilities of users tend to increase as they proceed into deeper ranking positions. Note that Moffat et al. [153, 155] also find the same interaction pattern on logged behaviours based on Web search, and refer to this observed behaviour as “sunk cost recovery”. Existing static user models, such as SDCG and INSQ, embody this observed behaviour, but RBP does not. We further propose a method for computing the accuracy between these three user models and the observed behaviour and found that INSQ is the most accurate among these three static models.

Unfortunately, impression sequences may not always be observable from logged behaviours, while clickthrough sequences usually are. Figure 3.17 already suggested that clickthrough sequences are not a direct surrogate for impression sequences, and thus are not useful for inferring observed continuation probabilities. However, impressions can, to some extent, be predicted from clickthrough sequences via impression models. This chapter has demonstrated that impression models are critical, particularly for computing continuation probabilities from click-based logged behaviours (see Figures 3.19 and 3.20).

In the absence of impression sequences, several studies used a simple impression model which assumes that the deepest click rank is the last rank position inspected by users. However, as described in Section 3.4, users examined items beyond the deepest click rank. In addition, further observations reveal other interaction patterns that are useful for the development of impression models and, in general, search effectiveness models: (1) users tend to scan the SERP from the top; (2) users tend to inspect all results before rank i before deciding to click at rank i; (3) the last click ranks in action sequences tend to be the deepest click rank; (4) mobile-based users (with unlimited scrolling) tend to examine more documents beyond the deepest click rank position, compared to desktop browser-based users (with pagination); and (5) the number of items inspected beyond the deepest click rank tends to increase with the rank position of the deepest click action.

Section 3.5 proposed a framework for impression models. The model generally assumes that users always inspect all items from the top until the deepest click rank position. (The evidence for this assumption has been provided in Section 3.4.) Thus, the problem is to model user viewing behaviour beyond the deepest click rank. One way to operationalise this behaviour is via *diff*, the difference between the deepest impression rank and the deepest click rank. Further analysis suggests that *diff* is affected by two factors: the number of clicks and the rank position of the deepest click. Model validations reveal that incorporating these factors into an impression model (Impression Model 2) yields better performance in terms of predicting empirical continuation probabilities (see Figure 3.19) compared to the other models, including the click gap-based impression model proposed by Zhang et al. [244]. We have also demonstrated that impression models can be useful for computing correlation between predicted behaviour and observed behaviour based on evidence from click logs.

Our findings suggest that the improvement of search engine effectiveness should be guided by behavioural patterns derived from impression distributions. Having constructed tools for inferring observed behaviour (particularly conditional continuation probabilities) from interaction logs, the next step is to incorporate observed behaviours into the development of session-based effectiveness metrics. This issue is elaborated in Chapter 4. Chapter 5 introduces a framework for meta-evaluation of effectiveness metrics through the lens of C/W/L paradigm, and demonstrates its use for investigating the connection between observed behaviour and predicted behaviour, and between scores and user satisfaction, at both query- and session- levels.

Chapter 4
Modelling Search Sessions

The evaluation metrics described in Chapters 2 and 3 have focused on assessing the quality of search engine systems based on the assumption that users only submit a single query to address an information need. However, in practice users often reformulate their queries, or even refine their information needs during the course of the session. Hence, a search session typically consists of a sequence of queries.

To allow for the evaluation of the success of multi-query sessions, the traditional query-based test collections and evaluation metrics need to be generalised. This chapter proposes a framework for session-based effectiveness metrics by extending the existing query-based C/W/L framework, and then utilises the structure to instantiate an adaptive session metric, incorporating behavioural analysis results on commercial search interaction logs. In addition to the session-based effectiveness metric, this chapter also investigates what factors affecting session-level user satisfaction, building a fitted relationship between session satisfaction and the individual query satisfaction ratings (or query scores) when user observation data is available.

While discussing the motivation and research questions, Section 4.1 also describes *session test collections*, where each topic is assigned to a sequence of static queries, and the simulated users always follow the same static sequence when reformulating queries. Section 4.2 addresses previous work on session evaluation, including existing session-based effectiveness metrics and recent studies that explore the connection between session satisfaction and the individual queries.

Section 4.3 describes three sets of interaction logs drawn from two commercial search engines, and three datasets collected from lab-based user studies. These datasets are all utilised for exploring user behaviours and factors influencing session satisfaction.

Section 4.4 describes a session-based C/W/L framework, an extension to the existing query-based C/W/L structure. This is done by introducing a session-level behaviour, *con-*

The material in this chapter (Sections 4.4, 4.5, and 4.6) is currently under review.

ditional reformulation probability. With this extension, a session metric can be specified using both conditional continuation probability (query-level behaviour) and conditional reformulation probability. Section 4.5 establishes several factors that affect these two behaviours, using the commercial interaction logs described in Section 4.3. An adaptive session-based effectiveness metric is then devised using the factors inferred from those logged behaviours (Section 4.6).

When a set of sessions and their constituent queries, and set of corresponding per-session user-reported satisfaction ratings are available, it is interesting to explore factors influencing session satisfaction by establishing a fitted relationship between session-level ratings and factors from individual queries. The final sections of this chapter discuss this issue. Section 4.7 shows several factors that influence session satisfaction, employing the three lab-based datasets described in Section 4.3. Finally, Section 4.8 proposes two session satisfaction models, allowing position- and query-based factors to be combined.

4.1 Motivation and Research Question

After describing the research motivation and questions, this section discusses session effectiveness model at both SERP- and session levels from the perspective of the C/W/L framework, and the problem of predicting session satisfaction ratings.

4.1.1 Motivation

In practice, a user with an information need to resolve typically submits an initial query and examines results in the ranking. If they failed to find a sufficient number of relevant documents or to fulfil their information need when inspecting the initial ranking, they may repeatedly reformulate queries before they quit the session. For example, a sample of `Seek.com` interaction logs collected between July and August 2018 reveals the following two instances of query reformulation. One user commenced a job search using the initial query “child care”, and then reformulated that query into “child care educator”. Another user started with the query “underground surveyor”, before reformulating their initial query into a more general query, “surveyor”. Hence, a session $\mathcal{S}$ can be regarded as a chronologically-ordered sequence of queries:

$$\mathcal{S} = \langle Q_1, Q_2, Q_3, \ldots Q_{|\mathcal{S}|} \rangle,$$

where $|\mathcal{S}|$ is the number of queries submitted in the search session.

Jansen and Spink [95] study a sample of `Excite.com` Web search engine data collected in 2001 and `AltaVista.com` search query logs from 2002, and find that 55% of `Excite.com` users reformulated their initial queries, and that 47% of `AltaVista.com` queries were submitted in the context of similar queries. Jansen et al. [98] further define six classifications of query reformulation, which include *specialisation* and *generalisation*. A specialisation is when the reformulated query contains additional terms in order to seek more specific information, while a generalisation is when the reformulated query consists of fewer terms than the previous one [98]. Either type of reformulation occur with the assumption that the original information need does not change during the course of the session. The other four categories are “new”, “assistance”, “content change”, and “reformulation” [98].

Further experiments suggest that users tend to pose short queries several times, instead of only one verbose query, in sessions with a single information goal [100, 121]. There are several explanations of why users reformulate their queries. Turpin and Hersh [218] and Smith and Kantor [194] demonstrate that users are able to compensate for the reduced effectiveness of search engine systems by adapting their behaviours, and one such adaptation is to submit more queries. Järvelin et al. [103] carry out a laboratory-based interactive searching study, and find that in some cases initial queries do not give good results because users submit query terms that do not accurately cover the topic description.

Knowing that in practice users may reformulate their queries during a search session, it is valuable to generalise query-based effectiveness metrics to multi-query session evaluation metrics that capture in a single number the quality of a whole search session. Session evaluation is also identified as one of the long-range issues in IR (see, for example, the report from SWIRL 2018 by Allan et al. [9]).

4.1.2 Session Effectiveness Model

There have been several approaches to measurement of the quality of a multi-query session [103, 114, 115, 135]. Such mechanisms make use of session-based test collections, such as the one used in the TREC 2010 Session Track [114, 115]. As with query-oriented test collections, session-based ones consist of three components: a collection of documents; a set of topics; and a set of relevance judgements. In addition, each topic in the session test collection is associated with a fixed sequence of queries, with simulated users assumed to follow that sequence when they reformulate queries, and allowed to stop before the end of the sequence [103, 114, 115, 135].

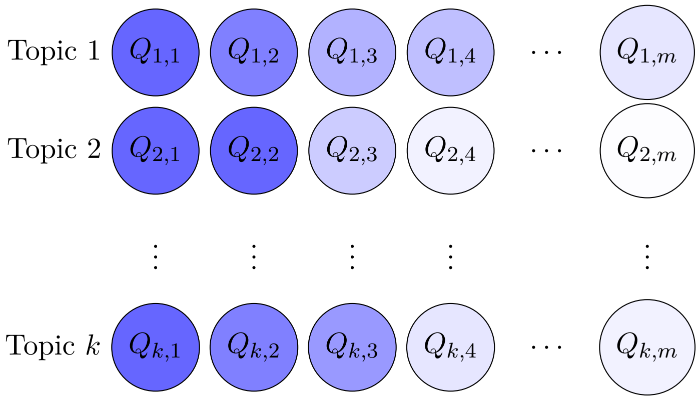
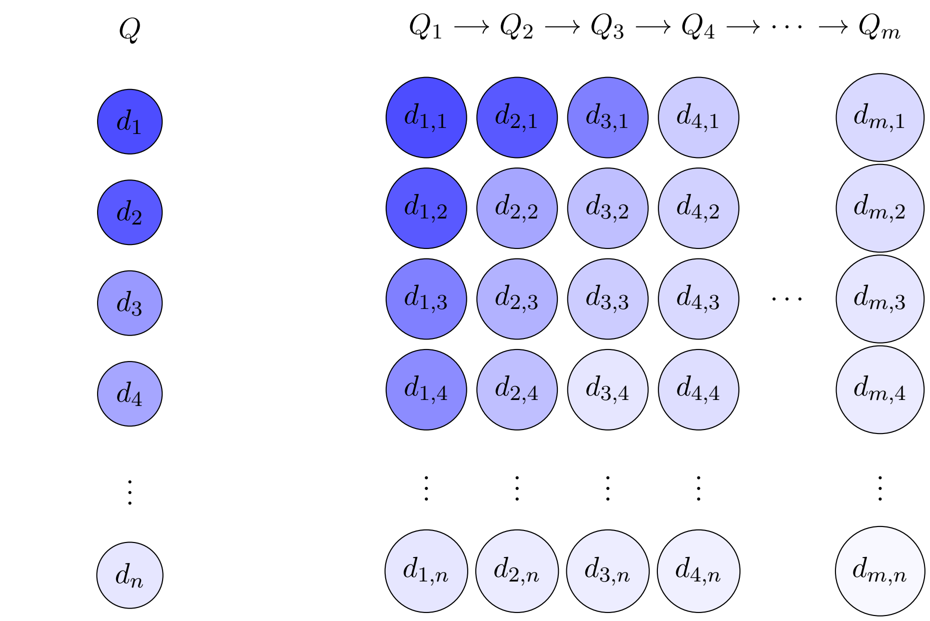


Figure 4.1: Illustration for a session test collection in which each of k topics is associated with a fixed sequence of m queries. The simulated user is assumed to always commence their search using the first query $Q_{l,1}$ for each topic l, and, reformulating the query $Q_{l,j}$ to $Q_{l,j+1}$ with a certain probability. The spectrum of blue color represents the overall probability that the simulated user enters query $Q_{l,j}$ in connection with topic l, where $1 \leq j \leq m$ and $1 \leq l \leq k$.

Figure 4.1 depicts an illustration for a session test collection, and Figure 4.2b provides a detail view for a particular topic. Note that, in Figures 4.1 and 4.2b, the number of queries the simulated user submits is not known, and that the score for a topic can be interpreted as the probability-weighted summation over all query scores in the static collection, with the weight associated with query Q_j related to the proportion of users in the population that would submit the sequence of queries $\langle Q_1, \cdots, Q_j \rangle$ when addressing that topic. To understand this model, note that most effectiveness metrics for single SERPs do not use specific knowledge of which documents the user examined (which could be obtained, for example, via an eye-tracking experiment). Instead, these query-based metrics usually make use of a certain assumption as to how users pay attention to the ranking, such as *top-weightedness* (as achieved via non-increasing discount function) [44, 102, 151, 195]. The session evaluation model portrayed in Figure 4.1 and Figure 4.2b is a two-dimensional equivalent of what happens when an effectiveness metric is applied to a single SERP in one dimension. Figure 4.2a and Figure 4.2b visually compare the two models.

The first goal of this chapter is to build a *predictive relationship* for an adaptive session effectiveness metrics. The resultant scores from this metric are intended to predict whole-session user performance when the end of query sequence is not known, and in doing so, extending the proposal of Moffat et al. [153, 155], who established the relationship between metrics, user models, and user behaviours. This chapter begins by investigating the empirical evidence for what had been postulated by Moffat et al. [153, 155] regarding


Q
d_1
d_2
d_3
d_4
d_n
$Q_1 \longrightarrow Q_2 \longrightarrow Q_3 \longrightarrow Q_4 \longrightarrow \cdots \longrightarrow Q_m$
$d_{1,1}$ $d_{2,1}$ $d_{3,1}$ $d_{4,1}$ $d_{m,1}$
$d_{1,2}$ $d_{2,2}$ $d_{3,2}$ $d_{4,2}$ $d_{m,2}$
$d_{1,3}$ $d_{2,3}$ $d_{3,3}$ $d_{4,3}$ $\cdots$ $d_{m,3}$
$d_{1,4}$ $d_{2,4}$ $d_{3,4}$ $d_{4,4}$ $d_{m,4}$
$d_{1,n}$ $d_{2,n}$ $d_{3,n}$ $d_{4,n}$ $d_{m,n}$

(a) SERP-based model (b) Session-based model

Figure 4.2: User interaction model for a particular topic in both query- (left-hand) and session-based test collections (right-hand). The depth of blue color represents the fraction of users in the universe who inspect a particular document (d_i for cases with a single query, or $d_{j,i}$ for cases with multiple queries and SERPs). Note that, in Figure 4.1 (page 124), an additional subscript l is used to represent topic dimension. In this figure, the subscript l is dropped, since a single topic is considered[1].

query-level user behaviours, using three large search interaction logs obtained from two commercial search engines, `Seek.com` and `Yandex.ru`. A model for session-level behaviours (that is, for query reformulation behaviours) is then developed, and is incorporated into session-based search effectiveness metrics, based on the user model framework described in Figure 4.3. In this framework, a user commences a search session by submitting a first query Q_j, $j = 1$, into an IR system. The user then sequentially inspects the ranked list of results from its top position $i = 1$, in effect, shown as the first vertical column depicted in Figure 4.2b. At each rank position i a decision is made: to continue inspecting the next item at rank $i + 1$; or to exit this j th SERP. In the latter case, the user then makes another choice: to issue a reformulated $j + 1$ th query; or to end the session entirely[2]. By imposing a probabilistic nature into this model, there are two quantities to be estimated:

[1] The remainder of the chapter will consider single topics, and hence it is useful to drop one of the subscripts, to avoid requiring triple subscripts on Q, d, or r.

[2] At the end of the session, the user can also choose to change the search engine system and start with a new query once again [212]. This, however, is outside the scope of this chapter.

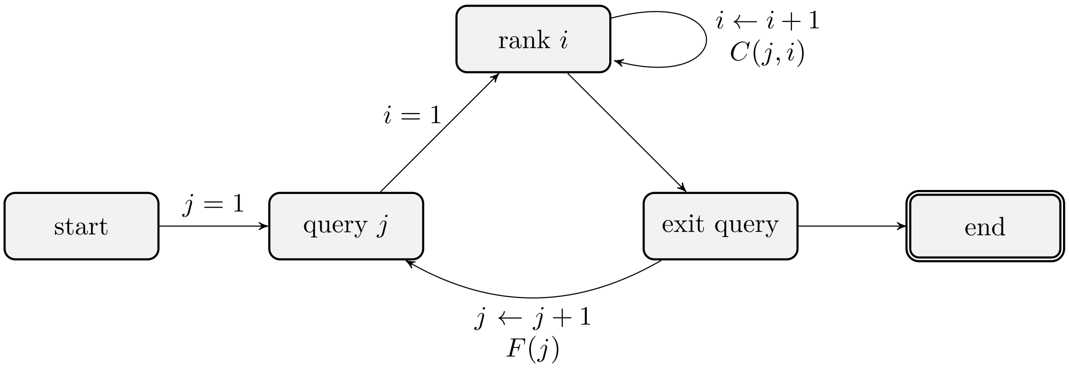


Figure 4.3: Search session model (adapted from the proposals by Moffat et al. [153] and by Thomas et al. [212]).

1. $C(j,i)$, the conditional continuation probability of the user inspecting the item at rank $i+1$ in the j th SERP, given that they have examined the item at rank i.
2. $F(j)$, the conditional probability of the user issuing a $j+1$ th query, given that they have just abandoned the j th SERP.

The former is a *query-level behaviour* and the latter is a *session-level behaviour*. From the perspective of the proposed framework, a range of interesting session-based metric can be characterised via these two functions. The challenge for the first goal is thus to find models for $C(j,i)$ and $F(j)$ that accurately predict observed behaviour. The following research question is formulated:

> **RQ 4.1**: In the context of Figure 4.3, what factors affect $C(j,i)$ and $F(j)$?

Section 2.3.2 (page 49) described several desired properties for user-oriented metrics. An ideal metric-based user model should be sensitive to the anticipated number of relevant documents for undertaking the search, or to different types of search tasks (goal-sensitive). Further, a metric should also be *adaptive*, meaning that the simulated user changes their behaviour as they encounter relevance in the part of the ranking seen so far. Hence, both $C(j,i)$ and $F(j)$ should be goal-sensitive and adaptive.

4.1.3 Observational Goal

Other researchers have also addressed a different goal in the context of session evaluation – an *observational* goal [104, 137, 139, 242]. Given the sequence of queries submitted by each individual user in a session, the challenge is to aggregate individual query scores (or query satisfaction ratings) via a weighting scheme that can be tuned to optimise a certain aspect, such as the relationship with user satisfaction [104, 137, 139, 242]. When fitting the relationship between session satisfaction ratings and the individual query scores, there is no probability-weighted sum and reformulation probabilities, since what queries the user posed and how they reformulated them are both known. For example, Zhang et al. [242] relate the weight for each query with the notion of *forgetfulness*, where the user tends to neglect the utility derived from earlier queries.

Figure 4.4 (page 128) provides an illustration of this *fitted relationship*, showing which queries in the sessions that dominantly contribute to the session satisfaction ratings. In Figure 4.4, five users (A, B, C, D, E) performed search activities under the same topic, and then provided 5-point session satisfaction ratings at the end of the sessions. Each SERP (or query) is associated with a score, which can be directly derived from a query-level satisfaction rating, or be computed via a particular query-based effectiveness metric. Suppose a session score is determined by linearly combining individual query scores in the session. The depth of red color visually describes query weights obtained by maximising the correlation between session scores generated by the linear combination model and session satisfaction ratings[3]. In this context, it is useful to investigate variables that contribute to the *best-fit* weights.

The second goal of this chapter is to address this challenge. The following research question is then considered:

> **RQ 4.2**: In the context of Figure 4.4, what factors influence session satisfaction?

Existing proposals for query-to-session aggregation mostly employ a linear combination approach, with the weights defined as a function of ordinal position of each query in the sequence [104, 137, 139, 242]. By using publicly-available user study data containing user-generated satisfaction ratings, this chapter explores the connection between SERP-level and session-level user satisfaction, including how satisfaction at the level of individual SERPs should be combined to predict satisfaction at session level. The insights gained from this analysis are then incorporated into two novel session satisfaction models, taking into account not only ordinal position of individual queries but also their qualities.

[3] Note that blue color is used to represent *the fraction of user attention* (see Figures 4.1 and 4.2), while red color is used to represent a different interpretation of weight, that is, the influence of a particular query on the session satisfaction (Figure 4.4).

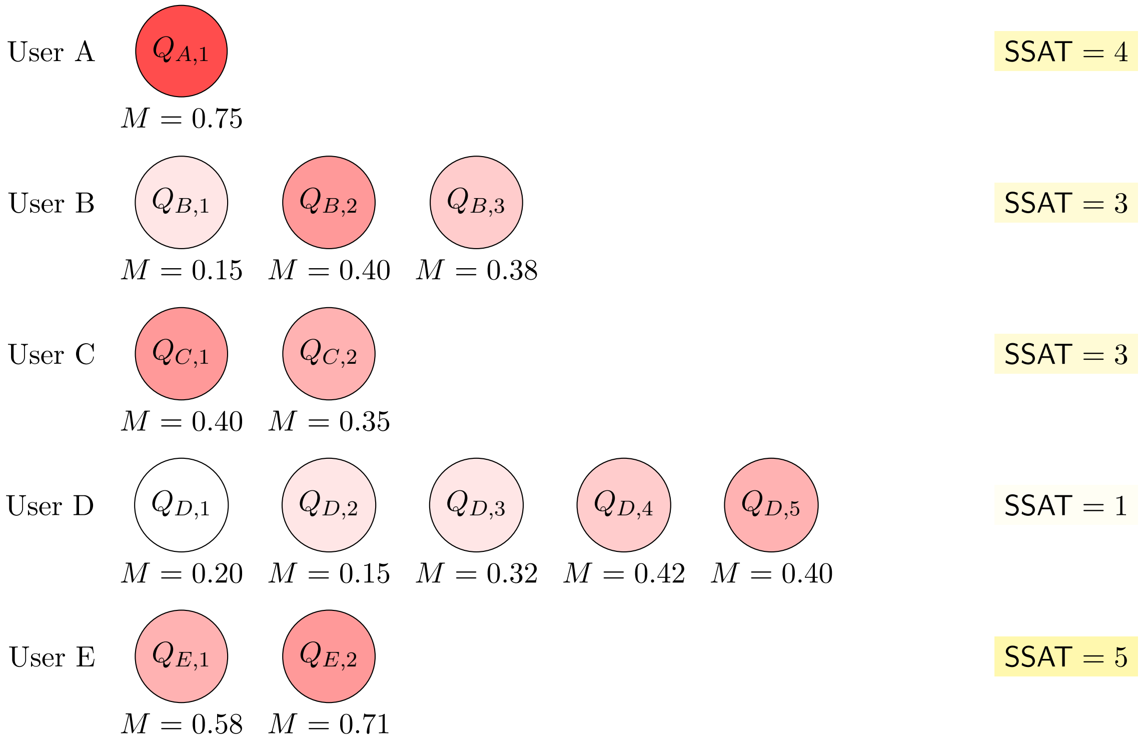


Figure 4.4: An illustration for a fitted relationship between individual query scores (denoted by $0 \leq M \leq 1$) and observed session satisfaction ratings ($\mathsf{SSAT} \in \{1, 2, 3, 4, 5\}$). The j th query submitted by user X is denoted by $Q_{X,j}$. The depth of red color represents the *weight* that corresponds to the influence of a particular query on the session satisfaction. In this illustration, the *best* and the *last* queries dominantly contribute to the session satisfaction.

4.2 Previous Work

This section presents existing session-based effectiveness metrics, such as session-based DCG and RBP, in the context of session test collection described in Figure 4.1 (page 124). Finally, this section describes recent work that builds a fitted relationship between individual query scores (or query satisfaction ratings) and session satisfaction ratings.

4.2.1 Session-Based Effectiveness Metrics

Kanoulas et al. [115] describe a session test collection as having a set of topics, with each topic associated with a sequence of static queries, as depicted in Figure 4.1. Let the number of queries in this static sequence be denoted by m (that is, an initial query, plus $m - 1$ reformulations). Several effectiveness metrics have been proposed for this kind of test collection, mostly based on ad-hoc assumptions about how users behave in a search session [103, 115, 135, 237].

Session-Based Discounted Cumulative Gain. Järvelin et al. [103] propose *session-based discounted cumulative gain* (sDCG), in which the DCG scores for the sequence of SERPs are discounted by $1/(1+\log_{bq} j)$, with $j \geq 1$ the index of the SERP in the session:

$$\begin{aligned} \mathsf{sDCG}(\overrightarrow{\mathbf{r}}; bq, b, m, n) &= \sum_{j=1}^{m} \frac{1}{(1+\log_{bq} j)} \cdot \mathsf{DCG@K}(\overrightarrow{r_j}; n, b) \\ &= \sum_{j=1}^{m} \sum_{i=1}^{n} \frac{1}{(1+\log_{bq} j) \cdot (1+\log_b i)} \cdot r_{j,i}\,, \end{aligned} \tag{4.1}$$

where $\overrightarrow{\mathbf{r}} = \langle \overrightarrow{r_1}, \overrightarrow{r_2}, \ldots, \overrightarrow{r_m} \rangle$ is the sequence of gain vectors associated with the SERPs in a session of length m; where $\overrightarrow{r_j} = \langle r_{j,1}, r_{j,2}, \ldots, r_{j,n} \rangle$ is the gain vector associated with the j th SERP in the session; where n is the length of each SERP; where b is the log base of DCG, with $b = 2$ the usual value; and where $1 < bq < 1000$ is the parameter governing the extent to which users reformulate their queries. With $1/(1+\log_{bq} j)$ being the session-level discount function, Järvelin et al. [103] argue that query reformulation involves additional effort, and thus SERPs returned by reformulated queries are less valuable than the one returned by the initial query. Järvelin et al. [103] further suggest using $bq = 4$. Note that Equation 4.1 employs a version of DCG that is different to the original version described by Järvelin and Kekäläinen [102].

Kanoulas et al. [115] propose a variant of sDCG using different discount functions at the session- and the SERP-levels. This version of sDCG is referred to as KsDCG, and employs a weaker penalisation for documents appearing in the later SERPs compared to the original sDCG:

$$\mathsf{KsDCG}(\overrightarrow{\mathbf{r}}; bq, b, m, n) = \sum_{j=1}^{m} \sum_{i=1}^{n} \frac{1}{\log_{bq}(j+bq-1) \cdot \log_b(i+b-1)} \cdot r_{j,i}\,.$$

Figure 4.5 shows the difference between two session-level discount functions of sDCG and KsDCG for $1 \leq j \leq 20$.

As is also the case with DCG (see Section 2.1.5 on page 29), for these two variants of sDCG, truncation at session length m and ranking depth n is necessary, since the infinite sum of the discount function is divergent. Note that the use of m and n is not to signify that the n th result of the m th SERP is the last point inspected by the user. The simulated

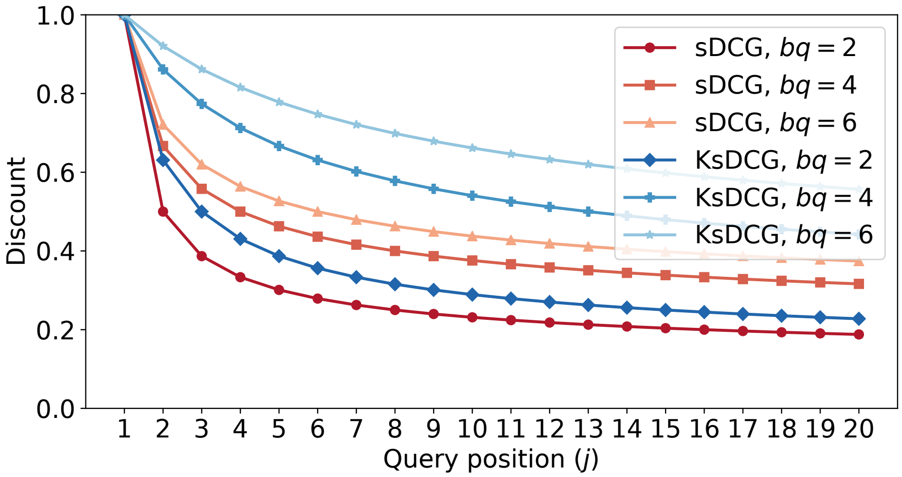


Figure 4.5: Plots of two session-level discount functions for sDCG and KsDCG: $1/(1+\log_{bq} j)$ and $1/\log_{bq}(j+bq-1)$, computed using $bq \in \{2,4,6\}$ and $1 \leq j \leq 20$.

user is assumed to never proceed beyond that point, but they also could stop prior to that point. The maximum scores of both sDCG and KsDCG increase as either ranking or session depth also increases, and hence require a scaling mechanism, if they are to be in the range from zero to 1.

As is shown later (see Section 4.4), continuation probabilities of these two session metrics increase with rank position i (sunk-cost property); and at the session-level, their conditional reformulation probabilities also comply with this property. However, neither of them are adaptive.

Session-Based Rank-Biased Precision. Recently, Lipani et al. [135] suggest a session-based RBP, derived from a probabilistic graphical model of user search activities:

$$\text{LCYsRBP}(\overrightarrow{\mathbf{r}}; p, q) = (1-p) \cdot \sum_{j=1}^{\infty} \sum_{i=1}^{\infty} \left[\left(\frac{p - q \cdot p}{1 - q \cdot p} \right)^{j-1} \cdot (q \cdot p)^{i-1} \cdot r_{j,i} \right] , \quad (4.2)$$

where $0 \leq p \leq 1$ is the user persistence parameter, and $0 \leq q \leq 1$ controls the balance between two actions: reformulation or keeping on inspecting items in the current SERP[4]. With this definition, both reformulation and continuation probabilities are constant during the course of the session, a concern that has also been raised in connection with RBP (see Section 2.4.1 on page 51).

[4] Lipani et al. [135] use $0^0 = 1$.

When $q = 1$, the simulated user will never reformulate the initial query, and hence LCYsRBP calculates an RBP score only for the first SERP, $\mathsf{RBP}(r_{1,*}; p)$. On the other hand, when $q = 0$, the simulated user will always reformulate upon the inspection of the first item in each SERP.

In contrast to both sDCG and KsDCG, the discount function of LCYsRBP gives a convergent sum, meaning that truncation at session length m and ranking depth n is not necessary. However, as with sDCG, LCYsRBP is not adaptive.

Other Session-Based Metrics. Kanoulas et al. [115] also propose a session-based metric esM computed as an expected score over all possible browsing paths in a session. Let this set of possible paths be denoted by Ψ. A path, $\psi \in \Psi$, is defined as a series of inspected items in a session [115]. The simulated user is also assumed to start examining from the top rank position of each SERP in the session. Consider the following path:

$$\psi_1 = \langle r_{1,1}, r_{1,2}, r_{1,3}, r_{2,1}, r_{3,1}, r_{3,2} \rangle \, .$$

In this example, the user submitted three queries in the session. They inspected the top three items in the first SERP, then reformulated to the second query, then examined only the first item in the second SERP, then issued the third query, then viewed at ranks 1 and 2 in the third SERP, and then ended the session. A path ψ also has a probability of $Pr(\psi)$ of being followed by a user, then

$$\mathsf{esM}(\Psi) = \sum_{\psi \in \Psi} Pr(\psi) \cdot M(\psi) \, ,$$

where $M(\psi)$ is the score of metric M computed for the browsing path ψ. Two RBP-like geometric distributions define $Pr(\psi)$: $p_{re}^{j-1}(1 - p_{re})$ denotes the probability that the j th query is the last one before the user ended their session; and $p_{down}^{i-1}(1 - p_{down})$ denotes the probability that the rank position i is the last one inspected by the user in a particular SERP. Kanoulas et al. [115] further suggest to use $p_{re} = 0.5$ and $p_{down} = 0.8$. Hence, esM is essentially an expected total gain version of LCYsRBP when $M(\psi)$ is a cumulative gain over all items in the path ψ. Yang and Lad [237] propose a similar approach, but without allowing the simulated users to abandon early queries. As is shown by our experiments, below, user models with fixed reformulation and continuation probabilities are not especially accurate.

Sakai and Dou [181] demonstrate that UM can be used to evaluate multi-query sessions when the knowledge of document length (that is, the number of characters) is available (see Equation 2.49 on page 58). Finally, when the amount of time spent to a examine a particular document is known, Cube Test (CT) metric can also be employed to measure the effectiveness of a session [142]. Let $D = \langle d_1, d_2, d_3, \ldots, d_{|D|} \rangle$ be a chronologically ordered sequence of documents inspected by the user, and D_t be a prefix of the sequence D, such that $D_t = \langle d_1, d_2, \ldots, d_t \rangle$. Luo et al. [142] then define CT as follows:

$$\mathsf{CT}(D) = \frac{1}{|D|} \cdot \sum_{t=1}^{|D|} \frac{\mathit{Gain}(D_t)}{\mathit{Time}(D_t)} \,,$$

where $\mathit{Gain}(D_t)$ is the gain volume that had been accumulated as a result of inspecting all documents in D_t, and $\mathit{Time}(D_t)$ is the time spent to examine all documents in D_t. Both UM and Cube Test do not have a parameter that accommodates the user's initial goal for commencing the search (such as, navigational or informational goals), and again are not adaptive.

4.2.2 Query-to-Session Aggregation Functions

When the sequence of queries submitted by each individual user and the corresponding satisfaction rating are observed, a fitted relationship between them can be established (consider again Figure 4.4 on page 128). In this regard one underlying question is to define a combination function over individual query scores in the sessions so that the aggregate session scores best correlate with session-level satisfaction ratings [104, 137, 139, 242]. Suppose the user has submitted $|\mathcal{S}|$ queries in a session:

$$\vec{\mathbf{r}} = \langle \vec{r_1}, \vec{r_2}, \vec{r_3}, \ldots, \vec{r_{|\mathcal{S}|}} \rangle \,,$$

where $\vec{r_j}$ is the relevance vector that corresponds to the j th SERP. One way to perform an aggregation is by employing a *linear combination*, defined as:

$$sM(\vec{\mathbf{r}}) = \sum_{j=1}^{|\mathcal{S}|} \theta(j) \cdot M(\vec{r_j}) \,, \tag{4.3}$$

where $0 \le \theta(j) \le 1$ is a weight associated with the j th SERP, and $M(.)$ is a query-based effectiveness metric. For example, Jiang and Allan [104] consider aggregation functions such as *summation* (that is, $\theta(j) = 1$) and *mean* (that is, $\theta(j) = 1/|\mathcal{S}|$). These functions, however, depend only on query position, denoted by j, and do not take into account query quality, denoted by $M(\overrightarrow{r_j})$.

Exponential Smoothing Function. Liu et al. [137, 139] argue that *recency* has a strong influence on session satisfaction, and suggest that an increasing weight function $\theta(j)$ is more appropriate for combining query-level metric scores. They define $sM(.)$ inductively using an *exponential smoothing* technique:

$$\begin{aligned} sM(\langle\overrightarrow{r_1}\rangle) &= M(\overrightarrow{r_1}) \\ sM(\langle\overrightarrow{r_1},\overrightarrow{r_2}\ldots,\overrightarrow{r_{|\mathcal{S}|}}\rangle) &= \left(1-\frac{1}{|\mathcal{S}|^\lambda}\right)\cdot sM(\langle\overrightarrow{r_1},\overrightarrow{r_2}\ldots,\overrightarrow{r_{|\mathcal{S}|-1}}\rangle)+\frac{1}{|\mathcal{S}|^\lambda}\cdot M(\overrightarrow{r_{|\mathcal{S}|}})\,, \end{aligned}$$

where parameter $0 \leq \lambda \leq 1$ models the decay of influence. When $\lambda = 0$, the session-level score is the score of the last query; and when $\lambda = 1$, $sM(.)$ simplifies to a *mean* aggregation function. More generally, this leads to the following specification:

$$\theta_{\mathsf{Liu}}(j) = \frac{1}{j^\lambda}\cdot\prod_{k=j+1}^{|\mathcal{S}|}\left(1-\frac{1}{k^\lambda}\right)\,. \tag{4.4}$$

As with the proposal of Jiang and Allan [104], $\theta_{\mathsf{Liu}}(j)$ does not consider query quality.

Decaying of Memories. Inspired by the work of Liu et al. [139], Zhang et al. [242] propose a forgetting function, *forget*(j), that corresponds to the j th query in the session with length $|\mathcal{S}|$:

$$\mathit{forget}(j) = e^{-\delta\cdot(|\mathcal{S}|-j)}\,, \tag{4.5}$$

where δ is the rate at which users forget previously issued queries. Zhang et al. [242] then combine this forgetting function and two session-based metrics, sRBP and sDCG, to develop two recency-aware aggregation functions. The proposed functions are referred to as recency-aware session RBP (RSRBP) and recency-aware session DCG (RSDCG):

$$\begin{aligned} \mathsf{RSDCG}(\overrightarrow{\mathbf{r}}) &= \sum_{j=1}^{|\mathcal{S}|}\mathit{forget}(j)\cdot\frac{1}{(1+\log_{bq} j)}\cdot\sum_{i=1}^{n}\frac{1}{(1+\log_b i)}\cdot r_{j,i} \\ \mathsf{RSRBP}(\overrightarrow{\mathbf{r}}) &= \sum_{j=1}^{|\mathcal{S}|}\mathit{forget}(j)\cdot\left(\frac{p-q\cdot p}{1-q\cdot p}\right)^{j-1}\cdot\sum_{i=1}^{n}(q\cdot p)^{i-1}\cdot r_{j,i}\,. \end{aligned}$$

Note that *forget*(j) depends on the knowledge of $|\mathcal{S}|$, the number of queries submitted in the session. Therefore, RSRBP and RSDCG are not principally intended for session test collection described in Figure 4.1, since the query sequence end point is known. Using the framework described in Equation 4.3 (page 132), both functions can also be specified as follows:

$$\theta_{\mathsf{RSDCG}}(j) = \frac{e^{-\delta\cdot(|\mathcal{S}|-j)}}{(1+\log_{bq} j)} \quad , \quad M_{\mathsf{RSDCG}}(\overrightarrow{r_j}) = \sum_{i=1}^{n} \frac{r_{j,i}}{(1+\log_b i)} \, ;$$

$$\theta_{\mathsf{RSRBP}}(j) = e^{-\delta\cdot(|\mathcal{S}|-j)} \cdot \left(\frac{p-q\cdot p}{1-q\cdot p}\right)^{j-1} \quad , \quad M_{\mathsf{RSRBP}}(\overrightarrow{r_j}) = \sum_{i=1}^{n} (q\cdot p)^{i-1} \cdot r_{j,i} \, .$$

Both $M_{\mathsf{RSDCG}}(.)$ and $M_{\mathsf{RSRBP}}(.)$ are SERP-based expected total gain (ETG) metrics. Moreover, $M_{\mathsf{RSRBP}}(.)$ is an ETG version of RBP with $\phi = q \cdot p$.

Like these examples, most composition functions are based on ordinal query position alone. A key finding of this chapter is that shows that quality information – the spectrum from best to worst queries – is also valuable, and that combining both position and quality information leads to session scores that better correlate with session satisfaction.

4.3 Interaction Logs

We now describe the search interaction logs used in this study. Three datasets are drawn from commercial search engines; three further datasets are pre-existing resources from lab-based user studies [107, 139, 145].

4.3.1 Industrial-Based Datasets

Table 4.1 describes the three search interaction logs used in this study. The first two are from the Australasian job search engine, `Seek.com`, sampled through an eight-week period (30 July to 23 September 2018). Note that this is also the same period used to sample the data described in Chapter 3 (see Table 3.1 on page 80). However, the data described in this chapter was collected in the unit of "session", instead of "action sequence", since session-level behaviours are the main focus in this chapter. Recall that the `Seek.com` data covers two modalities: via a mobile application (iOS and Android) that has infinite scrolling without pagination; and via a traditional browser-based Web application that has fixed pages containing 20 results. In the browser-based modality, the first page in each SERP might also contain up to two further paid items. The two modalities are based on different

	Seek.com		Yandex
	mobile	browser	
Users	4,962	4,868	unknown
Sessions	56,737	19,269	1,000,000
Queries	121,840	75,401	1,362,421
SERP size	unlimited	unlimited	truncated,10
Pagination	no	20–22	no
Domain	jobs	jobs	web
Clicks	yes	yes	yes
Impressions	yes	yes	no
Rel. judg.	no	no	binary

Table 4.1: Interaction logs from three commercial search engines (mobile- and browser-based Seek.com job search; and Yandex.ru Web search). Note that these logs are used to investigate search interaction patterns, and not to address system performances of both Seek.com and Yandex.ru.

	J&A [107]	THUIR2 [145]	THUIR3 [139]
Sessions	80	223	450
Avg. queries per session	4.85	4.11	3.44
SERP size	9	5	10
Rel. judgements	3-level	4-level	4-level
Impressions	eye gazes	no	no
Clicks	yes	yes	yes
Query ratings	no	5-level	5-level
Session ratings	5-level	5-level	5-level

Table 4.2: Data from three lab-based sessional Web search studies.

definitions of search session. On mobiles, a new session starts each time the application is opened, whereas with browser-based search a session is associated with a long-lived cookie that could last for months.

The third set of interaction data was obtained from a publicly-available Web search log of a Russian search engine, Yandex.ru. This dataset, which was initially constructed for a relevance prediction competition[5], contains a collection of ordered clickthrough sequences (rather than impressions or other actions), as well as a set of relevance judgements for a subset of the queries, made a year after the logs had been collected. In general, Yandex.ru Web search users are less inclined to reformulate than Seek.com job search users. Figure 4.6 shows the fraction of sessions, categorised by the session length, in these three datasets.

[5] https://academy.yandex.ru/events/data_analysis/relpred2011/

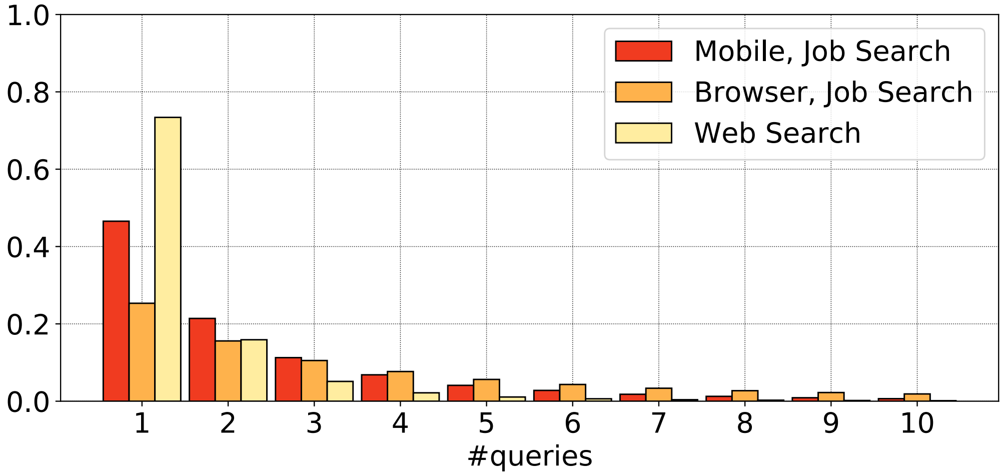


Figure 4.6: Fraction of sessions for job search and Web search, as a function of session length.

4.3.2 Laboratory-Based Datasets

As shown in Table 4.2, three laboratory-based session search logs are also employed in support of our analysis and experiments, and are particularly useful for investigating factors affecting session satisfaction, since they contain user-reported satisfaction ratings. Hence, these datasets are important for the development of query-to-session aggregation methods. Two datasets, J&A and THUIR3 datasets, have been used already in Chapter 3 for inferring SERP-level conditional continuation functions in the context of Web search (see Table 3.15 on page 115). In this chapter we are more interested in exploring the session-level information available from these three datasets. Observations from the J&A dataset are also useful for inferring user viewing behaviours for the `Yandex.ru` logs, relying on the fact that both arise from typical web search scenarios.

In addition to the J&A and the THUIR3 datasets, the analysis carried out in this chapter employs the THUIR2 dataset, developed by Mao et al. [145], which contains 223 multi-query sessions. Three features of the THUIR2 data are particularly useful: a set of SERPs with four-level relevance judgements covering the top-5 rank positions for each SERP; a set of five-level user-generated satisfaction ratings at the level of individual SERPs; and a set of five-level whole-of-session satisfaction ratings that serve as an overall evaluation of each session. The THUIR2 dataset has the same characteristics as THUIR3. However, THUIR3 has more sessions compared to THUIR2, and also contains relevance judgements for the top ten results in each SERP.

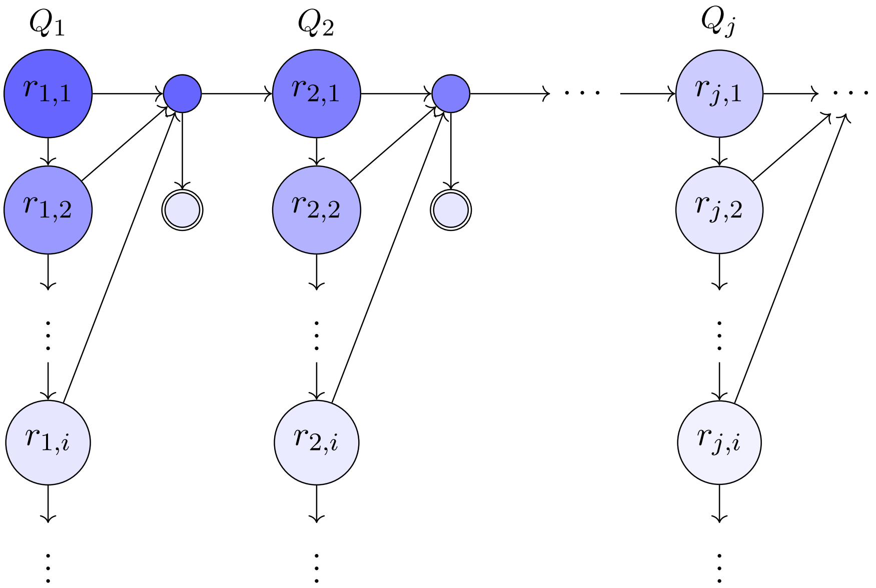


Figure 4.7: Unrolling Figure 4.3 to obtain possible browsing paths.

4.3.3 Organic SERPS

It can be helpful to explore the query-level behaviour of users when they are engaged with organic results and see no paid items, and to get a clear understanding of user activities around page boundaries. These subsets of the two `Seek.com` logs contain 58,645 organic SERPs derived from 3,970 mobile-based users, and 40,882 organic SERPs from 4,003 browser-based users.

4.4 A Session-Based C/W/L Framework

To obtain a session-based framework we add a second dimension to the query-level C/W/L definitions, spanning not only all results in a SERP but also all items across all SERPs in a session. Figure 4.7 (page 4.7) illustrates this idea by unrolling the processes described in Figure 4.3 (page 4.3), showing the set of browsing paths a user might follow through a complete search session. Two base cases are considered. First, the user is assumed to always start the session by submitting an initial query Q_1, and then inspecting the first item in the corresponding SERP to derive the gain $r_{1,1}$. Second, if the user has reformulated the initial query $j-1$ times, and is about to examine the j th SERP after submitting Q_j, then they always read the first result in that SERP to derive the gain $r_{j,1}$.

Other than the two base cases, the general situation is that the user has just inspected the item at rank i in the j th SERP. At each such step they make a binary decision: to continue inspecting the next item at rank $i+1$ in the same SERP, or to exit this SERP. This choice is modelled by $C(j,i)$, the conditional continuation probability of the user shifting their attention from the i th to the $i+1$ th item in the j th SERP. Note that $C(j,i)$ is a generalisation of the SERP-level $C(i)$.

If the user chooses to abandon the SERP for query Q_j, they might reformulate Q_j to a follow-up $j+1$ th query Q_{j+1}, with conditional probability $F(j)$; or, might end the entire search session with probability $1-F(j)$. That is, $C(j,i)$ and $F(j)$ are the two quantities required when computing the proportion of users in the population looking at the i th document in the j th SERP, and thus determine a set of weights that can be used to calculate a session effectiveness metric score.

As with the query-based C/W/L structure, it is also possible to compute the probability $L(j,i)$ that the i th item in the j th SERP is the last one inspected by the user, denoted by $L(j,i)$; and the weight function $W(j,i)$ that represents the fraction of user attention paid to the i th item in the j th SERP. It is important to note that both $W(j,i)$ and $L(j,i)$ are two-dimensional probability functions, and hence should sum to one: $\sum_j \sum_i W(j,i) = \sum_j \sum_i L(j,i) = 1$. Similarly, $C(j,i)$, $W(j,i)$, and $L(j,i)$ for any single SERP can be computed from each other. For example, $C(j,i)$ can be computed from $W(j,i)$ using

$$C(j,i) = W(j,i+1)/W(j,i)\,. \tag{4.6}$$

The conditional reformulation probability, $F(j)$, also has a relationship to any of the $C(j,i)$, $W(j,i)$, and $L(j,i)$ functions. For example, $F(j)$ and $W(j,i)$ are connected via

$$F(j) = W(j+1,1)/W(j,1)\,. \tag{4.7}$$

As can be seen in Figure 4.7, $W(j,1)$ is proportional to the fraction of users who examine the SERP for j th query, since users who arrive at the j th SERP must examine its first document at the top.

Expected Number of Inspected Items. The expected number of inspected items (ENI) for the session-based C/W/L structure is computed as follows:

$$\mathsf{ENI} = 1/W(1,1)\,. \tag{4.8}$$

To understand this relationship, consider the following sequence of observations:

- The per-SERP expected search length for jth SERP in the session, denoted by $\mathsf{ESL}(j)$, is computed as

$$\mathsf{ESL}(j) = \left(\sum_{i=1}^{\infty} W(j,i)\right) / W(j,1)\,.$$

- Knowing that $W(j,i)$ is non-increasing with respect to query position j and the infinite sum of $W(j,i)$ over $j \geq 1$ is convergent, $\sum_{j=1}^{\infty} W(j,i) \leq 1$, the following statement holds:

$$\lim_{j\to\infty} W(j,i) = 0 \quad \text{for all } i \geq 1\,.$$

- Using Equation 4.7 (page 138), the probability that the user submits exactly k queries before ending the search session, denoted by $P(|\mathcal{S}| = k)$, is computed as:

$$\begin{aligned} P(|\mathcal{S}| = k) &= (1 - F(k)) \cdot \prod_{j=1}^{k-1} F(j) \\ &= \frac{W(k,1) - W(k+1,1)}{W(1,1)}\,. \end{aligned}$$

If the query sequence is observed and it is known that the user poses k queries, $\mathcal{S} = \langle Q_1, \cdots, Q_k \rangle$, the expected number of documents inspected upon exit from kth SERP is determined as $\sum_{j=1}^{k} \mathsf{ESL}(j)$. However, when the number of query reformulations is not known, but its probability distribution is known, the expected value for the number of inspected items in the session (that is, ENI) can be computed as:

$$\begin{aligned} \mathsf{ENI} &= \mathsf{ESL}(1) \cdot P(|\mathcal{S}| = 1) + \\ &\quad [\mathsf{ESL}(1) + \mathsf{ESL}(2)] \cdot P(|\mathcal{S}| = 2) + \\ &\quad \cdots \\ &= \frac{\sum_{i=1}^{\infty} W(1,i)}{W(1,1)} \cdot \frac{W(1,1) - W(2,1)}{W(1,1)} + \\ &\quad \left[\frac{\sum_{i=1}^{\infty} W(1,i)}{W(1,1)} + \frac{\sum_{i=1}^{\infty} W(2,i)}{W(2,1)}\right] \cdot \frac{W(2,1) - W(3,1)}{W(1,1)} + \\ &\quad \cdots \end{aligned}$$

$$\begin{aligned}
= \; & \frac{\sum_{i=1}^{\infty} W(1,i)}{W(1,1)} \cdot \left[\frac{W(1,1) - W(2,1)}{W(1,1)} + \frac{W(2,1) - W(3,1)}{W(1,1)} + \ldots\right] + \\
& \frac{\sum_{i=1}^{\infty} W(2,i)}{W(2,1)} \cdot \left[\frac{W(2,1) - W(3,1)}{W(1,1)} + \frac{W(3,1) - W(4,1)}{W(1,1)} + \ldots\right] + \\
& \ldots \\
= \; & \frac{\sum_{i=1}^{\infty} W(1,i)}{W(1,1)} \cdot \left[\frac{W(1,1)}{W(1,1)}\right] + \\
& \frac{\sum_{i=1}^{\infty} W(2,i)}{W(2,1)} \cdot \left[\frac{W(2,1)}{W(1,1)}\right] + \\
& \ldots \\
= \; & \frac{\sum_{i=1}^{\infty} W(1,i) + \sum_{i=1}^{\infty} W(2,i) + \ldots}{W(1,1)} \\
= \; & \frac{\sum_{j=1}^{\infty} \sum_{i=1}^{\infty} W(j,i)}{W(1,1)} \\
= \; & 1/W(1,1)\,.
\end{aligned}$$

ERG and ETG Metrics. Given the definition of the session-based C/W/L framework, the session-based expected rate of gain (*ERG*) metric is defined via:

$$sM_{ERG}(\overrightarrow{\mathbf{r}}) = \sum_{j=1}^{\infty} \sum_{i=1}^{\infty} W(j,i) \cdot r_{j,i}\,.$$

Further, using the expected number of inspected documents described in Equation 4.8, the expected total gain (*ETG*) metric is computed as:

$$sM_{ETG}(\overrightarrow{\mathbf{r}}) = sM_{ERG}(\overrightarrow{r})/W(1,1)\,.$$

Computing $W(j,i)$. To determine the $W(j,i)$ values, let $V(j,i)$ be the proportion of users that examine the i th result in the SERP associated with the j th query. The user is assumed to always look at the first item listed in the first SERP, and hence: $V(1,1) = 1$. If function $F(j)$ is not affected by $r_{j,i}$, and depends only on query position j and per-SERP item position i, then the conservation of flow described in Figure 4.7 (page 137) suggests that $V(j,i)$ can be inductively computed as:

$$V(j,i) = \begin{cases} 1 & j = 1,\, i = 1 \\ F(j-1) \cdot V(j-1,1) & j > 1,\, i = 1 \\ V(j,i-1) \cdot C(j-1,i) & i > 1\,. \end{cases} \tag{4.9}$$

The weight function $W(j,i)$ is then a normalisation of $V(j,i)$:

$$W(j,i) = \frac{V(j,i)}{\sum_k^\infty \sum_l^\infty V(k,l)}\,. \tag{4.10}$$

Here, the denominator $\sum_k^\infty \sum_l^\infty V(k,l)$ should be precisely known in order to determine an accurate $W(j,i)$. In practice, this can be achieved by calculating many values of $V(j,i)$. Note that, with this definition, the ETG score (Equation 4.9) can also be computed as: $\sum_{j=1}^\infty \sum_{i=1}^\infty V(j,i) \cdot r_{j,i}$.

However, if function $F(j)$ depends on the gain values that have been encountered by the user so far, and the metric is adaptive, Equations 4.9 and 4.10 cannot be applied, since each random user in the population might have different total accumulated gain upon exit from a particular SERP. In this case, a more general computation method, Monte Carlo simulation, can be applied to calculate $W(j,i)$, counting what happens through the course of a large number of randomised trials using the automaton shown in Figure 4.3 (page 126).

Existing Session Metrics. Given these definitions, three existing session-based metrics can be explained using the C/W/L framework: the original session-based DCG (sDCG) [103], a variant of sDCG proposed by Kanoulas et al. [115] (KsDCG) , and LCYsRBP [135]. The latter is an *ERG* metric with two parameters, and is defined as:

$$C_{\mathsf{LCYsRBP}}(j,i) = q \cdot p \quad \text{and} \quad F_{\mathsf{LCYsRBP}}(j) = \frac{p - q \cdot p}{1 - q \cdot p}\,.$$

For both sDCG and KsDCG, truncation at a defined evaluation depth is required, since the infinite sum of the original discount function does not converge. Two additional parameters, m and n, are employed to limit the computation over a sequence of m SERPs, and to depth of n in each SERP. This is to ensure that the simulated user will end their search activities at some point. Without this truncation, the user will never stop searching, and thus $W(j,i) \approx 0$ and the metric score becomes zero. Hence, sDCG and KsDCG are two instances of an *ETG* metric with four parameters – bq and b, representing the session- and query-level persistence, the session depth m, and the ranking depth n – and are specified by:

$$C_{\mathsf{sDCG}}(j,i) = \begin{cases} \frac{1+\log_b(i)}{1+\log_b(i+1)} & i < n \\ 0 & i \geq n \end{cases}, \quad F_{\mathsf{sDCG}}(j) = \begin{cases} \frac{1+\log_{bq}(j)}{1+\log_{bq}(j+1)} & j < m \\ 0 & j \geq m\,. \end{cases}$$

$$C_{\mathsf{KsDCG}}(j,i) = \begin{cases} \frac{\log(i+b-1)}{\log(i+b)} & i < n \\ 0 & i \geq n \end{cases}, \quad F_{\mathsf{KsDCG}}(j) = \begin{cases} \frac{\log(j+bq-1)}{\log(j+bq)} & j < m \\ 0 & j \geq m\,. \end{cases}$$

With this specification, the user modelled by LCYsRBP has a constant tendency to reformulate their queries, regardless of how many queries they have submitted and how many relevant items they have seen; whereas both sDCG and KsDCG assume that $F(j)$ increases with query position j. However, none of them are adaptive.

4.5 Search Behaviours

Having described the session-based C/W/L structure consisting of two key quantities, conditional continuation probability (query-level behaviour) and conditional reformulation probability (session-level behaviour), this section employs interaction logs from two commercial search engines, `Seek.com` and `Yandex.ru`, to answer **RQ 4.1** regarding factors contributing to both levels of user behaviour.

4.5.1 Query-Level Behaviours

The interaction logs are used to explore query-level user behaviours through the lens of C/W/L, focusing on how $C(\cdot,\cdot)$ varies with respect to: (1) the rank position currently being inspected, i; (2) the anticipated number of relevant items to fulfill an information need, T; and (3) the unmet volume of relevance after i documents have been encountered. These three factors were originally used by Moffat et al. [155] to define an adaptive query-based metric, INST.

Inferring $C(\cdot,\cdot)$. To investigate the effect of those three factors on $C(\cdot,\cdot)$, it is necessary to compute empirical estimates from the action sequences. Chapter 3 described a methodology for computing empirical $C(\cdot,\cdot)$ from search logs. With this estimation method, each action in action sequence is associated with an indicator variable $0 \leq \mathbf{c}_i \leq 1$, where $\mathbf{c}_i = 1$ if a continuation is observed at that position, and $\mathbf{c}_i = 0$ otherwise. Recall that three rules were proposed for the operational definition of a continuation, specifying how to determine the value of $\mathbf{c}_i$ in each action in the action sequence (see Table 3.2 on page 82).

In contrast to the `Seek.com` data, the `Yandex.ru` data only contains clickthroughs and no impression signals. As is noted in Section 3.5, this situation potentially underestimates the notion of "examine rank i". The phenomenon of good abandonment suggests that a

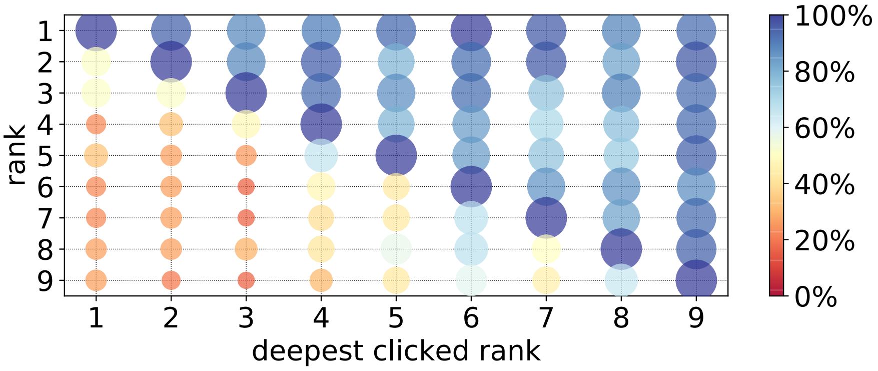


Figure 4.8: Percentage of action sequences in which an impression at a particular rank position is observed, stratified by the rank position of the deepest click action (J&A dataset).

user should be considered to have examined the item at rank i when they read the snippet at that point without clicking at it [133]. To address this underestimation issue, Section 3.5 described impression models that predict impression distribution using clickthrough information, exploiting a behavioural pattern that users tend to examine the majority of the items prior to the deepest click rank position, as well as results beyond that point (see Figure 3.14 on page 99).

Figure 4.8 shows that this pattern is also supported by the J&A dataset. To infer gaze distributions from clickthrough sequences, including beyond the deepest click, Impression Model 1 (see Equation 3.7 on page 107) is employed to estimate impression distributions for the `Yandex.ru` log using the J&A dataset as a basis for parameter estimation. Recall that Impression Model 1 has one parameter, K, that represents user persistence beyond the deepest click position. Using this model, the continuation variable, $\mathbf{c}_i$, is determined as follows:

$$\mathbf{c}_i = \frac{\hat{V}(i+1 \mid u,q)}{\hat{V}(i \mid u,q)} = \begin{cases} 1 & \text{if } i \leq DC(u,q) \\ e^{-1/K} & \text{if } i > DC(u,q)\,, \end{cases} \tag{4.11}$$

where $\hat{V}(i \mid u,q)$ is the probability that user u inspects the item at rank i for query q, and $DC(u,q)$ is the corresponding deepest click rank. A best-fit process using the J&A dataset yields $K = 1.4$.

Inferring Relevance. To understand how T_i influences $C(\cdot,\cdot)$, the relevance information for each item in each SERP needs to be available. However, the `Yandex.ru` dataset (Table 4.1) only contains relevance judgements for clicked items. To infer the relevance of

Θ	$\hat{P}(r = 0 \mid \Theta)$	$\hat{P}(r = 1 \mid \Theta)$	$\hat{P}(r = 2 \mid \Theta)$	$\mathbb{E}[r]$
1	0.263	0.301	0.436	1.173
2	0.518	0.211	0.271	0.753

Table 4.3: Estimated probability of three different relevance levels when the user clicked on a particular item ($\Theta = 1$), and when the user viewed an item but did not click on it ($\Theta = 2$), computed from J&A dataset with 3-level relevance judgements ($r \in \{0, 1, 2\}$). The last column shows the expected relevance grade for both conditions. Note that the sum in each of the two rows is 1.0. The difference between the two conditions is significant ($\chi^2_2 = 125.7, p < 0.01$).

non-clicked items, this study makes use the assumption that documents that are viewed but not clicked are not relevant. Table 4.3, which is generated from the J&A dataset, shows that the expected relevance of a clicked item is significantly higher than that of an item that is viewed but not clicked. This observation provides evidence for that assumption.

With the `Seek.com` data, the situation is more complex. The `Seek.com` dataset does not contain editorial relevance judgements, and the relevance information needs to be inferred from available implicit feedback, such as clickthroughs and job applications. While clickthrough information, at least to some extent, can be used for modelling relevance [41, 162], it is difficult to directly interpret clickthroughs as absolute relevance judgements [110]. A clickthrough action on a particular item is an indication that the user was attracted by that item, and perceived that it might be relevant. However, after clicking the item and reading its content, the user might decide that it is non-relevant. Fortunately, in the context of job search, the application action is more tightly coupled with relevance than the clickthrough action.

To support that argument, a logistic regression analysis was carried out with a relevance value (r_i) as the response variable, and two binary indicators, $\mathsf{did\_click}$ and $\mathsf{did\_app}$ associated with clickthrough and job application actions, as the explanatory variables. This regression analysis employs a small set of relevance judgements (qrels) consisting of 5,145 $\langle \mathit{query}, \mathit{document}, r_i \rangle$ triples on a four-point relevance scale. These qrels were gathered using the 205 most frequent queries, via a crowd-sourcing platform between June and August 2017[6], before the introduction of the "impression tracking" system that generated the data described in Table 4.1.

Using the `Seek.com` qrels, a sample of search logs containing around 7,000,000 tuples $\langle \mathit{query}, \mathit{document}, r_i, \mathsf{did\_click}, \mathsf{did\_app} \rangle$ were also collected for the duration between

[6] These qrels were collected by Damiano Spina (RMIT University) and Bahar Salehi (The University of Melbourne) as part of the overall collaboration between the two institutions and `Seek.com`. The contribution of Damiano and Bahar in sharing the resource that they constructed is acknowledged.

Factor	coef.	p
intercept (w_0)	2.721	0.000
did_click (w_1)	0.527	0.000
did_app (w_2)	0.173	0.000

Table 4.4: Multiplicative effect sizes for did_click and did_app, optimised to fit the Seek.com editorial relevance values using a logistic regression for a total of 5,970,120 tuples $\langle query, document, r_i, \textsf{did\_click}, \textsf{did\_app}\rangle$.

February and November 2017. Let this collection of tuples be denoted by H. To estimate the coefficients of the regression model, we only used a subset of the logs, $H' \subset H$ that contains 5,970,120 tuples, for which the query-document pairs are either fully-relevant or non-relevant. Note that a logistic regression model requires a binary response variable. The coefficients $\mathbf{w}$ of explanatory variables are then optimised according to the following linear model:

$$\ln(r_i/(1-r_i)) = w_0 + w_1 \cdot \textsf{did\_click} + w_2 \cdot \textsf{did\_app}\,.$$

Table 4.4 shows the effect sizes for did_click and did_app, when optimised to fit the editorial relevance values. Note that a job application is always preceded by a clickthrough, and hence:

$$\textsf{did\_app} = 1 \implies \textsf{did\_click} = 1\,.$$

The relatively large positive intercept value might be affected by the fact that the majority of the $\langle query, document\rangle$ pairs in the sample of search logs are fully relevant. The corresponding odds ratios are exp(0.527) = 1.694 (a 69% increase in the odds of being relevant when clicked) and exp(0.527 + 0.173) = 2.014 (a 101.4% increase in the odds of being relevant (or double odds of being relevant) when also applied for.

To reinforce the finding described in Table 4.4, two quantities are also computed: clickthrough rate (CTR) and application probability conditioned on click action, $P(App \mid Click)$, using the original collection of tuples H. The results are then stratified by the relevance grade (r_i). Both clickthrough rate and $P(App \mid Click)$ for a query-document pair are estimated as follows:

$$CTR(\langle query, document\rangle) \;=\; \frac{\#Click(\langle query, document\rangle)}{\#Count(\langle query, document\rangle)}\,,$$

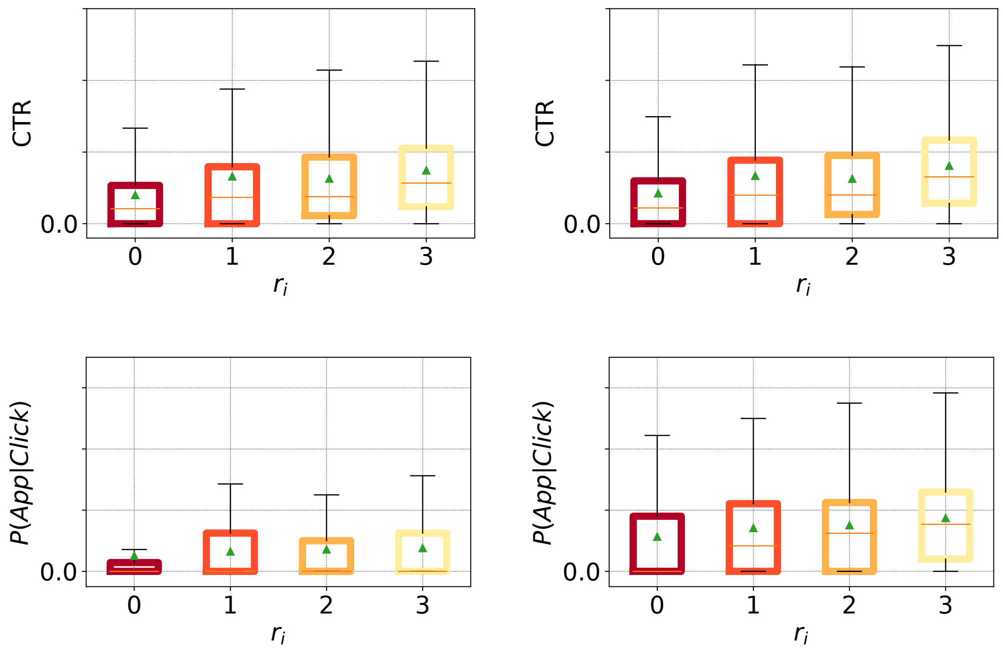


Figure 4.9: Clickthrough rate (top row) and distribution of $P(App \mid Click)$ (bottom row), each as a function of relevance grade (r_i), computed from the collection of tuples H. The green triangle in each box-whisker is the mean. Mobile-based queries are shown in the left-hand column, and browser-based queries in the right-hand column. The vertical scale is linear; but for commercial-in-confidence reasons is not labeled.

$$P(App \mid Click, \langle query, document\rangle) \quad = \quad \frac{\#App(\langle query, document\rangle)}{\#Click(\langle query, document\rangle)}\,,$$

where $\#Count(.)$ is the frequency of the $\langle query, document\rangle$ pair in the set of tuples H, and $\#Click(.)$ and $\#App(.)$ are the numbers of occurrences of the $\langle query, document\rangle$ pair that corresponds to, respectively, $\mathsf{did\_click} = 1$ and $\mathsf{did\_app} = 1$. Figure 4.9 shows the resultant distributions from this analysis. In general CTR and $P(App \mid Click)$ tend to increase as a function of document relevance (r_i). Hence:

Assumption 1. *For job search, the item at rank position i in sequence $\mathcal{A}$ is relevant if and only if a job application action was observed at i:*

$$(\text{“}A\text{”}, i) \in \mathcal{A} \iff r_i = 1\,.$$

Inferring T. To estimate the user's anticipated number of relevant documents (that is, the user's target) T from interaction logs, the following assumption is made:

Assumption 2. *Users complete their search session at some point after they have met their expected volume of relevance, that is, at some point after T_i reaches* 0.

Moffat et al. [153] conducted a user study that, to some extent, supports Assumption 2. In connection with a laboratory-based search experiment, participants were asked about the expected number of useful web pages they would expect to see to complete a search task. Each answer to this question is an estimate of T. While not arguing that users only exit their search once $T_i = 0$, Moffat et al. [153] show that the stopping probabilities do tend to increase as T_i decreases.

With Assumptions 1 and 2, it can be inferred that T at the beginning of a session is equal to the number of job applications observed in the session. However, this creates a problem when T is inferred from sessions with no applications, since $T > 0$ would seem to be implicit in the fact that the user has commenced a search. To smooth this discontinuity, $T_\alpha > 0$ is introduced as a background expectation, the minimal target that triggers the user to perform a search task. In the absence of prior knowledge, $T_\alpha = 0.5$ is assumed, and all of the experiments reported in this chapter use this number.

It is also desirable to assign T to individual queries in the session. To allow that, this study distinguishes between the session-level T, denoted by T_0; and a per-query value T_j, representing the anticipated number of relevant documents still sought at the commencement of the j th query. Other quantities are also defined: $T_{j,i}$, representing the remaining anticipated volumes of relevance after the user inspected the i th item in the j th SERP for query Q_j, with $T_{j,0} \equiv T_j$ (useful as a notational convenience); and $T_{j,*}$, remaining anticipated relevant documents when the user exits from the j th SERP, with $T_{0,*} \equiv T_0$. Furthermore, while $T_{j,i}$ is allowed to be negative, it is not desirable for T_j to be zero or less. To start a new query, T_j should be at least T_α. Based on these considerations, it is proposed that $T_{j,i}$ be computed as follows:

$$T_{j,i} = \begin{cases} \max(T_{j-1,*}, T_\alpha) & i = 0 \\ T_{j,i-1} - r_{j,i} & i > 0\,. \end{cases} \tag{4.12}$$

Finally, if $napp(\mathcal{A})$ is the number of jobs applied for in one of the `Seek.com` action sequences associated with that session, then T_0 is estimated via

$$\hat{T}_0 = T_\alpha + \sum_{k=1}^{\infty} napp(\mathcal{A}_k)\,. \tag{4.13}$$

For the `Yandex.ru` data, $napp(\mathcal{A})$ is replaced by $nrc(\mathcal{A})$, the number of distinct relevant items clicked in the action sequence.

For example, consider a session $\mathcal{S}_1 = \langle Q_1, Q_2, Q_3 \rangle$, with three corresponding action sequences:

$$\begin{aligned}
\mathcal{A}_1 &= \langle (\text{“I”}, 1), (\text{“I”}, 2), (\text{“I”}, 4), (\text{“C”}, 4), (\text{“I”}, 2), (\text{“I”}, 3) \rangle \,, \\
\mathcal{A}_2 &= \langle (\text{“I”}, 1), (\text{“I”}, 2), (\text{“C”}, 2), (\text{“A”}, 2), (\text{“I”}, 3), (\text{“I”}, 5), (\text{“I”}, 6) \rangle \,, \\
\mathcal{A}_3 &= \langle (\text{“I”}, 1), (\text{“I”}, 3), (\text{“C”}, 3), (\text{“A”}, 3), (\text{“I”}, 4), (\text{“I”}, 7), (\text{“I”}, 5) \rangle \,.
\end{aligned} \tag{4.14}$$

In the first query the user examined the items at ranks 1, 2, and 4, then clicked at rank 4, then viewed the items at rank 2 (again) and rank 3. In the third query, after a second reformulation, the user viewed two items, clicked at rank 3 and started an application, and then examined three further items before completing their session. Table 4.5 shows the computation of various values, T_0, T_j, $T_{j,i}$, and $\mathbf{c}_i$, for this three-query example session.

According to rule G, one of three rules for determining the value of $\mathbf{c}_i$ (see Table 3.2 on page 82), there are three non-continuations observed at rank positions 3 and 4 in the sequence $\mathcal{A}_1$, since neither of them are followed by an action that took place in a higher rank positions. Complete assignments of $\mathbf{c}_i$ for all actions in the three action sequences are shown in the last column of Table 4.5. Note that click and job application actions are not considered for inferring $C(\cdot, \cdot)$, since both of them always follow impressions anyway.

Hypotheses Regarding Factors Affecting $C(\cdot, \cdot)$. Recall that Moffat et al. [153, 155] suggest that the conditional continuation probability:

1. increases with rank position i;
2. increases with T, the expected volume of relevance; and
3. decreases as relevant documents are accumulated (or as $T_{j,i}$ decreases).

Moffat et al. [153, 155] go on to define INST, and adaptive metric based on these hypothesised user behaviours. Next an experiment is carried out using commercial interaction logs to see evidence for or against the hypothesised relationships.

Analysis. A set of independent one-variable logistic regression analyses was conducted to obtain an overview of how these various factors, particularly rank i, T, and $T_{j,i}$ contribute to $C(\cdot, \cdot)$, using the continuation indicator $\mathbf{c}_i$ as a response variable. An additional binary factor “$i\%20 = 0$” was also added into the experiment, to quantify the effect of

	T_0	T_j	$T_{j,i}$	i	a_t	$\mathbf{c}_i$
$\mathcal{A}_1$, $j = 1$	2.5	2.5	2.5	1	"I"	1
	2.5	2.5	2.5	2	"I"	1
	2.5	2.5	2.5	4	"I"	0
	2.5	2.5	2.5	4	"C"	–
	2.5	2.5	2.5	2	"I"	1
	2.5	2.5	2.5	3	"I"	0
$\mathcal{A}_2$, $j = 2$	2.5	2.5	2.5	1	"I"	1
	2.5	2.5	2.5	2	"I"	1
	2.5	2.5	2.5	2	"C"	–
	2.5	2.5	1.5	2	"A"	–
	2.5	2.5	1.5	3	"I"	1
	2.5	2.5	1.5	5	"I"	1
	2.5	2.5	1.5	6	"I"	0
$\mathcal{A}_3$, $j = 3$	2.5	1.5	1.5	1	"I"	1
	2.5	1.5	1.5	3	"I"	1
	2.5	1.5	1.5	3	"C"	–
	2.5	1.5	0.5	3	"A"	–
	2.5	1.5	0.5	4	"I"	1
	2.5	1.5	0.5	7	"I"	0
	2.5	1.5	0.5	5	"I"	0

Table 4.5: Calculation of T_0, T_j, $T_{j,i}$, and a continuation indicator $\mathbf{c}_i$, for a session of three action sequences, and assuming $T_\alpha = 0.5$. Note that $\mathbf{c}_i$ is computed only for impression actions, since a click and/or a job application imply an impression.

pagination, an issue with the browser-based queries in the `Seek.com` logs. This analysis employs the subset of the `Seek.com` data that contains only organic results (without promoted items). Note that the two alternatives of inferring T, namely T_0 and T_j, were tested in two separate regression models, since T_0 and T_j are highly correlated ($r > 0.8$). Putting them together in a generalised linear model would generate unreliable estimates of their individual coefficients, since they largely explain the same variance.

Table 4.6 shows multiplicative effect sizes for i, T_0 or T_j, and $T_{j,i}$, aggregated over queries and all sessions. Intercepts are not meaningful in this context, and thus are not shown. The indicator "$i\%20 = 0$" was not included when modelling continuation probability on the continuous-scroll data. In these results the signs are of more interest than the magnitudes, and show that all of the rank position i, the user goal T (both as T_0 and as T_j), and $T_{j,i}$ are all positively correlated with $\mathbf{c}_i$, providing empirical corroboration for the relationships proposed by Moffat et al. [153, 155]. Pagination has a strong negative relationship with continuation behaviour, and in the browser-based `Seek.com` queries, users are more likely to end their inspection on page boundaries than at other ranks, with 64% decrease in the odds of continuing.

Factor	iOS/Android		browser		yandex	
	coef.	p	coef.	p	coef.	p
i	0.019	0.000	0.019	0.000	0.039	0.000
T_j	0.107	0.000	0.037	0.000	0.266	0.000
T_0	0.062	0.000	0.018	0.000	0.199	0.000
$i\%20 = 0$	–	–	-1.023	0.000	–	–
$T_{j,i}$	0.105	0.000	0.029	0.000	0.934	0.000

Table 4.6: Effect sizes for factors in a fitted model of a binary continuation indicator, $\mathbf{c}_i$, across all of the queries in the sessions, with each factor computed independently of all other factors.

Table 4.7 stratifies the effects based on each query's position in the session, covering the first three queries in each session. The same general patterns arise for the first three queries. It can also be seen that the effect of "$i\%20 = 0$" slightly decreasing with query position j, suggesting that each of the SERPs is treated broadly the same by the user inspecting them. Figure 4.10 visually illustrates the effect of the factors i and "$i\%20 = 0$" for the first three query positions that are described in Table 4.7.

To conclude this sub-section, this study has found an empirical support that conditional continuation probability (or query-level behaviour) has a positive correlation with all of the rank position i, the user target T, and the progress towards goal $T_{j,i}$. These results provide further evidence that INST provides a user model that is helpful when analysing query-level behaviour.

4.5.2 Session-Level Behaviours

According to the user model depicted in Figure 4.3 (page 126), the probabilistic nature of $C(\cdot,\cdot)$ allows users to end a query at any rank position. Once that happens, they have two choices: to end the entire session, or to continue by reformulating their query. The factors that drive that decision are also of interest, and this section focuses on modelling the user's propensity to reformulate to Q_{j+1}, given that they have exited from the j th SERP. This is the conditional reformulation probability, $F(j)$.

Several plausible explanatory factors are considered: the query's position in the session, j; the initial target, T_0; and the unmet volume of anticipated relevance upon exit from the j th SERP, $T_{j,*}$. Hypotheses in regard to these factors are as follows:

1. $F(j)$ increases with query count j, so that the more the user reformulates their queries, the more likely it is that they reformulate again in the future. Note that

Query	Factor	iOS/Android	browser	yandex
$j = 1$	i	0.02	0.02	0.03
	T_j	0.12	0.08	0.27
	T_0	0.12	0.08	0.27
	$i\%20$	–	-1.05	–
	$T_{j,i}$	0.14	0.07	1.06
$j = 2$	i	0.02	0.02	0.06
	T_j	0.09	0.09	0.26
	T_0	0.07	0.07	0.16
	$i\%20$	–	-1.03	–
	$T_{j,i}$	0.08	0.08	0.76
$j = 3$	i	0.02	0.02	0.08
	T_j	0.12	0.07	0.23
	T_0	0.08	0.06	0.12
	$i\%20$	–	-0.94	–
	$T_{j,i}$	0.11	0.07	0.51

Table 4.7: Effect sizes in a fitted model of the continuation indicator, $\mathbf{c}_i$, tabulated separately for first three queries in each sessions, and again computed as a sequence of independent regressions.

LCYsRBP does not comply with this hypothesis, because $F_{\textsf{LCYsRBP}}(j)$ is constant. In contrast, sDCG and KsDCG are compliant with this hypothesis.

2. $F(j)$ increases with T_0, so that the likelihood of reformulation increases as the total anticipated relevance increases. This hypothesis suggests that a user model should provide parameters to allow different numbers of reformulations; and sDCG, KsDCG, and LCYsRBP support this to some extent via their parameters.

3. $F(j)$ decreases as $T_{j,*}$ decreases, so that as the user accumulates answers toward their goal, they are more likely to end their search session. None of these session metrics, sDCG, KsDCG, and LCYsRBP, are compliant with this hypothesis.

Other explanatory variables are also possible, such as the proportion of relevant items clicked in the j th query, and the rate at which gain has been accumulated upon exit from j th SERP. Exploration of other factors, including the possibilities listed, will be undertaken as future work.

Analysis. An experiment is carried out to see the effect sizes of factors described in the previous numbered list in regard to $F(j)$. Logistic regression is again employed, now to model the query reformulation decision at the end of query j, denoted by $\mathbf{f}_j$, a dichotomous

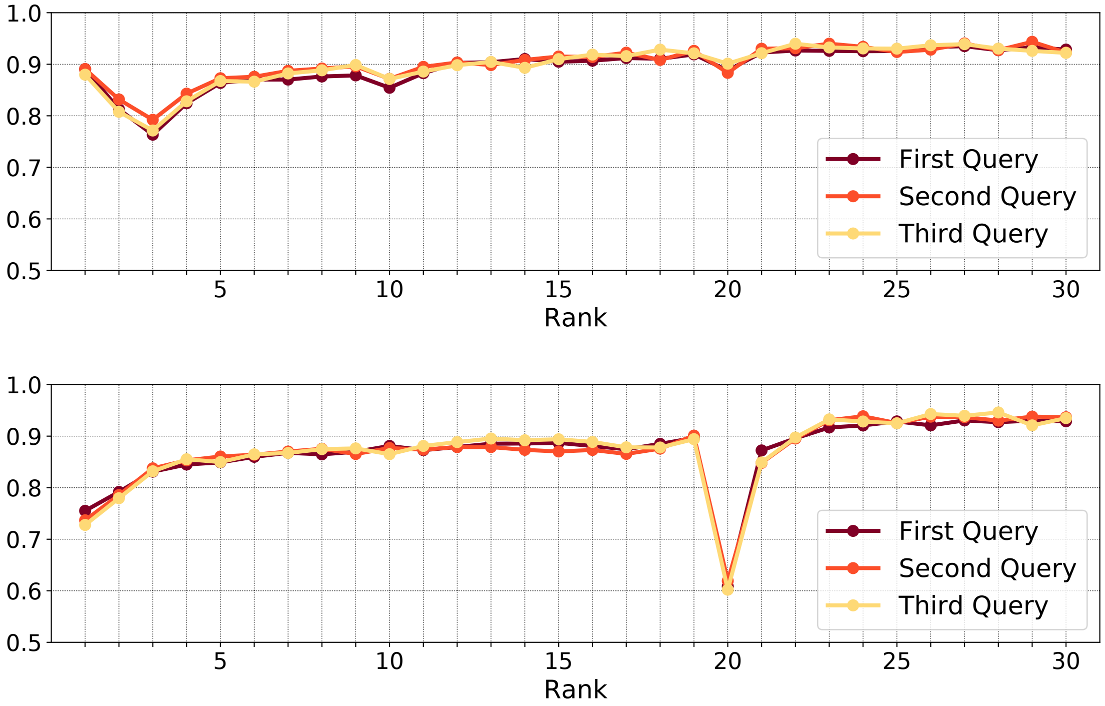


Figure 4.10: Empirical conditional continuation probabilities for ranks $i \leq 30$, for the first three queries in each session ($j \leq 3$), for `Seek.com` app-based (top) and browser-based (bottom) queries.

variable: whether to reformulate ($\mathbf{f}_j = 1$), or not ($\mathbf{f}_j = 0$). If a session with m queries was observed, $\mathbf{f}_j$ is determined as follows:

$$\mathbf{f}_j = \begin{cases} 1 & \text{if } j < m \\ 0 & \text{if } j = m\,. \end{cases}$$

The coefficients of explanatory variables are again trained independently via a set of logistic regression models, and the sign of the resultant coefficient is of interest.

Table 4.8 summarises the effects of the three factors, j, T_0, and $T_{j,*}$, for the prediction of $\mathbf{f}_j$. In the case of the `Yandex.ru` data, L1 regularisation is used, since a quasi-complete separation with respect to $T_{j,*}$ led to a very large coefficient value. In general, the three hypotheses regarding $F(j)$ are validated, with positive coefficients for all three factors. Figure 4.11 shows the positional distribution of last application (`Seek.com`, left) and last relevant click (`Yandex.ru`, right), as a function of query position in the session, stratified by the number queries in the session. These "success" actions, which provide *prima-facie* evidence of goal fulfillment, have a strong tendency to appear in the last query of the

Factor	iOS/Android		browser		yandex	
	coef.	p	coef.	p	coef.	p
j	0.140	0.000	0.079	0.000	0.333	0.000
T_0	0.162	0.000	0.148	0.000	0.141	0.000
$T_{j,*}$	6.826	0.000	3.999	0.000	22.912	0.000

Table 4.8: Independent-regression effect sizes and corresponding p values for factors in a fitted model for the binary reformulation indicator $\mathbf{f}_j$.

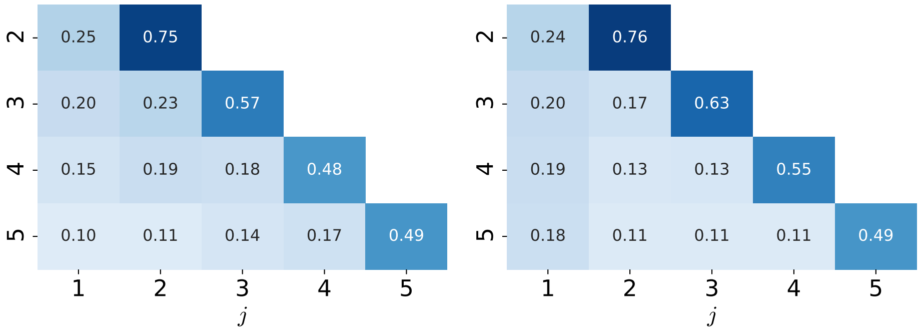


Figure 4.11: Positional distribution of last application (`Seek.com`, left) and last relevant click (`Yandex.ru`, right) as a function of query number in the session (x-axis), stratified by the number of queries in the session (y-axis). The values across each row sum to one. The `Seek.com` browser-based users have a similar trend.

session. This reinforces the finding that users have tendency to end the session as they accumulate relevant documents (that is, $T_{j,*}$ has a positive correlation with $\mathbf{f}_j$). Finally, Figure 4.12 plots empirical conditional reformulation probabilities, $\hat{F}(j)$, for $1 \le j \le 10$, computed as:

$$\hat{F}(j) = \frac{\sum_{\mathcal{S}} \mathbb{I}(Q_j \in \mathcal{S} \wedge \mathbf{f}_j = 1)}{\sum_{\mathcal{S}} \mathbb{I}(Q_j \in \mathcal{S})} ,$$

where $\mathbb{I}(P)$ is an indicator function that returns 1 if P holds, and 0 if otherwise. Note that the denominator decreases with j (compare with Figure 4.6 on page 136).

The increasing trend of $\hat{F}(j)$ is obvious, and matches the "sunk cost" within-query continuation behaviour illustrated in Figure 4.10 (page 152). This trend is also consistent with the fact that the factor j is positively correlated with $\mathbf{f}_j$, described in Table 4.8 (page 153). Note also how `Yandex.ru` Web search users are far less inclined to reformulate queries than are job search users observed from the `Seek.com` dataset.

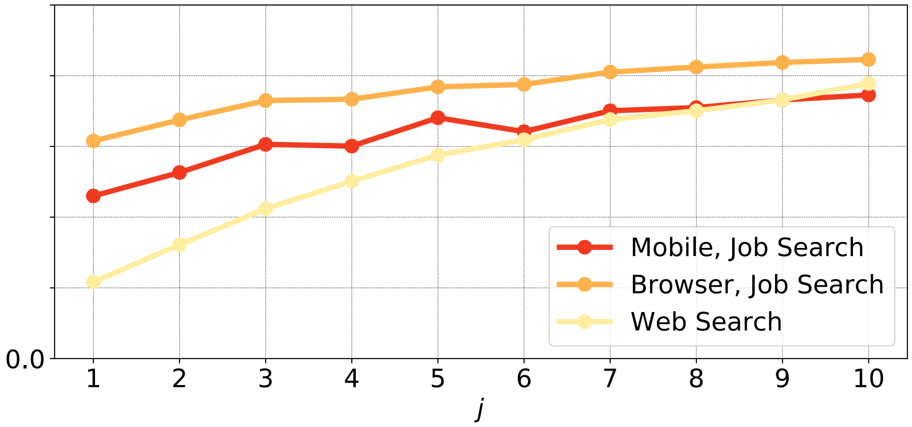


Figure 4.12: Conditional reformulation probabilities, $\hat{F}(j)$, for the first ten query positions, $1 \leq j \leq 10$. Recall that $\hat{F}(j)$ is the empirical probability of users reformulating the jth query. The vertical scale is linear; but for commercial-in-confidence reasons is not labeled.

4.6 A Model-Based Session Metric

Modelling $F(j)$. The previous section established factors explaining the variance of user reformulation behaviour, $F(j)$ (Table 4.8, Figure 4.12). This section now seeks to incorporate those behavioural patterns into a model for session evaluation. The model should depend only on relevance judgements, since the main goal is to devise an offline effectiveness metric based on session collection. Nevertheless, additional parameters are still allowed, so as to take into account user variability, including each user's initial search target T_0. Recall that query count j, session target T_0, and $T_{j,*}$ (remaining volume after the $j-1$th SERP is exited), are all positively correlated with $F(j)$. With these findings, and the inspiration from the formulation of INST [153, 155], captured as Equation 2.44 on page 55, the following "idealised" model for $F(j)$ is proposed:

$$F(j) = \left(\frac{j + T_0 + T_{j,*}}{j + T_0 + T_{j,*} + \kappa} \right)^2 . \tag{4.15}$$

Here $\kappa > 0$ is an constant that controls the rate at which the reformulation probability increases with j.

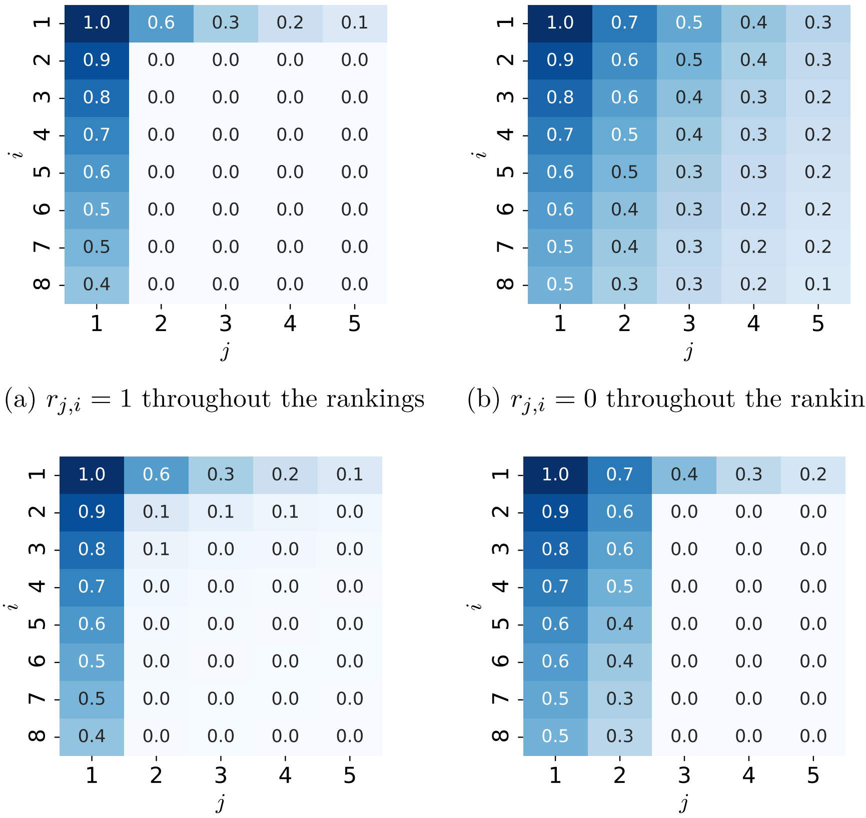
Session-Based INST. New session-based effectiveness metrics can be proposed by taking Equation 4.15 and by adding a $C(j, i)$ function for governing per-query conditional continuation probabilities. Again drawing on the proposal of INST (Equation 2.44 on page 55), a "session INST" (sINST) is specified via the following definition of $C(j,i)$:

$$C_{\mathsf{sINST}}(j,i) = \left(\frac{i + T_j + T_{j,i} - 1}{i + T_j + T_{j,i}}\right)^2 . \tag{4.16}$$

Although T_j depends on volumes of relevance that the user has encountered in the previous queries, $\langle Q_1, \ldots, Q_{j-1}\rangle$, the j th query is treated independently within the session, in the sense that the user is assumed to commence their inspection of the j th SERP according to the target of gaining T_j units of relevance. The computation of $W(j,i)$ involves an aggregation over all SERPs in the session (Equations 4.9 and 4.10), and thus the metric score calculated using $W(j,i)$ indicates the effectiveness of the entire session.

Recall from Equation 4.9 (page 140) that $W(j,i)$ is determined from $V(j,i)$, the fraction of the user population that commence the session and go through to view that document, and that $W(j,i)$ is a normalisation of $V(j,i)$. Figure 4.13 shows the distribution of $V(j,i)$ for sINST ($T = 8$ and $\kappa = 3$), categorised by four scenarios. As can be seen, $V(j,i)$, which is directly proportional to $W(j,i)$, alters its value depending on what has been inspected by users. When the SERP for the first query is full of relevant answers (Figures 4.13a and 4.13c), users make good progress towards their expected target T_0 in that SERP, and are less motivated to reformulate the query, since the first SERP provides the information they need ($T_{1,*} \approx 0$ and hence T_2 is very small). However, when the opposite happens (Figures 4.13b and 4.13d), users inspect the first SERP; and when they abandon the first SERP, they have motivation to submit follow-up queries and to view further SERPs, since $T_{1,*} \approx T_0$ and hence $T_2 \approx T_0$. These definitions make session INST an adaptive metric, since the user behaviour (operationalised by $C(j,i)$ and $F(j)$) varies as a function of total volume of relevance accumulated by the user.

With these specifications, T_j and $T_{j,i}$ adapt as the user examines items in a SERP and as they reformulate their queries. From the perspective of a single user, $T_{j,i}$ is completely defined by Equation 4.12 on page 147. However, the probabilistic nature of the user model proposed in this chapter (Figure 4.3) emerges because a metric represents a population of users. As a consequence, the last rank position inspected in each SERP is a random variable that has a distribution, implying that $T_{j,*}$ also has a distribution that should be allowed for when $T_{j+1,0}$ is being computed (Equation 4.12). That is, users making up the population can have different T_j values at the beginning of the same j th query. Hence, even though the metric score might be deterministic in principle, it seems difficult to compute its value, a case which means that a Monte Carlo simulation can serve as a solution. Note that this is not the issue for query-based C/W/L metrics, such as INST, since the browsing



(a) $r_{j,i} = 1$ throughout the rankings

(b) $r_{j,i} = 0$ throughout the rankings

(c) $r_{1,i} = 1$, but $r_{j,i} = 0$ for $j > 1$

(d) $r_{1,i} = 0$, but $r_{j,i} = 1$ for $j > 1$

Figure 4.13: Distribution of $V(j,i)$ (the proportion of users that examine the i th result in the SERP associated with the j th query) for sINST ($T = 8$ and $\kappa = 3$), for four scenarios.

path within a single SERP is simply linear.

A Monte Carlo approach requires a large number of randomised trials so as to yield an accurate approximation. To avoid the expense of such computations, we propose that in practice the evolution of T_j be computed using expectations:

$$T_j = \max(T_{j-1} - M_{ETG}(\mathrm{SERP}_{j-1}),\, T_\alpha)\,, \tag{4.17}$$

where $M_{ETG}(\mathrm{SERP}_{j-1})$ is the expected total gain computed from the $j-1$ th SERP when a probabilistic user modelled by INST inspects that SERP with the expectation of needing T_{j-1} relevant items. That is, the proposal is that $T_{j,*}$, essentially a random variable, be condensed to a representative “point” value in order to compute both T_j (Equation 4.12 on page 147) and $F(j)$ (Equation 4.15 on page 154). This also allows for the same computation for the next SERP ($j+1$ th query) using a single value of T_{j+1}.

Meta-Evaluation Via Held-Out Data. Section 3.3 (page 87) described a method to measure the accuracy of continuation probabilities through the lens of the C/W/L framework, making use of the *weighted mean squared error* to compute the fit between the function $C(\cdot)$ associated with a query-based metric and its empirical value, $\hat{C}(\cdot)$, observed via logged behaviours. The weights on the mean squared error, which correspond to the relative frequency of the i th item being examined, are required since $C(\cdot)$ is not a probability distribution that sums to one. The next experiment extends this approach to evaluate the accuracy of $C(j,i)$ and $F(j)$ in the context of session-based user models. Note that optimal values of both $C(j,i)$ and $F(j)$ are determined with respect to a single error function, since they might have shared parameters:

$$\mathsf{WMSE}(\omega) = \sum_j \sum_i w_c(j,i) \cdot \left(C(j,i;\omega) - \hat{C}(j,i)\right)^2 + \sum_j w_f(j) \cdot \left(F(j;\omega) - \hat{F}(j)\right)^2 ,$$

where ω is a set of parameters of a particular user model; $w_c(j,i)$ is the relative frequency of the item at rank i being viewed in the j th query; and $w_f(j)$ is the weight associated with the fraction of sessions that contains at least j queries.

Held-out datasets containing sessions initiated by 1,000 users for both browser- and mobile-based `Seek.com` data, and 100,000 search sessions for `Yandex.ru` data, were employed to compute the $\mathsf{WMSE}(\omega)$ function. Table 4.9 shows the parameter combinations providing the best fit for four session-based metrics (that is, minimising the $\mathsf{WMSE}(\omega)$ function), across the first five queries in each session, and the first 50 results in each SERP. Further, the metric best-fit parameters are found using grid search method with the following space of search:

- for sDCG and KsDCG: $bq \in \{1.5, 2.0, 2.5, \cdots, 10\}$, $b \in \{1.5, 2.0, 2.5, \cdots, 10\}$, $M \in \{1, 2, 3, \cdots, 10\}$, and $N \in \{1, 2, 3, \cdots, 60\}$;
- for LCYsRBP: $q \in \{0.01, 0.02, 0.03, \cdots, 1.0\}$ and $p \in \{0.01, 0.02, 0.03, \cdots, 1.0\}$;
- for sINST: $T \in \{1, 1.5, 2.0, \cdots, 5.0\}$ and $\kappa \in \{1, 1.5, 2.0, \cdots, 5.0\}$.

As can be seen, among the three session-based user models, sINST provides the closest fit with the empirical observations, once suitable parameters have been identified.

We have demonstrated that sINST has a more accurate user model than sDCG, KsDCG, and LCYsRBP. However, it remains unclear whether scores generated by sINST are also well correlated with user satisfaction. Hence, one clear direction is to investigate this correlation using a session test collection, together with a non-trivial number of participants in a lab-based environment. Three pre-existing lab-based datasets (J&A, THUIR2, and THUIR3)

User Model	Best-fit ω	$\mathsf{WMSE}(\omega)$
`Seek.com`, mobile app-based		
sDCG	$bq, b, M, N = 2.0, 10.0, 6, 51$	4.63
KsDCG	$bq, b, M, N = 1.5, 3.0, 6, 51$	3.75
LCYsRBP	$q, p = 0.93, 0.94$	0.40
sINST	$T, \kappa = 4.5, 4.0$	0.19
`Seek.com`, browser-based		
sDCG	$bq, b, M, N = 3.0, 4.5, 6, 51$	2.76
KsDCG	$bq, b, M, N = 2.0, 2.5, 6, 51$	2.12
LCYsRBP	$q, p = 0.88, 0.95$	0.59
sINST	$T, \kappa = 4.0, 2.0$	0.28
`Yandex.ru`		
sDCG	$bq, b, M, N = 1.5, 2.5, 2, 10$	7.56
KsDCG	$bq, b, M, N = 1.5, 2.0, 1, 10$	8.96
LCYsRBP	$q, p = 0.88, 0.74$	0.78
sINST	$T, \kappa = 2.0, 4.5$	0.43

Table 4.9: Best-fit parameters, found by minimising $\mathsf{WMSE}(\omega)$ ($\times 10^{-2}$) for three session-based user models, across the first 5 queries in each session and 50 results in each query. Note that small numbers indicate better fit to the observed data.

that contain satisfaction ratings cannot be used for this investigation, since these datasets were not constructed based on *static* sequences of queries per topic. These datasets are (only) useful for modelling session satisfaction when sequences of queries are observed. This issue will be discussed in the next section (Section 4.7).

Computation of sINST. Figure 4.14 (page 159) compares the sINST computed by the "expectation" method (Equation 4.17) and the sINST scores computed via a Monte Carlo simulation over 100,000 random "users". The fact that the correlation between two sets of scores in each dataset is very high ($r \approx 1.0$) provides a clear support for the proposed sINST computation using the more efficent "expectation" method.

4.7 Factors Affecting Session Satisfaction

Sections 4.4, 4.5, 4.6 have addressed the first goal of this chapter, the development of a session-based user model for an adaptive offline session metric. This section and the rest of this chapter discuss the second goal (see **RQ 4.2** on page 127), an investigation of the fitted relationship between session satisfaction ratings and individual query satisfaction ratings (or query scores). The insights gained from this study will then be incorporated

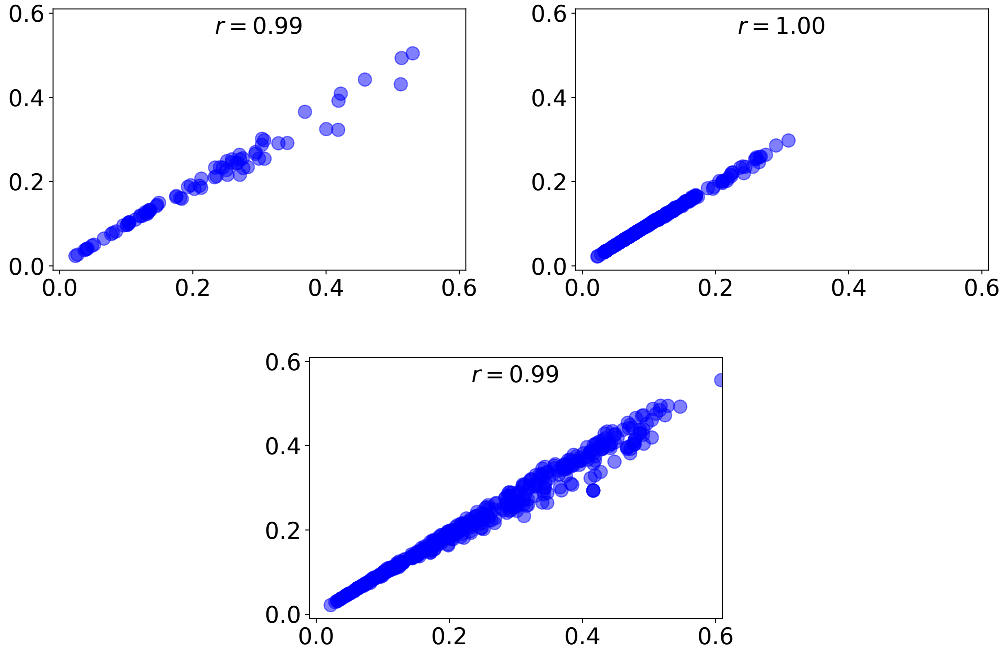


Figure 4.14: Monte Carlo simulation method (y-axis) versus "expectation" method (Equation 4.17, x-axis) for computing sINST score (ERG version, $T = 8$ and $\kappa = 3$). The two plots show 80 sessions from the J&A dataset (top-left), 223 sessions from the THUIR2 dataset [145] (top-right), and 450 sessions from the THUIR3 dataset [139] (bottom).

into the development of novel session satisfaction models (Section 4.8), which are useful for scoring sessions for which the query sequence is known. Here the THUIR3 dataset is used for model exploration; and THUIR2 and J&A are employed as held-out datasets only for model validation. These three datasets contain session-level satisfaction ratings.

Last Query vs. Best Query. Analysis of the THUIR3 data led by Liu et al. [137, 139] suggests that the last query is the most important factor for modelling session-level user satisfaction. However, as shown in Figure 4.15, the last query is also likely to be the best query in the session. This sparks a critical question: *which is more important, the last query or the best query?* To explore this issue, a linear regression model is employed to observe the influence of a range of positional and quality factors, including the *first*, *last*, *best*, and *worst* query, seeking to predict the corresponding session-level satisfaction rating. Query-level satisfaction ratings were taken as representing the individual query scores, since they provide a ground-truth that reflects what the user experienced when interacting with the SERPs. Sessions with one query were not considered, since the correlation between query- and session-level ratings is very strong with $r = 0.93$, suggesting that an aggregation task for sessions with one query ($|\mathcal{S}| = 1$) is more trivial than an aggregation task for those

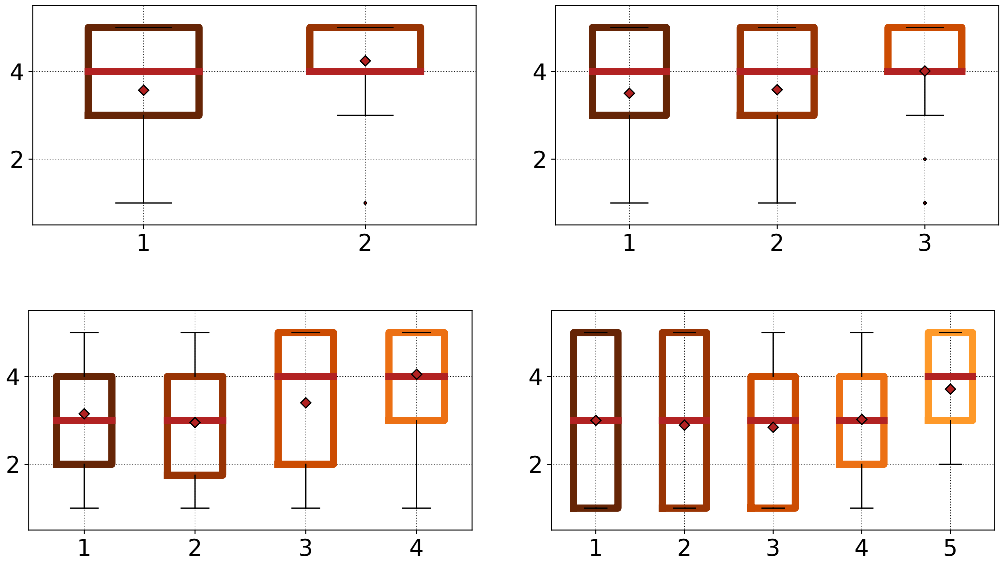


Figure 4.15: Query-level satisfaction ratings (y-axis) across positions in the session (x-axis) for $|\mathcal{S}| \in \{2, 3, 4, 5\}$ in the THUIR3 dataset. The diamond in each bar is the mean.

with at least two queries ($|\mathcal{S}| \geq 2$). This filtering leaves 360 sessions with $|\mathcal{S}| \geq 2$.

Table 4.10 describes the regression analysis result, categorised into three cases: using all 360 sessions; using sessions where the last query is the best one (239 sessions); and using sessions where the last query is not the best one (121 sessions). In general four factors are significant: the best, last, and second best queries, which have strong positive relationships with session ratings, together with the number of queries in the session, which is slightly negatively correlated with session-level satisfaction; and that the best query seems to be more meaningful than the last one, with the difference most notable in the cases where the last query is not the best one.

When those four significant features (*best*, *last*, *secondbest*, and $|\mathcal{S}|$) are used to build the full model, the Akaike information criterion (AIC) is 665.4. Removing the *best* factor leads to $\mathsf{AIC} = 711.6$ ($\Delta_{\mathsf{AIC}} = 46.2$), while filtering out the *last* factor yields $\mathsf{AIC} = 707.3$ ($\Delta_{\mathsf{AIC}} = 41.9$). This AIC-based analysis provides further evidence that the best query is more valuable than the last one. In addition, analysis of Pearson's r correlation coefficients between the predicted final scores based on various linear combinations of these features (as suggested by the coefficients in Table 4.10) and overall user-generated session satisfaction ratings was carried out. As can be seen in Table 4.11, a linear model built upon the *best*-only factor gives a slightly better performance than one based on only the *last* factor.

Factor	All		Last = Best		Last ≠ Best	
	coef.	p	coef.	p	coef.	p
intercept	0.70	0.00	0.85	0.00	0.62	0.21
$\lvert\mathcal{S}\rvert$	−0.09	0.00	−0.15	0.00	−0.02	0.65
first	−0.02	0.69	0.07	0.23	−0.04	0.56
second	0.01	0.88	0.08	0.14	−0.06	0.34
prevlast	−0.04	0.40	0.10	0.14	−0.08	0.24
last	0.23	0.00	0.35	0.00	0.10	0.28
best	0.42	0.00	0.35	0.00	0.34	0.02
secondbest	0.20	0.00	0.08	0.30	0.31	0.01
secondworst	0.10	0.05	−0.01	0.89	0.20	0.05
worst	−0.02	0.71	−0.14	0.06	0.09	0.39

Table 4.10: Effect sizes and p values for positional and quality factors in a fitted linear regression model for session-level satisfaction ratings in THUIR3. Sessions with only one query were not considered. Low p values (< 0.05) indicates that the factor is meaningful for the model.

Combining Positional and Quality Information. Past aggregation function proposals mostly presuppose that $\theta(j)$ varies as a function of (only) query position in the sequence [104, 137, 139, 242]. However, as is demonstrated in Table 4.11, a simple linear model including both *last* and *best* factors performs significantly better than using either of them individually (two-sided Hotelling's t test with $p < 0.05$). This finding is also in agreement with a well-known cognitive bias, called *peak-end rule*, in which people tend to remember "the best" and "the final" moments of an experience, suggesting that those two critical moments should receive more attention than the others [113]. The best correlation coefficient in Table 4.11 was achieved when all four significant factors (Table 4.10) were employed, including $|\mathcal{S}|$, the number of queries in the session. These outcomes provide further evidence that combining positional and quality information might yield session scores that better capture the user's experience with the entire session.

To reinforce that finding, a second AIC-based analysis was carried out, to see whether a joint model using positional and quality information is better than using only one of them individually. For a session with $|\mathcal{S}|$ queries, there are $2 \cdot |\mathcal{S}|$ features in total, one set based on positions and one set based on quality. For example, a session with three queries is associated with three positional features, the *first*, *second*, and *third* queries; and three quality features, the *best*, *second-best*, and *third-best* queries. Table 4.12 shows the results of this analysis. Here AIC scores are computed using fitted regression models with session satisfaction rating being the response variable, for sessions with two, three, four, and

Linear Model	r	p
$0.23 \cdot \mathit{last}$	0.66	–
$0.42 \cdot \mathit{best}$	0.67	0.75
$0.42 \cdot \mathit{best} + 0.23 \cdot \mathit{last}$	0.74	0.00
$0.42 \cdot \mathit{best} + 0.23 \cdot \mathit{last} + 0.20 \cdot \mathit{sndbest}$	0.74	0.83
$0.42 \cdot \mathit{best} + 0.23 \cdot \mathit{last} + 0.20 \cdot \mathit{sndbest} - 0.09 \cdot \lvert\mathcal{S}\rvert$	0.78	0.00

Table 4.11: Correlation coefficients (Pearson's r) between session satisfaction ratings and session scores as computed via five linear models based on the four most significant factors identified by the THUIR3 regressions in Table 4.10. The p values relate to the difference between each row and its predecessor, computed using Hotelling's t test for comparing two Pearson coefficients with overlapping variables [90]. Only sessions with at least two queries are included.

# Queries	# Sessions	Pos.	Qual.	Pos. + Qual.
2	79	128.5	136.9	127.3
3	110	206.3	215.3	201.7
4	60	103.1	106.7	96.1
5	45	97.4	89.9	96.6

Table 4.12: AIC scores for a joint positional-quality model and two individual models when predicting session-level user satisfaction ratings, using THUIR3. Lower numbers are better.

five queries independently. (Sessions with more than five queries were not considered.) In general the joint positional and quality-based model performs better than either of the individual models. Here the individual positional-based model is generally better than the individual quality-based model. However, this does not contradict the previous finding, suggesting that the best factor is more important than the last factor. Note that the term "positional" does not correspond only to a single "last" one, but refers to a linear combination of all queries in a session, a spectrum from the first to the last one. Similarly, the term "quality" refers to a combination of all queries, from the best to the worst one.

In the next experiment, the whole THUIR3 dataset (360 sessions with $|\mathcal{S}| \geq 2$) was used to find a set of weights with which each sessions' query satisfaction ratings are aggregated into a session score, allowing both position and quality factors to exert influence. Each session length $|\mathcal{S}|$ is associated with a different weight vector, with the weights summing to one for each session length. For a quality-based vector, the weights correspond to the spectrum from the best to the worst queries. The maximum session length in THUIR3 is $|\mathcal{S}| = 12$, and thus there are 12 vectors and $1 + 2 + \cdots + 12 = 78$ parameters (weights)

	#Param.	Pearson r
Positional	78	0.830
Quality	78	0.818
Pos. + Qual.	156	0.854

Table 4.13: Best correlation coefficients between predicted session scores and observed session ratings using optimal query weights for three different models, for THUIR3. Higher numbers are better.

in total to be used for models based on quality or position alone. In the case where both quality and position are employed to influence the predicted session score, each session length $|\mathcal{S}|$ corresponds to a weight vector of length $2 \cdot |\mathcal{S}|$ and there are 156 trainable parameters in total.

Table 4.13 shows the Pearson's correlation coefficients arising from this arrangement, calculating the correlation between predicted session scores and session satisfaction ratings. The joint positional- and quality-based optimisation gives a better outcome than an optimisation based on an individual model. Figure 4.16 shows the best-fit weights that were generated for $|\mathcal{S}| \leq 5$, and confirms that the last and the best queries are the two most important factors in both the two individual models (top row) and the combined model (bottom row). The patterns of weight values in the top-left and bottom-left heatmaps confirm the findings of Liu et al. [139], that recency has a strong influence, and hence that an increasing weight function is appropriate. When quality-based influence is allowed for, a decreasing function over the quality spectrum from the best to the worst queries better fits the observations (top-right, bottom-right).

4.8 Modelling Session Satisfaction

Recall that, starting from Section 4.7, this chapter has explored the connection between session satisfaction ratings and individual query satisfaction ratings (or query scores). Further, Section 4.7 has explored several factors influencing session satisfaction, and has found that session satisfaction ratings are more accurately predicted when positional- and quality-based factors are both combined than when an individual model is used.

This section makes use of findings from Section 4.7 to develop two session satisfaction models, which are useful for scoring a session when user observation data is available. The first model is based on a weighted mean approach, where the positional- and quality-based factors are merged using a combination method with the weights summing to one. The

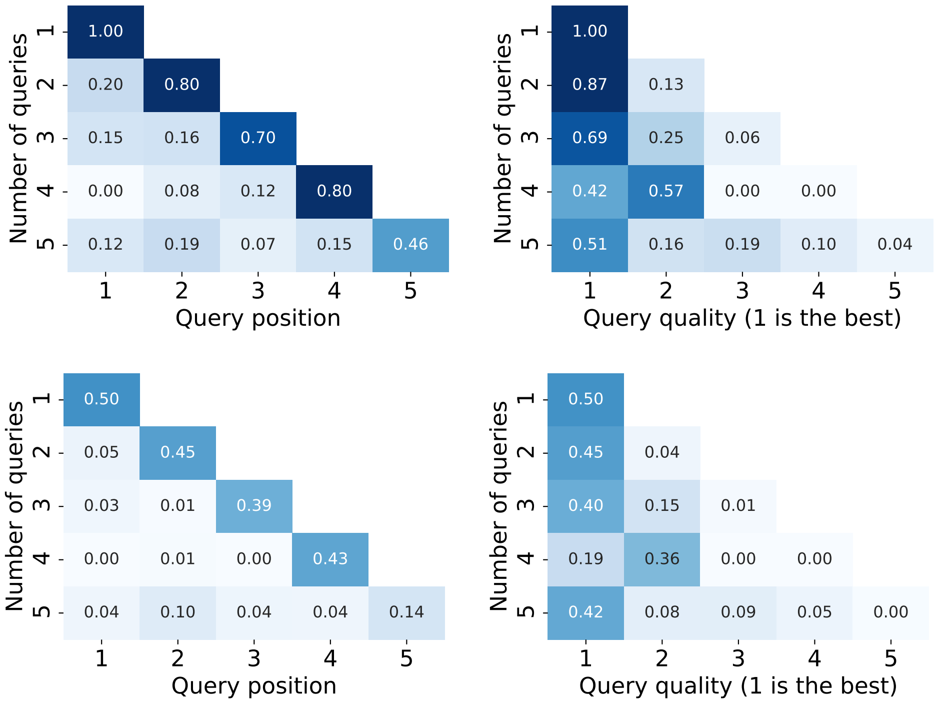


Figure 4.16: Query weights, stratified by the number of queries in the session, for sessions of length $|\mathcal{S}| \in \{1, 2, 3, 4, 5\}$. There are three cases: an optimisation based on query position in the sequence (top-left); based on query-level satisfaction (top-right); and on a joint positional- and quality-based optimisation (bottom pair). These three cases correspond respectively to the three rows in Table 4.13.

second model relies on the notion of forgetfulness [242], and is based on the rationale that the user to some extent forgets bad early queries, but still remembers the good ones. Results comparing our proposed models and baseline models are also reported.

4.8.1 Query Aggregation Using Weighted Mean Method

Recall from Equation 4.3 (page 132) that establishing a fitted relationship between individual queries in the session and session-level satisfaction can be done via a linear combination over individual query scores. With this definition, each SERP (or query) in the session has an association with a weight $0 \leq \theta(j) \leq 1$, where $1 \leq j \leq |\mathcal{S}|$.

Existing definitions of $\theta(j)$ are only affected by query positions in the session [139, 242]. The analysis in Section 4.7 suggests that combinations of positional- and quality-based factors yields better models compared to positional-only approaches. Hence, we propose the following generalisation:

$$\theta(j) = (1 - \gamma) \cdot \theta_{\mathsf{pos}}(j) + \gamma \cdot \theta_{\mathsf{qty}}(j)\,, \tag{4.18}$$

where $\theta_{\mathsf{pos}}(j)$ and $\theta_{\mathsf{qty}}(j)$ represent the positional- and quality-based contributions, and $0 \leq \gamma \leq 1$ is a balance parameter, with $\gamma > 0.5$ preferring the quality-based factor, and $\gamma < 0.5$ preferring the positional-based counterpart. Figure 4.16 confirms that positional-based weighting regimes that emphasize recent queries, such as $\theta_{\mathsf{Liu}}(j)$ [139], $\theta_{\mathsf{RSDCG}}(j)$ [242], and $\theta_{\mathsf{RSRBP}}(j)$ [242] should be employed for $\theta_{\mathsf{pos}}(j)$. To emphasize queries with good scores, the following formulation is introduced:

$$\theta_{\mathsf{qty}}(j) = (M(\overrightarrow{r_j})^{\mu}) \,/\, \left(\sum_{k=1}^{|\mathcal{S}|} M(\overrightarrow{r_k})^{\mu}\right)\,, \tag{4.19}$$

where μ determines the extent to which the user emphasizes good queries, with $\mu > 0$ favouring good queries, and $\mu < 0$ preferring low-scoring queries. The normalising denominator is needed since some query-based effectiveness measures, such as DCG, allow the scores to be greater than one. This specification now includes (at least) two parameters that must be tuned.

Baseline Approaches. In addition to the proposals of Liu et al. [139] and Zhang et al. [242], two baseline aggregation functions are also used. The first function is a “mean” function, $\theta(j) = 1/|\mathcal{S}|$. Jiang and Allan [104] show that the mean function performs better than other simple methods, such as “sum”, “max”, and “min”. While arguing that the last query is valuable, Liu et al. [139] also suggest that middle queries have less contribution to the session satisfaction. Based on their findings, we propose a second baseline function, an asymmetric *U-shape* weighting function, which is defined as follows:

$$\theta_{\mathsf{U}}(j) = \frac{(j - |\mathcal{S}|/2)^2 + 1}{\sum_{k=1}^{|\mathcal{S}|}(k - |\mathcal{S}|/2)^2 + 1}\,. \tag{4.20}$$

This function assumes that session middle queries are relatively less important than early and late queries, and that late queries are slightly preferred to early queries.

Tuning and Evaluation. Parameters for these various weighting schemes were developed using *sequential least squares programming* [125], optimising Pearson’s correlation coefficients between user-generated session satisfaction ratings and session scores computed using the aggregation functions. Two query-level score options were used: five-level *query satisfaction scores* assessed by users, mapped to $\langle 0.00, 0.25, 0.50, 0.75, 1.00 \rangle$; and *query effectiveness scores* computed using RBP (parameter $\phi = 0.8$), and derived from the SERPs and relevance judgements using the gain mapping function $g(r_i) = (2^{r_i} - 1)/(2^{r_{max}} - 1)$. Rank-biased precision ($\phi = 0.8$) was chosen because it has reasonable correlation with query-level satisfaction ratings compared to other offline effectiveness metrics, ERR and DCG, according to the recent study conducted by Liu et al. [139] (but still relatively low, at Pearson's $r \approx 0.38$).

Table 4.14 (page 167) shows the resultant coefficients from this experiment. Red coefficients are the results from "self-tuned" arrangement (upper-bound correlation coefficient), whereas blue values indicates non-self-tuned column maxima. Several patterns are observed:

- When the tuning involves aggregation methods that employ query satisfaction scores (the benchmark for query effectiveness scores), weighted mean models outperform other aggregation models;
- When query scores are computed via query effectiveness scores (RBP) in the tuning process, all composition functions yield low correlation coefficients (Pearson's $r \leq 0.45$), regardless of tuning scenario, but the weighted mean methods are still better than previous aggregation approaches; and
- When the tuning makes use of query effectiveness scores, the weighted mean methods are generally better than previous approaches when combining query effectiveness scores, but are worse than those when combining query satisfaction scores.

In the latter case, note that the correlation between query effectiveness scores and query satisfaction scores is low (Pearson's $r \approx 0.38$) [139].

Amongst the self-tuned arrangements, the weighted mean model with $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{Liu}}(j)$ gives the highest correlation coefficients among all aggregation models on three different datasets. Hotteling's t test [90], a statistical tool for comparing correlation coefficients with overlapping variables, was then employed to see whether this improvement is significant. The test result suggests that the weighted mean model with $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{Liu}}(j)$ significantly outperforms baseline approaches on both THUIR2 and THUIR3, but not on the J&A dataset. Table 4.15 shows the p values of this test for three datasets.

Experiment results using two recently proposed aggregation methods, RSDCG and RSRBP [242], are also reported in Table 4.16 (page 169). Self-tuned results show that the

Aggregation method	THUIR3		THUIR2		J&A
	QSat	RBP	QSat	RBP	RBP
Baseline approaches					
Mean, $\theta(j) = 1/\lvert\mathcal{S}\rvert$	0.72	0.27	0.71	0.36	0.41
Asymmetric U-shaped, $\theta_{\mathsf{U}}(j)$	0.78	0.28	0.71	0.34	0.42
Tuned on THUIR3 (QSat)					
Liu [139], with $\lambda = 0.40$	0.78	0.25	0.67	0.32	0.41
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{Liu}}(j)$, $\lambda = 0.00$, $\gamma = 0.79$, $\mu = 1.27$	0.83	0.26	0.76	0.29	0.41
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{U}}(j)$, $\gamma = 0.69$, $\mu = 1.72$	0.82	0.25	0.76	0.28	0.39
Tuned on THUIR3 (RBP $\phi = 0.8$)					
Liu [139], with $\lambda = 1.00$	0.72	0.27	0.71	0.36	0.41
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{Liu}}(j)$, $\lambda = 0.72$, $\gamma = 1.00$, $\mu = -2.07$	0.53	0.35	0.51	0.42	0.40
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{U}}(j)$, $\gamma = 1.00$, $\mu = -2.07$	0.52	0.35	0.51	0.42	0.40
Tuned on THUIR2 (QSat)					
Liu [139], with $\lambda = 0.94$	0.73	0.27	0.71	0.36	0.41
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{Liu}}(j)$, $\lambda = 0.53$, $\gamma = 1.00$, $\mu = 0.95$	0.81	0.27	0.77	0.28	0.37
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{U}}(j)$, $\gamma = 0.86$, $\mu = 1.08$	0.81	0.27	0.77	0.28	0.38
Tuned on THUIR2 (RBP $\phi = 0.8$)					
Liu [139], with $\lambda = 0.98$	0.73	0.27	0.71	0.36	0.41
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{Liu}}(j)$, $\lambda = 1.00$, $\gamma = 1.00$, $\mu = -5.00$	0.50	0.35	0.48	0.43	0.40
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{U}}(j)$, $\gamma = 1.00$, $\mu = -5.00$	0.50	0.35	0.48	0.43	0.40
Tuned on J&A (RBP $\phi = 0.8$)					
Liu [139], with $\lambda = 0.69$	0.76	0.26	0.70	0.35	0.42
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{Liu}}(j)$, $\lambda = 0.00$, $\gamma = 0.44$, $\mu = -1.00$	0.74	0.30	0.66	0.37	0.45
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{U}}(j)$, $\gamma = 0.31$, $\mu = -5.00$	0.70	0.32	0.65	0.39	0.43

Table 4.14: Correlation between session satisfaction ratings and computed session scores using either query satisfaction (QSat) or query scores (RBP), for a range of score aggregation options, and with tuning based on a variety of resources. Values in red are *self-optimised*, with parameter tuning and selection based on the reported quantity.

Other method	Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{Liu}}(j)$		
	THUIR3 (QSat)	THUIR2 (QSat)	J&A (RBP)
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{U}}(j)$	**0.012**	0.606	0.538
Liu [139]	**0.000**	**0.001**	0.302
Asymmetric U-shaped, $\theta_{\mathsf{U}}(j)$	**0.000**	**0.003**	0.397
Mean, $\theta(j) = 1/\lvert\mathcal{S}\rvert$	**0.000**	**0.001**	0.377

Table 4.15: The p values of two-sided Hotteling's t test [90] for comparing Pearson's correlation coefficients between the weighted mean model with $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{Liu}}(j)$ and any of other four aggregation models, computed upon self-tuned arrangements on THUIR3 (only QSat), THUIR2 (only QSat), and J&A (RBP). A significance level of 0.05 is used to test the null hypothesis that the correlation coefficients between any two models are not different.

weighted mean approach has higher correlation for both of the recency-aware aggregation methods. However, none of these improvements are significant by Hotteling's t test [90].

4.8.2 Memory-Based Query Aggregation

Memory-Based Aggregation Framework. Section 4.8.1 proposed a method for combining query scores using a weighted mean approach. Although this query-to-session aggregation method empirically performs better than the previous approaches, it lacks connection to an obvious user model. Zhang et al. [242] recently suggest that users, to some extent, forget the utility derived from early queries. Inspired by their work, we propose a memory-based aggregation framework that generalises two recent methods by Zhang et al. [242] and Liu et al. [139]. This framework consists of two key quantities:

1. $\sigma(j,k)$, the memory of the jth query upon exit from the kth SERP, with $j \le k$, $0 \le \sigma(j,k) \le 1$.

2. $\beta(j,k)$, the instantaneous *forget factor* for the jth query, given that the user has just started inspecting the kth SERP, with $j < k$ and $0 \le \beta(j,k) \le 1$.

Suppose the user inspects each SERP in turn, always starting at the first document of each. With this model, $\sigma(j,k) = 1$ denotes that the user still fully remembers the utility derived from the jth SERP upon exit from the kth SERP, while $\sigma(j,k) = 0$ represents that the user has completely forgotten the jth SERP. Other assumptions regarding $\sigma(j,k)$ are that the forgotten memory will never be recovered (that is, $\sigma(j,k+1) \le \sigma(j,k)$), and that the user will have a fresh memory of what they have just inspected (that is, $\sigma(j,j) = 1$).

Aggregation method	THUIR3	THUIR2	J&A
Tuned on THUIR3			
RSDCG $bq = 1.42, b = 2.34, \delta = 0.46$	0.34	0.39	0.37
RSRBP $p = 0.90, q = 0.89, \delta = 0.44$	0.35	0.40	0.34
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{RSDCG}}(j)$, $M(.) = M_{\mathsf{RSDCG}}(.)$ $bq = 1.10, b = 1.86, \delta = 0.28, \gamma = 0.59, \mu = -2.03$	0.36	0.40	0.39
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{RSRBP}}(j)$, $M(.) = M_{\mathsf{RSRBP}}(.)$ $p = 0.85, q = 0.88, \delta = 0.37, \gamma = 0.55, \mu = -2.12$	0.38	0.42	0.39
Tuned on THUIR2			
RSDCG $bq = 1.83, b = 2.86, \delta = 1.15$	0.31	0.41	0.38
RSRBP $p = 0.95, q = 0.79, \delta = 3.70$	0.33	0.44	0.38
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{RSDCG}}(j)$, $M(.) = M_{\mathsf{RSDCG}}(.)$ $bq = 2.33, b = 5.00, \delta = 5.00, \gamma = 0.62, \mu = -4.62$	0.33	0.44	0.41
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{RSRBP}}(j)$, $M(.) = M_{\mathsf{RSRBP}}(.)$ $p = 0.97, q = 0.87, \delta = 5.00, \gamma = 0.52, \mu = -3.80$	0.36	0.46	0.41
Tuned on J&A			
RSDCG $bq = 5.00, b = 3.01, \delta = 1.34$	0.32	0.38	0.41
RSRBP $p = 0.99, q = 0.75, \delta = 5.00$	0.28	0.35	0.44
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{RSDCG}}(j)$, $M(.) = M_{\mathsf{RSDCG}}(.)$ $bq = 5.00, b = 2.25, \delta = 4.61, \gamma = 0.50, \mu = -0.79$	0.34	0.41	0.42
Comp. Model, $\theta_{\mathsf{pos}}(j) = \theta_{\mathsf{RSRBP}}(j)$, $M(.) = M_{\mathsf{RSRBP}}(.)$ $p = 0.99, q = 0.72, \delta = 5.00, \gamma = 0.42, \mu = -0.54$	0.31	0.38	0.46

Table 4.16: Correlation between session satisfaction ratings and computed session scores using RSDCG, RSRBP, and two weighted mean models with $\theta_{\mathsf{pos}} = \theta_{\mathsf{RSDCG}}$ and $\theta_{\mathsf{pos}} = \theta_{\mathsf{RSRBP}}$. Values in red are “self-optimised”, with parameter tuning and selection based on the reported quantity.

Figure 4.17 shows how the memory-based user model works for a session with five queries. After the user exited from the kth SERP, a new memory of information about that SERP is then created. That is, $\sigma(k, k) = 1$. If the user submits the $k + 1$th query and inspects the corresponding SERP, all memories of previous SERPs are updated at the end of inspection using the following formula:

$$\sigma(j, k+1) = \sigma(j, k) \times \beta(j, k+1) \qquad \text{for } j \leq k\,.$$

When the user exits from a session with a total of $|\mathcal{S}|$ queries, the quality of that session can be measured by a metric score, $sM(.)$, computed using Equation 4.3 (page 132), $sM(\vec{\mathbf{r}}) = \sum_{j=1}^{|\mathcal{S}|} \theta(j) \cdot M(\vec{r_j})$, with the following $\theta(j)$:

$$\theta(j) = \sigma(j, |\mathcal{S}|) \Big/ \left(\sum_{k=1}^{|\mathcal{S}|} \sigma(j, k) \right) ,$$

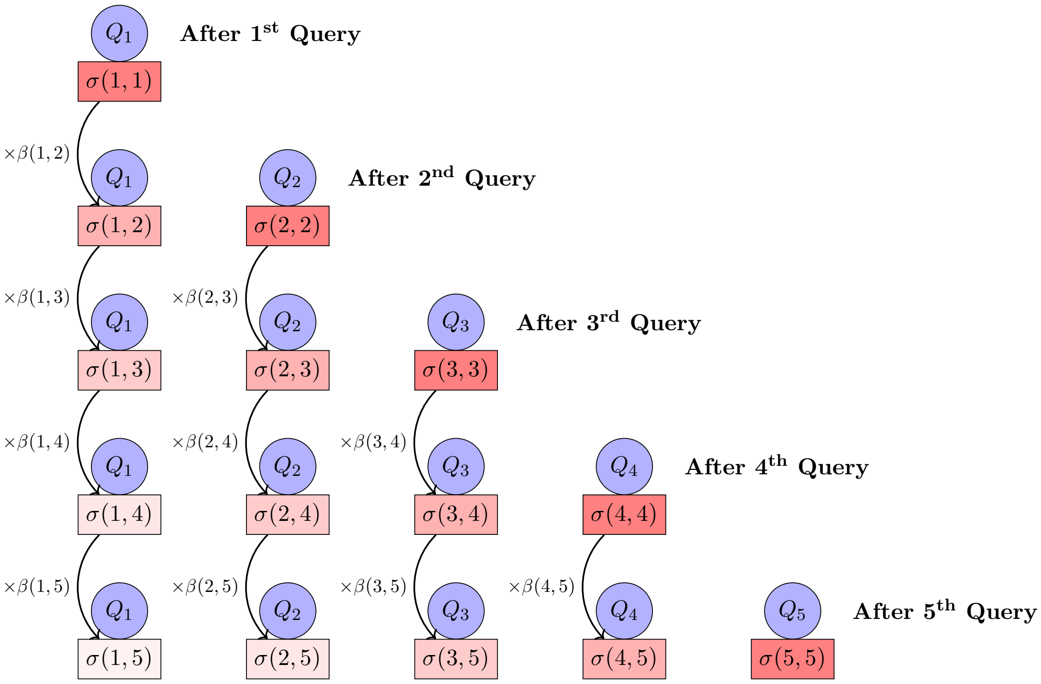


Figure 4.17: Illustration of how memories of past SERPs decay as the user reformulates their queries. This illustration shows an instance for a session with a total of five queries.

where $\sigma(j, |\mathcal{S}|) = \prod_{k=j+1}^{|\mathcal{S}|} \beta(j, k)$. Note that the normalisation factor (or the denominator) is needed because $|\mathcal{S}|$, the number of queries carried out, is one of the determinants for session satisfaction (see Table 4.10 on page 161). Suppose $M(.)$ is bounded to lie between zero and one, and represents *query utility*, the information gain derived from the SERP of the query with respect to the user's goal [246]. In this case, $sM(.)$ can be interpreted as the *rate of remembered query utility* per SERP inspected.

The challenge of this framework is to find a good model for $\beta(j, k)$, so that $sM(.)$ correlates with session satisfaction. Recall that Liu et al. [139] use an exponential smoothing technique to compute $\theta_{\mathsf{Liu}}(j)$ (see Equation 4.4). Hence:

$$\beta_{\mathsf{Liu}}(j, k) = \frac{k^{\lambda} - 1}{(k-1)^{\lambda}},$$

where $0 \leq \lambda < 1$ is the parameter that controls how fast each memory decays as the user reformulates their queries. This definition also implies that the forget factor $\beta_{\mathsf{Liu}}(j, k)$ is the same for each query (that is, $\beta_{\mathsf{Liu}}(j, k)$ does not depend on j), and also increases as a function of k, the number of queries that have been submitted by the user so far.

Zhang et al. [242] propose that the remaining memory of the j th query at the end of a session is computed by $e^{-\delta\cdot(|\mathcal{S}|-j)}$, where $\delta > 0$ controls the rate at which the user forgets (see Equation 4.5 on page 133). This suggests that each memory $\sigma(j,.)$ decays by factor of $e^{-\delta}$ at each step of k:

$$\beta_{\mathsf{Zhg}}(j,k) = e^{-\delta} \,.$$

Hence, $\beta_{\mathsf{Zhg}}(j,k)$ is constant across all queries, and also does not depend on the query submission step k.

Quality-Sensitive $\beta(j,k)$. Analysis and experimental results described in Sections 4.7 and 4.8.1 suggested that the query-level quality information should be incorporated into the forget factor $\beta(j,k)$ in order to improve the correlation strength between $sM(.)$ and session satisfaction. This is also related to the famous psychological heuristic, *peak-end rule* [113]. Hence, we propose the following definition:

$$\beta_{\mathsf{Qty}}(j,k) = M(\overrightarrow{r_j})^{\nu} \,,$$

where $\nu \geq 0$ is the parameter controlling how the user will forget the query based on its quality. Hence, the rationale of this model is that the user will gradually forget SERPs generated using past queries, but they will not easily forget good queries.

Tuning and Evaluation. With the same experiment setting as is used to generate Table 4.14, three proposals for $\beta(j,k)$ were validated on three datasets. Table 4.17 shows these results. For self-optimised cases, the use of $\beta_{\mathsf{Qty}}(j,k)$ provides superior results, particularly when the aggregation is based on query satisfaction scores (QSat) for both THUIR2 and THUIR3, and on query effectiveness scores (RBP) for the J&A dataset. However, as shown in Table 4.18, this superiority is only significant on THUIR3 data.

Consider again Table 4.14 on page 167. Even though the memory-based aggregation method (particularly with $\beta_{\mathsf{Qty}}(j,k)$) is no better than the weighted mean approach, the former has at least three key points. First, it is still significantly better than baseline approaches described in Table 4.14. Second, it corresponds to an obvious user model (that is, the notion of forgetfulness), and is easy to interpret the scores. Third, it is simpler than the latter method, and has only a single parameter that need to be fitted or otherwise selected.

$\beta(j,k)$	THUIR3		THUIR2		J&A
	QSat	RBP	QSat	RBP	RBP
Tuned on THUIR3 (QSat)					
$\beta_{\mathsf{Liu}}(j,k)$, $\lambda = 0.40$	0.78	0.25	0.67	0.32	0.41
$\beta_{\mathsf{Zhg}}(j,k)$, $\delta = 0.73$	0.77	0.25	0.67	0.32	0.41
$\beta_{\mathsf{Qty}}(j,k)$, $\nu = 0.88$	0.80	0.25	0.72	0.28	0.42
Tuned on THUIR3 (RBP $\phi = 0.8$)					
$\beta_{\mathsf{Liu}}(j,k)$, $\lambda = 1.00$	0.72	0.27	0.71	0.36	0.41
$\beta_{\mathsf{Zhg}}(j,k)$, $\delta = 5.55 \times 10^{-17}$	0.72	0.27	0.71	0.36	0.41
$\beta_{\mathsf{Qty}}(j,k)$, $\nu = 2.30 \times 10^{-15}$	0.72	0.32	0.71	0.35	0.42
Tuned on THUIR2 (QSat)					
$\beta_{\mathsf{Liu}}(j,k)$, $\lambda = 0.94$	0.73	0.27	0.71	0.36	0.41
$\beta_{\mathsf{Zhg}}(j,k)$, $\delta = 5.55 \times 10^{-17}$	0.72	0.27	0.71	0.36	0.41
$\beta_{\mathsf{Qty}}(j,k)$, $\nu = 0.32$	0.78	0.27	0.73	0.29	0.41
Tuned on THUIR2 (RBP $\phi = 0.8$)					
$\beta_{\mathsf{Liu}}(j,k)$, $\lambda = 0.98$	0.73	0.27	0.71	0.36	0.41
$\beta_{\mathsf{Zhg}}(j,k)$, $\delta = 0.00$	0.72	0.27	0.71	0.36	0.41
$\beta_{\mathsf{Qty}}(j,k)$, $\nu = 0.00$	0.72	0.27	0.71	0.36	0.41
Tuned on J&A (RBP $\phi = 0.8$)					
$\beta_{\mathsf{Liu}}(j,k)$, $\lambda = 0.69$	0.76	0.26	0.70	0.35	0.42
$\beta_{\mathsf{Zhg}}(j,k)$, $\delta = 0.20$	0.75	0.27	0.70	0.35	0.41
$\beta_{\mathsf{Qty}}(j,k)$, $\nu = 3.22$	0.77	0.25	0.68	0.29	0.45

Table 4.17: Correlation between session satisfaction ratings and computed session scores using either query satisfaction (QSat) or query scores (RBP), for three definitions of $\beta(j,k)$, and with tuning based on a variety of resources. Values in red are self-optimised, with parameter tuning and selection based on the reported quantity. This table can be compared with Table 4.14 on page 167 and Table 4.16 on page 169.

4.9 Summary

Users typically interact with search engines by submitting multiple queries when addressing an information need. This behaviour requires extending traditional query-based IR evaluation, so that a multi-query session can be assessed as a single unit. This chapter has addressed two goals for session evaluation. The first goal is the development of a user model for an adaptive session metric in the context of a session test collection, where each topic is assigned to a sequence of static queries (Sections 4.4, 4.5, and 4.6). The second goal is an investigation of the fitted relationship between session satisfaction ratings and individual query satisfaction ratings (or query scores), which is critical for the development of query-to-session aggregation functions when user observation data (such as the query sequence exit point) is known (Sections 4.7 and 4.8).

Other method	$\beta_{\mathsf{Qty}}(j,k)$		
	THUIR3 (QSat)	THUIR2 (QSat)	J&A (RBP)
$\beta_{\mathsf{Liu}}(j,k)$	**0.024**	0.165	0.448
$\beta_{\mathsf{Zhg}}(j,k)$	**0.001**	0.169	0.464

Table 4.18: The p values of two-sided Hotteling's t test [90] for comparing Pearson's correlation coefficients between the aggregation method that uses $\beta_{\mathsf{Qty}}(j,k)$ and any of other two methods, computed upon self-tuned arrangements on THUIR3 (only QSat), THUIR2 (only QSat), and J&A (RBP). A significance level of 0.05 is used to test the null hypothesis that the correlation coefficients between any two models are not different.

User Model and Adaptive Session Metric. In Section 4.4, we extended the C/W/L framework to session-based effectiveness evaluation, and demonstrated that existing session-based user models can be explained by this generalised evaluation framework. In the session-based C/W/L framework a user model (describing a population of users) is characterised by two behaviours: their *conditional continuation probability* at rank i when examining the jth SERP, $C(j,i)$; and their *conditional reformulation probability*, $F(j)$. These two quantities are sufficient to specify a session-based effectiveness metric.

Section 4.5 identified factors that contribute to $C(j,i)$ and to $F(j)$ using three commercial search interaction logs. Our results support the observations of Moffat et al. [155] in regard to the conditional continuation probability within each query. The findings derived from the logged behaviours also confirm that at least three factors affect the conditional reformulation probability $F(j)$: the query position j in the session; the user's expected number of relevant documents T at the beginning of search; and the unmet number of relevant items to date, $T_{j,*}$. Further, this study has confirmed that these three factors are all positively correlated with $F(j)$.

Section 4.6 proposed a new session-based metric, session INST (sINST), by crystalising the relationships in regard to both $C(j,i)$ and $F(j)$ described in Section 4.5. In contrast to existing session-based metrics, such as LCYsRBP, sDCG, and KsDCG; sINST is adaptive, and provides a better fit to observed user behaviour than do those previous metrics. Further, we also propose a less-expensive "expectation" method for computing sINST, providing an alternative to the Monte Carlo approach that requires a large number of randomised trials.

Session Satisfaction Model. When the knowledge of how many times the user reformulated, what queries they submitted, and what satisfaction ratings they provided is available, a connection between session satisfaction and factors from the individual queries can be established. Section 4.7 has shown that models based on the combination of both quality and positional variables provide a better correlation with user session satisfaction than those based on query position alone. This confirms the effect of the peak-end rule [113] on search session experience.

With the findings established in Section 4.7 in regard to factors affecting session satisfaction, Section 4.8 proposed two session satisfaction models, which are useful for scoring a session when user observation data is available. The first model (Section 4.8.1) is based on a weighted mean approach, where the positional- and quality-based factors are merged using a linear combination method. Further experiment results demonstrated that this method outperforms previous approaches for self-tuned arrangements on three different search datasets.

The weighted mean approach, however, lacks an obvious user model, and it is difficult to explain its rationale from a human perspective, though our experiments showed that it is better than the baseline approaches. In Section 4.8.2, we generalised two recent aggregation approaches [139, 242], and proposed a memory-based aggregation framework, which has a clear connection with a user model. With this framework, one important quantity that needs to be estimated is the forget factor, denoted by $\beta(.)$. Our experiment suggested that $\beta(.)$ is sensitive to query quality, indicating that the user tends to remember good queries, and that there is a clear gap between the use of query effectiveness score and the use of query satisfaction score for the aggregation mechanism. An important direction for future work is to undertake a more detailed exploration into the difference between satisfaction- and effectiveness-based aggregations.

Chapter 5

Metrics, User Models, and Satisfaction

A good metric should give rise to scores that have a strong positive correlation with user satisfaction ratings. A metric should also correspond to a plausible user model, and hence provide a tangible manifestation of how users interact with search rankings. Recent work has focused on metrics whose user models accurately portray the behaviour of search engine users. In this regard, Chapter 3 introduced several tools for inferring continuation probabilities and gaze distributions from user observation data, which are useful for the development of *user-centric* metrics. Further, Chapter 4 presented an approach that makes use of the tools described in Chapter 3 to develop a metric-based user model for session evaluation through the lens of continuation and reformulation probabilities, and also reported an investigation of factors affecting session-level satisfaction.

This chapter concerns the issue of meta-evaluation. Section 5.1 describes an integrated view of the C/W/L meta-evaluation framework, connecting several concepts, including metrics, users, user model accuracy, and satisfaction. The research motivation and questions are also presented in Section 5.1.

After reviewing previous work for the relationship between offline metrics and user satisfaction in Section 5.2, and describing datasets used in this study (Section 5.3), Section 5.4 then presents correlation coefficients between the scores of a wide range of metrics and user-reported satisfaction ratings at both query- and session-levels. Section 5.5 then describes our approach for measuring the accuracy of a user model from the perspective of the C/W/L framework. Finally, Section 5.6 considers the question of whether the accuracy of a metric is connected with its correlation with user satisfaction. This study develops an important new framework for metric meta-evaluation, and at the same time demonstrates

The material in this chapter (except Sections 5.5.4 and 5.6.2) is based on the following published paper:

- Alfan F. Wicaksono and Alistair Moffat. Metrics, User Models, and Satisfaction. In *Proc. WSDM*, pages 654–662, 2020.

that the metrics and parameter settings that correlate well with user satisfaction closely match the user models and parameter settings that best fit observed user behaviours.

5.1 Motivation and Research Question

Meta-evaluation of search effectiveness metrics depends on which aspects of the measures are being evaluated. Historically, the focus has been on the relationship between metric scores and user satisfaction, measuring how accurately metric scores predict the satisfaction or performance of users as they carry out particular search tasks. However, satisfaction is an indirect observable, meaning that the ground-truth of this concept lies in the user's mental state [117]. Kelly [117] further suggests that dealing with satisfaction for the evaluation of IR systems requires two important instruments: indirect measures that approximate the ground-truth of satisfaction, and a method for how the approximation should be elicited from users. In regard to the latter, Likert scale survey questions are typically employed to capture user satisfaction, with a five-point style (ranging from *unsatisfied* to *very satisfied*) being particularly popular [46, 84, 106, 107, 139, 145, 242]. In addition to the Likert-style satisfaction item, experiments can consider questions that address system preference [89] and system response time [117].

Even though the use of user satisfaction as an indicator of search success has provided conflicting results [84] (see Section 2.5.1 on page 62), user satisfaction is nevertheless tightly coupled with the effectiveness of an IR system [199]. Cooper [55] argues that subjective user satisfaction is a primary measure of system performance, and should be a basis for the development of any effectiveness metric. Recent developments in effectiveness metrics have made use of user-reported satisfaction ratings (an approximation to the ground-truth satisfaction) to meta-evaluate the extent to which metric scores are aligned with satisfaction [106, 139, 241, 242]. This set of ratings is typically collected by asking users *how satisfied were you with the set of results returned by the system?*

The connection between metric scores and user satisfaction is just one desirable aspect of an effectiveness metric. A good metric should also correspond to an obvious user model, and that this model should have a strong relationship with observed user behaviours, with the benefit that the metric can explain the visible activities engaged in by the user as they interact with the ranked list of results. Chapters 3 and 4 have described our efforts to realise this idea by incorporating user behaviours into two hypothetical functions: the continuation probability, $C(\cdot)$, at the SERP level; and the reformulation probability, $F(.)$, at the session level. The interrelated links among metric, satisfaction, and user

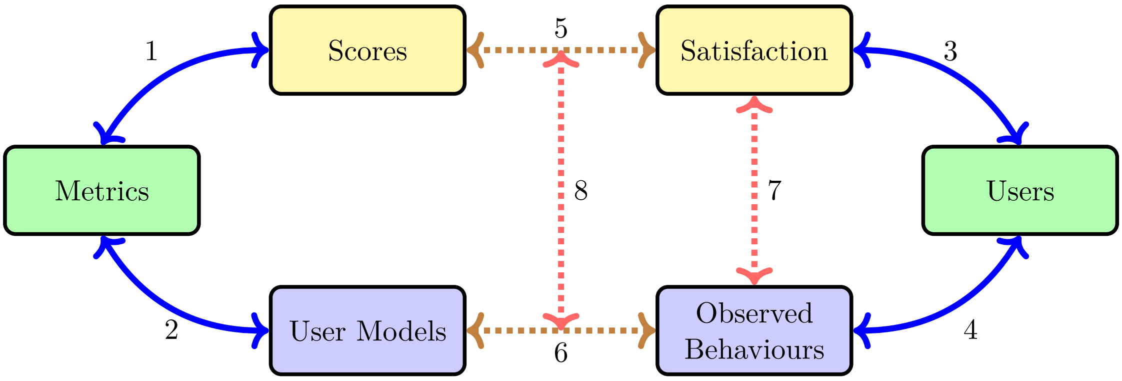


Figure 5.1: Proposed meta-evaluation framework. The C/W/L framework is illustrated using three entities on the left-hand side (metrics, scores, and user models).

behaviour form a C/W/L-based meta-evaluation framework for effectiveness metrics. This framework is visually described in Figure 5.1. There are six entities, and at least eight useful relationships in this framework.

The C/W/L framework resides in three entities (metrics, scores, and user models) and two links (links 1 and 2). The link between a metric and its score is deterministic (link 1), where a score can be computed from a gain vector, $\overrightarrow{r}$, and the $W(\cdot)$ function. Other types of scores can also be computed using the $L(\cdot)$ function. Link 2 describes the *duality* between metrics and user models via three functions: $C(\cdot)$, $W(\cdot)$, $L(\cdot)$, plus $F(.)$ at the session level.

Both links 3 and 4 require several methodologies for eliciting *direct* and *indirect* observables from users. Lab-based user studies, living lab [25], and online experimentations (such as A/B testing) are examples of those methodologies. Link 3 is established by two components: an operational definition of satisfaction, and instruments for the approximation of the ground-truth of satisfaction [117]. In this study, link 3 is specified by Likert-style satisfaction ratings reported by users when they exit a SERP or complete a search session. Link 4 deals with *direct* observables from users, such as eye fixations, clicks, mouse-hovers, mouse-scrolls, and dwell-time, and may require a specific technology, such as an eye-tracker.

In general both links 5 and 6 measure to what extent a metric reflects the quality of user experience, a fundamental question for any user-based metric. Link 5 calculates the extent to which metric scores predict satisfaction ratings, a qualitative measure of search success. Computing correlation coefficients, such as Pearson's r, Spearman's ρ, and Kendall's τ, is very popular in this regard [5, 46, 104, 139, 145, 213, 241]. Link 6 calculates the extent to which predicted user behaviour correlates with observed user behaviour. Azzopardi et al. [20] propose a method for comparing a metric's stopping probability with observed last-click positions. Link 5 and link 6 can both be used to gain insights or corroboration.

Satisfaction ratings cannot be observed at scale, particularly for online experiments. In the absence of such ratings, online experiments usually make use of other observed behaviours that predict satisfaction, such as clicks [65] or query reformulations [78, 213]. In contrast to satisfaction ratings, observed behaviours, such as clicks, can be collected at scale, and can be available in real time. However, such implicit feedback must have a proper interpretation in regard to the criteria of user satisfaction [167]. For example, when a user clicks on multiple items in the SERP, two opposing interpretations are possible: *unsatisfied* or *satisfied* [117]. Link 7 addresses the signals users exhibit when they are satisfied, and how those signals should be interpreted.

Saracevic [188] and Moffat and Zobel [151] argue that user experience is the primary aspect for measuring search utility. It follows that user experience should be quantified based on the set of results that has been inspected by the user, and not on part of the ranking that has not been viewed, or on items that were not retrieved by the system. In this chapter we also ask whether the metrics with user models that fit typical observed behaviours tend to be the metrics that correlate well with user satisfaction ratings. This is particularly critical because collection of satisfaction ratings is usually limited as they require user studies in a laboratory, while logged behaviours can be collected at scale from an operational system. Link 8 in Figure 5.1 depicts that relationship.

Research Questions. This chapter emphasizes links 5, 6, and 8, and addresses the following research questions:

> **RQ 5.1**: To what extent do metric scores correlate with user satisfaction, at both query- and session- levels?
>
> **RQ 5.2**: To what extent do user models predict observed behaviours?

To answer these questions, this chapter investigates the relationship between various user-based metrics and satisfaction ratings for both query- and session-level evaluation, exploring possible aspects that affects this relationship. Next, this chapter presents our proposed method for measuring the accuracy of a user model (that is, the extent to which the user model predicts user behaviour) from the perspective of the C/W/L framework. Finally, this chapter addresses the question of whether a correlation between metric score and user satisfaction is, to some extent, connected with the user model accuracy.

5.2 Previous Work

Similar to the study described in this chapter, several authors have computed correlation coefficients between effectiveness metric scores and user-reported satisfaction ratings. Note that different work might use different datasets and experiment settings. Hence, the reported correlation coefficients (such as Pearson's r, Spearman's ρ, or Kendall's τ) cannot be compared.

In 2007, Al-Maskari et al. [5] collected a sample of 104 queries from `Google.com`, and compute correlation coefficients between three-point user-reported SERP satisfaction with each of four effectiveness metrics: precision, cumulative gain, discounted cumulative gain (DCG), and normalised discounted cumulative gain (NDCG), and find that satisfaction has relatively low correlation with NDCG (Pearson's $r \approx 0.20$), but moderate correlation with the other three metrics ($0.50 \leq r \leq 0.79$). In the same year, Huffman and Hochster [91] randomly draw 200 queries from `Google.com` in the middle of 2006, and then ask raters to provide relevance and satisfaction ratings. They find that the relevance of the first query in a session, measured by a DCG-like metric, has a strong positive relationship with session satisfaction ratings ($r \approx 0.72$), and that the correlation becomes stronger when the taxonomy information (navigational or non-navigational) is incorporated into the model ($r \approx 0.80$).

In 2016, Jiang and Allan [105], and Jiang and Allan [104] employ a lab-based search log containing 80 sessions with user-reported 5-point session performance ratings to observe the relationship between session satisfaction and several metrics. Session scores are computed by taking the mean over the individual query-level metric scores. Several query-level metrics are used: Prec@K, AP, RR, graded average precision [171], RBP, ERR, DCG, NDCG, TBG, and U-measure (UM). The resultant coefficients show that the session metric sDCG has a negligible correlation with session satisfaction ($r \approx 0.01$). However, when effort information is incorporated (sDCG divided by the number of queries in the session), the correlation dramatically increases ($r \approx 0.40$). Other session metrics such as expected NDCG [115], gives $r \approx 0.35$. When session scores are inferred by averaging query scores, two effort-based query-level metrics, TBG and U-measure, give the highest correlation coefficients ($r \approx 0.44$); graded average precision, AP, and RR result in very low correlation coefficients ($r \leq 0.21$); and the remaining metrics lead to $0.30 < r < 0.40$.

Mao et al. [145] investigate the difference between *relevance* and *usefulness* and their relationship to satisfaction. They compute correlation coefficients between several metrics (including those based on click sequence) and both SERP- and session-level satisfaction ratings. At SERP-level, they show that metrics based on click sequence achieve better correlation coefficients when the gain function is defined based on the document usefulness ($r \approx 0.75$) than when it is defined as a function of the document relevance ($r \approx 0.56$). Other relevance-based metrics are also used: AP with $r = 0.19$, DCG with $r = 0.30$, and ERR with $r = 0.26$. At the session level, session metrics utilising the document usefulness also correlate with session satisfaction ($r \approx 0.52$) better than those defined based on the document relevance ($r \approx 0.33$).

Following the work of Mao et al. [145], Liu et al. [139] recently compute correlation coefficients between user-generated satisfaction ratings and two types of metrics: those based on click sequence and those based on ranked list. They suggest that metrics based on ranked list should have a decreasing weight function, while those based on click sequence do not need to consider top-weightedness.

Zhang et al. [241] use a dataset consisting of 2,685 single-query sessions to compute correlation coefficients between 5-point satisfaction ratings and each of both S-BPM and D-BPM. The results show that both metrics achieve $r \approx 0.55$ for informational queries, an outcome that is better than any of RBP with $\phi = 0.8$, DCG, AP, and ERR ($0.39 < r < 0.50$); and that incorporating adaptivity into BPM (that is, D-BPM) increases the correlation with satisfaction.

Chen et al. [46] compare four offline metrics (cumulative gain, DCG, RBP, and err) and various online metrics based on clicks, mouse-scrolls, mouse-hovers, and dwell-time, computing correlation coefficients with query-level satisfaction ratings. In their experiments, RBP with $\phi = 0.8$ provides the highest positive correlation coefficient ($r = 0.45$); and some online metrics (*maximum scroll distance*, minimum reciprocal click rank, the number of clicks, and query dwell time) are useful surrogates for satisfaction; ERR provides better correlation with satisfaction on navigational queries, rather than on either informational or transactional queries; and incorporation of mouse-hover information into click-based online metric, such as minimum reciprocal click rank, improves the relationship with satisfaction.

In 2018, Thomas et al. [213] use a sample set of 994 (mostly) navigational queries derived from Bing.com to compute correlation correlation between several effectiveness metrics (Prec@K, RR, ERR, RBP, SDCG@K, and INST) and query *non-reformulation rates*, a surrogate for user-reported satisfaction ratings. After tuning the metric parameters, shallow metrics such as RBP with $\phi = 0.1$, Prec@1, and RR provide reasonable prediction of success (Spearman's $\rho \approx 0.20$) when all items on the SERP are presented, including organic items and advertisements. More importantly, when user behaviours related to various types of results (such as, advertisements and images) are incorporated into $C(i)$,

	J&A	THUIR1	THUIR2	THUIR3	MS [213]
Source	lab	lab	lab	lab	`Bing.com`
Sessions	80	2,435	223	450	–
SERPs	388	291	933	1,548	994
SERP size	9	10	10	≥ 10	10 – 12
Rel. judgement	3-level	4-level	4-level	4-level	4-level
Usefulness judg.	no	no	4-level	4-level	no
Impression	eye gazes	no	no	no	no
Clicks	yes	yes	yes	yes	no
Mouse-hover	no	yes	no	yes	no
Query ratings	no	5-level	5-level	5-level	no
Non-reform. rate	no	no	no	no	yes
Session ratings	5-level	no	5-level	5-level	no

Table 5.1: Collection of datasets from four lab-based Web search user studies, and one commercial search engine.

the correlation coefficients increase for all metrics ($\rho \approx 0.23$).

This chapter utilises some of the datasets from these past studies [46, 104, 105, 139, 145, 213] and re-runs some of their experiments, but with different scenarios, and with the use of a wide range of model-based metrics. This chapter also concerns the interaction between metric scores, user model accuracy, and satisfaction, which were not addressed in these past studies. Recall that we regard a metric as not only generating scores, but also as describing user behaviour. The latter can then be compared with observed behaviour, to yield a second and equally important set of correlation scores.

5.3 Datasets

The study described in this chapter employs five pre-existing datasets. Four are from past lab-based user studies [46, 104, 139, 145], and one from a commercial search engine, `Bing.com` [213]. The four lab-based datasets have also been used in Chapters 3 and 4. Table 5.1 summarises these five datasets.

The J&A dataset contains logged behaviours from 80 sessions (388 queries) on 20 tasks, relevance judgements for the top-9 results in each SERP, user-reported 5-point session-level satisfaction ratings, and 5-point task difficulty ratings [107]. During a search session, two sources of user behaviours were recorded: clicks and eye-fixations. An eye-fixation is recorded using a Tobbi 1750 eye-tracker, and is for a minimum of 100 milliseconds. On a per-session basis, users on average submitted 4.9 queries, clicked on 9.3 unique

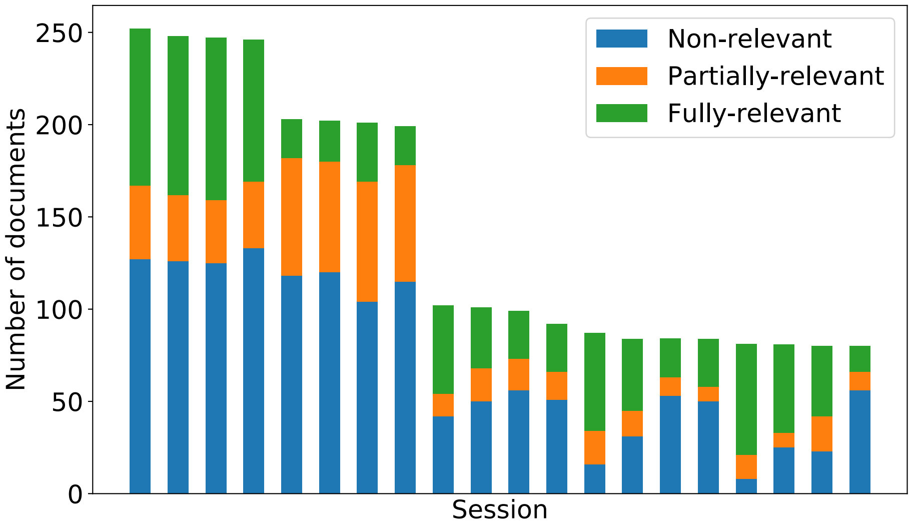


Figure 5.2: Total number of judged documents per session in the J&A dataset for 20 sessions with the highest number judged documents. The set of judged documents is further divided based on the relevance level.

items, and viewed 16.1 unique snippets [107]. Jiang and Allan [104] further show that satisfaction ratings have a strong negative correlation with task difficulty ratings (Pearson's $r = -0.79$). The relevance of a document is judged by the participant who runs the search session with respect to a topic, not with respect to a query. Figure 5.2 shows the number of judged documents per session in the J&A dataset for 20 sessions with the highest number of judged documents.

Chen et al. [46] employ THUIR1 to meta-evaluate several offline and online metrics. This dataset contains 291 static queries (and their corresponding SERPs), and user behaviours from 98 users. In contrast to other datasets, THUIR1 contains taxonomy information (navigational, transactional, or informational) and cognitive level (remember or understand) for each task. Participants were not permitted to submit their own queries or to reformulate the queries. At the beginning of a session, the participant was asked to read a pre-defined query and a task description. After that, they were directed to a SERP containing 10 items, where they interacted with the list of documents to complete the task. After they exited the SERP, they were asked to provide a satisfaction rating, reflecting their search experiences. With this scenario, there are 2,435 single-SERP sessions, each a combination between SERP ID and User ID. Hence, a SERP can be evaluated by a group of users, enabling insights about general behaviours from a group of users, as opposed to just a single user, on a single SERP.

Mao et al. [145] construct THUIR2 for investigating the difference between document *usefulness* and document *relevance*, and their connection with satisfaction ratings. Each SERP contains 10 items, but only the top-5 results plus those clicked were judged. Usefulness-based judgements are also included, but only for clicked items in each SERP. This dataset originally contains 225 sessions and 935 queries. There are 933 SERPs for which at least the top-5 items were judged, and there are 223 sessions with all corresponding SERPs that have been judged on their top-5 positions.

The THUIR3 dataset, constructed by Liu et al. [139], has the same properties as the THUIR2 dataset. However, THUIR3 contains more sessions than THUIR2, and the relevance judgements for the former dataset covers most items in each SERP. There are 1,259 SERPs for which at least top-10 items have been judged. Similar to THUIR2, THUIR3 provides usefulness judgements only for items that were clicked in each SERP.

The fifth dataset (MS) was constructed by Thomas et al. [213]; and contains 994 head queries drawn from `Bing.com`, four-point relevance judgements, and a set of *query non-reformulation rates* as a proxy for user satisfactions. However, there are only 876 queries that are associated with the query non-reformulation rates. Thomas et al. [213] further define a query reformulation as a situation where a query is followed by a second one in the same session that has at least $1/3$ of the query terms in common (such as, "restaurant near me" → "pizza near me").

In addition to the five datasets described in Table 5.1, this study also utilises a sample of $1,060,216$ queries from `Yandex.ru`, for which the corresponding SERPs are fully judged. Note that this collection of queries does not contain user-reported ratings or query non-reformulation rates, and is also a subset of the `Yandex.ru` dataset that has been used in Chapter 4 to find empirical evidence for several hypothetical behaviours regarding $C(\cdot)$ and $F(.)$. In this chapter, this dataset will be primarily employed in Section 5.5 to measure the extent to which user models of several static metrics predict observed behaviours in terms of three functions $C(\cdot)$, $W(\cdot)$, and $L(\cdot)$.

5.4 Metric Scores and Satisfaction

We now address **RQ 5.1**, exploring correlation coefficients between metric scores and satisfaction ratings at both query- and session-levels (link 5 in Figure 5.1 on page 177). A range of metrics are tested, including static and adaptive ones. This study also investigates the difference between the *expected total gain* (ETG) and *expected rate of gain* (ERG) versions of several adaptive metrics (see Equation 2.26 on page 2.26 and Equation 2.29 on page 48), and computes a distribution of correlation coefficients for a set of SERPs that is evaluated by a group of users. Before addressing those issues, *residual measurements* are discussed, and computed for the five datasets used in this study.

5.4.1 Query-Level Satisfaction

Residual Measurement. Moffat and Zobel [151] introduce the notion of residuals, the uncertainty in a metric score as a result of unjudged items. This is a critical issue for unbounded effectiveness metrics, such as RBP, INST, and IFT. In practice, the length of a ranking for an offline evaluation is finite at depth d. Hence, two sources of uncertainty arises: unjudged documents prior to rank d, and unknown items beyond rank d. With a weighted-precision metric, the residual is just the difference between a lower bound and an upper bound score. The former is computed by assuming that all unjudged documents are nonrelevant, and the latter by assuming that they are all fully relevant.

Residuals for some metrics, such as RBP, can be formulaically computed using a closed form. When the items prior to rank d are all judged, the uncertainty value of RBP raised by the unknown documents from depth $d+1$ is computed as ϕ^d. Adaptive metrics, however, require a more complex computation, because their $W(i)$ functions are affected by relevance information.

The judgement pooling depths for the J&A, THUIR1, THUIR2, THUIR3, and MS datasets were 9, 10, 5, 10, and 12, respectively. With RBP $\phi = 0.8$, the corresponding residuals raised by unjudged documents are 0.13, 0.11, 0.33, 0.11, and 0.07, for the ERG versions. The residuals associated with $\phi = 0.1$ are much lower than those values (10^{-9}, 10^{-10}, 10^{-5}, 10^{-10}, and 10^{-12}). Figure 5.3 shows the residuals for two adaptive metrics: INST and IFT with $T \in \{2, 3\}$ computed using all five datasets. Residuals on THUIR2 are the highest, since the judgements are only for top-5 documents. Meanwhile, datasets that contain deeper judgements, such as THUIR3, have much lower residuals. All datasets except THUIR2 on average yield moderate residuals. Hence, the metric scores generated from the judgements in the J&A, THUIR1, THUIR3, and MS datasets, can be used with confidence to compute correlation coefficients with satisfaction ratings in this study.

Query-Level Correlations. This section calculates correlation coefficients (Pearson's r) between the query-level satisfaction ratings and the scores computed using various effectiveness metrics, including several model-based metrics that are not addressed in the initial explorations [46, 104, 139, 145, 213]. For example, INSQ and IFT were not considered

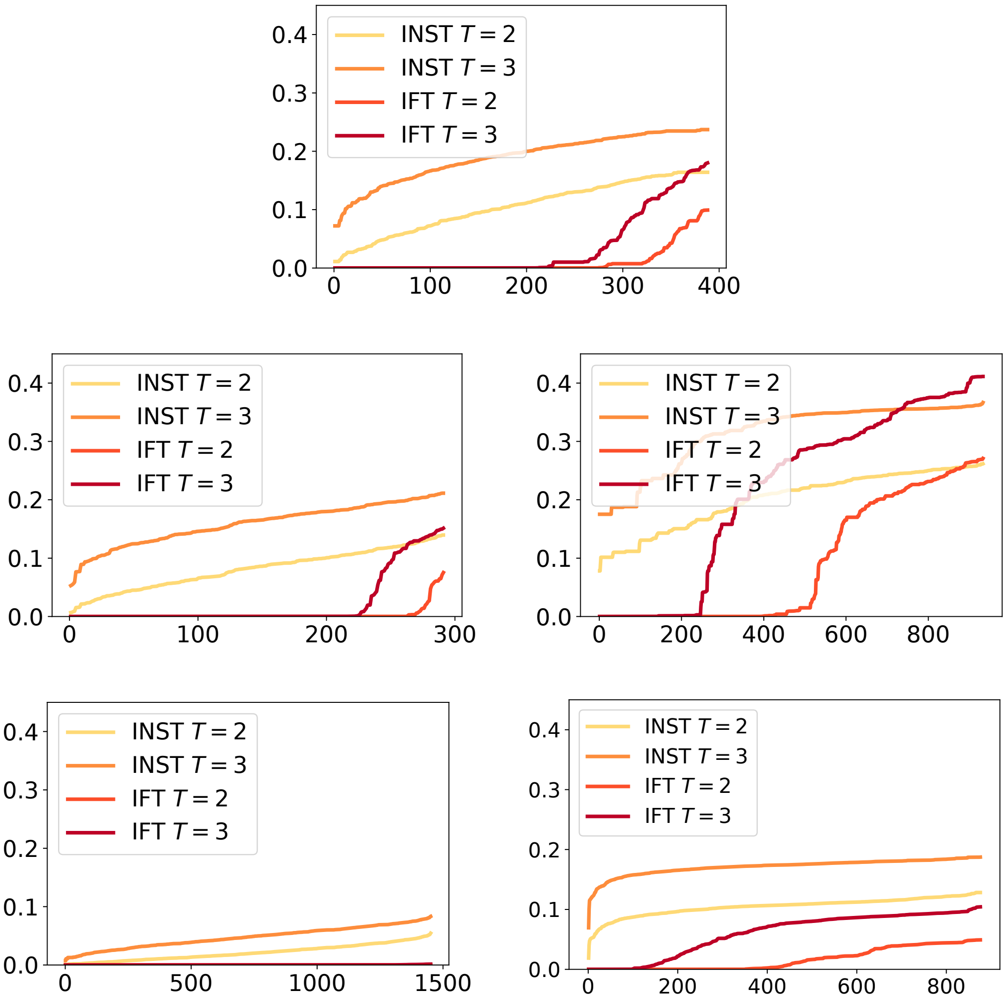


Figure 5.3: Residuals of INST and IFT using the J&A (first row), THUIR1 (second row, left), THUIR2 (second row, right), THUIR3 (third row, left), and MS (third row right). The queries (x-axis) are sorted by residual value (y-axis).

in the original experimentation that originally created the five datasets. The THUIR1, THUIR2, THUIR3, and MS datasets are employed; for THUIR2, all SERPs are truncated at $K = 5$, since relevance judgements are only available for the top-5 results in the SERP; the J&A dataset is not considered, since it does not contain query-level satisfaction ratings (see Table 5.1 on page 181).

In the first experiment, three metrics that depend on knowledge of R (that is, the number of relevant documents for a query) are considered: AP@K, QM@K, and NDCG@K (see Equation 2.9 on page 24, Equation 2.17 on page 29, and Equation 2.18 on page 30). For all datasets except J&A relevance judgements were on a per-SERP basis with respect to the issued queries. Note also that rankings are all truncated at depth K (that is, $K = 10$ for THUIR1, THUIR3, and MS; and $K = 5$ for THUIR2). Hence the normalisation factor R is based only on items in the truncated SERP, since the remaining documents beyond depth K are not available.

Two alternative normalisation factors for both AP@K and QM@K are considered (see three alternatives for defining AP@K on page 24). Let Z be the normalisation term for these two metrics, as a generalisation for R. For example, the definition of AP@K described in Equation 2.9 (page 24) can be generalised as follows:

$$\mathsf{AP@K}(\overrightarrow{r}; K) = \frac{1}{Z} \cdot \sum_{i=1}^{K} [\mathsf{Prec@K}(\overrightarrow{r}; i) \cdot r_i] \ .$$

The first alternative is to utilise $Z = \sum_{i=1}^{K} r_i$, where $r_i \in \{0, 1\}$ [23]; the second is to use $Z = K$ [80]. When the former (latter) normalisation is employed, the two metrics are denoted by AP1@K and QM1@K (AP2@K and QM2@K). For NDCG@K, an ideal ranking should ideally be from all documents in the collection that have been judged for a particular query. However, in the case of the four datasets (THUIR1, THUIR2, THUIR3, and MS), an ideal ranking is constructed based on what has been displayed on the SERP. That is, an ideal gain vector is obtained by sorting the original gain vector in a decreasing order.

Table 5.2 shows correlation coefficients between query-level satisfaction ratings and SERP scores using the three metrics: average precision, Q-Measure, and NDCG. Two binary gain mapping functions are considered for the average precision, since it requires the assumption that the relevance is binary (see Section 2.1.5 on page 27). The first gain mapping function, $g_1(r_i)$, returns 1 if $r_i > 0$ and 0 if $r_i = 0$. The second, $g_2(r_i)$, gives 1 if $r_i = r_{max}$ and 0 if $r_i < r_{max}$. For NDCG, an exponential gain mapping function is employed: $g_3(r_i) = (2^{r_i} - 1)/(2^{r_{max}} - 1)$. In the case of ERR, the denominator of $g_3(.)$ is replaced by $2^{r_{max}}$.

Note that average precision, Q-Measure, and NDCG, were not tested in the previous experiments carried out by Thomas et al. [213] and Liu et al. [139]. Chen et al. [46] and Mao et al. [145] did not employ Q-Measure and NDCG; but they used average precision, although they did not explicitly state how they implemented it. Nevertheless, the correlation coefficient computed using AP2@K with $g_1(r_i)$ is close to the coefficient reported by Mao et al. [145] when AP was used; and the coefficient for AP reported by Chen et al. [46] is close to the coefficient computed using AP1@K with $g_2(r_i)$ in our experiment. Recall that all four datasets used in this experiment contain 4-point relevance judgements,

Metric	THUIR1	THUIR2	THUIR3	MS
AP1@K, with $g_1(.)$	0.274	0.060	0.193	0.094
AP1@K, with $g_2(.)$	0.409	0.263	0.322	0.152
AP2@K, with $g_1(.)$	0.258	0.164	0.226	0.027
AP2@K, with $g_2(.)$	0.284	0.241	0.296	-0.041
QM1@K, $\beta = 1$	0.334	0.080	0.209	0.103
QM1@K, $\beta = 10$	0.326	0.043	0.185	0.125
QM2@K, $\beta = 1$	0.331	0.132	0.257	0.028
QM2@K, $\beta = 10$	0.345	0.128	0.265	0.032
NDCG@K	0.364	0.055	0.261	0.097

Table 5.2: Correlation coefficients (Pearson's r) between SERP-level satisfaction ratings and scores from three metrics: average precision, Q-Measure, and NDCG. This experiment uses $K = 10$ for THUIR1, THUIR3, and MS; and $K = 5$ for THUIR2. Note that Q-Measure has the persistence parameter β (see Equation 2.18 on page 30). Blue color represents the three largest coefficients in each column.

$r_i \in \{0, 1, 2, 3\}$. In general AP1@K with $g_2(.)$ provides the highest correlation coefficient for all four datasets. The correlation coefficients are $r = 0.41$ in THUIR1, $r = 0.26$ in THUIR2, $r = 0.32$ in THUIR3, and $r = 0.15$ in MS. Correlation coefficients in the MS column tend to be the lowest compared to values on other columns, indicating that all metrics lack a relationship with the query non-reformulation rates.

In the second experiment, three ad-hoc metrics that do not have R are considered: DCG@K, RR, and ERR. Correlation coefficients are also computed for two recently proposed metrics, iRBU@K (see Equation 2.55 on page 68) and both versions of BPM: static (S-BPM) and dynamic (D-BPM) [241]. The correlation coefficient between iRBU@K scores and satisfaction ratings has not been reported in the previous work. Both S-BPM and D-BPM involve multiple parameters; and this study makes use of default parameters recommended by their developers [241]. Table 5.3 show the resultant coefficients from the second experiment. The metrics in Table 5.3 tend to have higher correlation coefficients than the metrics that depend on R that are described in Table 5.2. The iRBU@K provides the best correlation with query satisfaction ratings in all datasets, except in the THUIR2 dataset, regardless of its parameter value. Pearson's $r \approx 0.50$ is attainable in the THUIR1 dataset for iRBU@K with $\phi = 0.99$; while shallow metrics, such as RR with the gain mapping $g_2(.)$ and iRBU@K with $\phi = 0.10$ perform the best in the MS dataset, where most of the queries are navigational. Dynamic BPM (D-BPM), an adaptive metric, gives the highest correlation coefficient for the THUIR2 dataset ($r = 0.37$); and DCG correlates relatively well with query satisfaction in the THUIR3 dataset (the dataset with the largest number

Metric	THUIR1	THUIR2	THUIR3	MS
iRBU@K $\phi = 0.10$	0.404	0.202	0.265	0.153
iRBU@K $\phi = 0.50$	0.435	0.257	0.321	0.133
iRBU@K $\phi = 0.80$	0.473	0.322	0.373	0.073
iRBU@K $\phi = 0.99$	0.502	0.337	0.333	−0.020
RR@K, with $g_1(.)$	0.199	0.050	0.104	0.125
RR@K, with $g_2(.)$	0.394	0.266	0.318	0.151
ERR@K	0.443	0.271	0.318	0.128
S-BPM	0.427	0.302	0.266	−0.137
D-BPM	0.307	0.365	0.292	−0.083
DCG@K	0.431	0.301	0.363	−0.080

Table 5.3: Correlation coefficients (Pearson's r) between SERP-level satisfaction ratings and scores from six metrics: iRBU@K, RR, ERR, S-BPM, D-BPM, and DCG. This experiment uses $K = 10$ for THUIR1, THUIR3, and MS; and $K = 5$ for THUIR2. Blue color represents the three largest coefficients in each column.

of queries used in this study) compared to other metrics in the same table.

The first and second experiments use metrics employing the original definitions provided by the proposers of the metrics. Some of them can be mapped into the C/W/L framework (such as average precision, RR, and DCG), but the others might not be. The third experiment makes use of various metrics that fit the C/W/L framework. Most of them are originally developed by modelling the $C(\cdot)$ functions that are intended to predict observed behaviours. Three foraging-based metrics, IFT-C1, IFT-C2, IFT are employed using default parameters suggested by Azzopardi et al. [20], unless explicitly noted as variations, such as the parameter T (the user's anticipated volume of relevance). Table 5.4 (page 189) shows the results for the third experiment using the exponential gain mapping function: $g_3(r_i) = (2^{r_i} - 1)/(2^{r_{max}} - 1)$, and Table 5.5 (page 190) displays the resultant correlation coefficients using the linear gain mapping function: $g_4(r_i) = r_i/r_{max}$. In order to draw some general patterns across metrics, all columns except the MS columns are associated with the overall geometric mean (*gmean*) values in the last row. The *gmean* scores for the MS columns are not computed, since the correlation coefficients are very low, and are mostly negative.

For static metrics, the ERG formulations result in the same correlation coefficients as the corresponding ETG versions, since $W(1)$ is constant across SERPs, and since $M_{\mathsf{ETG}}(\overrightarrow{r}) = M_{\mathsf{ERG}}(\overrightarrow{r})/W(1)$ (Equation 2.29 on page 48). The overall *gmean* scores indicate that it is difficult to conclude which of these two metric versions is more correlated with query satisfaction ratings. However, in the case of adaptive metrics, the difference

Metric	THUIR1		THUIR2		THUIR3		MS	
	ERG	ETG	ERG	ETG	ERG	ETG	ERG	ETG
Prec, $K = 1$	0.397	0.397	0.191	0.191	0.252	0.252	0.153	0.153
Prec, $K = 5$	0.407	0.407	0.307	0.307	0.362	0.362	−0.057	−0.057
Prec, $K = 9$	0.387	0.387	–	–	0.352	0.352	−0.185	−0.185
Prec, $K = 10$	0.351	0.351	–	–	0.341	0.341	−0.215	−0.215
SDCG, $K = 1$	0.397	0.397	0.191	0.191	0.252	0.252	0.153	0.153
SDCG, $K = 5$	0.441	0.441	0.298	0.298	0.365	0.365	0.036	0.036
SDCG, $K = 9$	0.448	0.448	–	–	0.373	0.373	−0.057	−0.057
SDCG, $K = 10$	0.431	0.431	–	–	0.367	0.367	−0.077	−0.077
RBP, $\phi = 0.1$	0.406	0.406	0.206	0.206	0.269	0.269	0.151	0.151
RBP, $\phi = 0.8$	0.435	0.435	0.313	0.313	0.372	0.372	−0.071	−0.071
INSQ, $T = 1$	0.455	0.455	0.279	0.279	0.355	0.355	0.080	0.080
INSQ, $T = 2$	0.451	0.451	0.302	0.302	0.371	0.371	−0.006	−0.006
INSQ, $T = 3$	0.439	0.439	0.309	0.309	0.371	0.371	−0.060	−0.060
INST, $T = 1$	0.451	0.469	0.266	0.277	0.332	0.353	0.112	0.103
INST, $T = 2$	0.444	0.484	0.292	0.308	0.358	0.387	0.012	0.014
INST, $T = 3$	0.421	0.466	0.299	0.314	0.356	0.386	−0.054	−0.046
IFT-C1, $T = 1$	0.437	0.132	0.296	0.297	0.338	0.155	0.117	−0.051
IFT-C1, $T = 2$	0.448	0.342	0.305	0.335	0.358	0.289	−0.128	−0.090
IFT-C1, $T = 3$	0.408	0.399	0.201	0.273	0.359	0.340	−0.221	−0.173
IFT-C2	0.420	0.400	0.297	0.268	0.369	0.339	−0.097	−0.073
IFT, $T = 1$	0.433	0.355	0.283	0.261	0.328	0.269	0.135	0.093
IFT, $T = 2$	0.443	0.456	0.302	0.299	0.348	0.334	−0.088	0.054
IFT, $T = 3$	0.404	0.474	0.228	0.264	0.352	0.357	−0.183	−0.017
gmean	0.423	0.399	0.268	0.275	0.341	0.325	–	–

Table 5.4: Correlation coefficients (Pearson's r) between SERP-level satisfaction ratings and C/W/L-based metric scores. Metric scores are computed using the gain mapping function $g_3(r_i) = (2^{r_i} - 1)/(2^{r_{max}} - 1)$. Blue color represents the three largest coefficients in each column.

between ERG and ETG seems to be affected by the search depth of the corresponding user model, as governed by the T parameter for goal-sensitive metrics. Figure 5.4 (page 191) shows correlation coefficients between query satisfaction ratings and both ERG and ETG versions of INST and IFT, described as a function of T. Here it can be observed that the ETG scores tend to be better correlated, as T becomes larger.

Consider all results from Tables 5.2 (page 187), 5.3 (page 188), 5.4 (page 189), and 5.5 (page 190). The recently proposed adaptive metrics, INST, IFT, and iRBU@K have higher correlations than the traditional ad-hoc metrics Prec@10, AP1@K, and RR in three datasets containing queries with a diverse task complexity (THUIR1, THUIR2, and THUIR3). Two

Metric	THUIR1		THUIR2		THUIR3		MS	
	ERG	ETG	ERG	ETG	ERG	ETG	ERG	ETG
Prec, $K = 1$	0.378	0.378	0.167	0.167	0.239	0.239	0.147	0.147
Prec, $K = 5$	0.427	0.427	0.306	0.306	0.359	0.359	-0.037	-0.037
Prec, $K = 9$	0.391	0.391	–	–	0.348	0.348	-0.156	-0.156
Prec, $K = 10$	0.353	0.353	–	–	0.336	0.336	-0.189	-0.189
SDCG, $K = 1$	0.378	0.378	0.167	0.167	0.239	0.239	0.147	0.147
SDCG, $K = 5$	0.452	0.452	0.291	0.291	0.361	0.361	0.043	0.043
SDCG, $K = 9$	0.454	0.454	–	–	0.370	0.370	-0.046	-0.046
SDCG, $K = 10$	0.436	0.436	–	–	0.364	0.364	-0.068	-0.068
RBP, $\phi = 0.1$	0.388	0.388	0.183	0.183	0.256	0.256	0.145	0.145
RBP, $\phi = 0.8$	0.445	0.445	0.301	0.301	0.370	0.370	-0.058	-0.058
INSQ, $T = 1$	0.454	0.454	0.266	0.266	0.349	0.349	0.079	0.079
INSQ, $T = 2$	0.458	0.458	0.291	0.291	0.368	0.368	-0.000	-0.000
INSQ, $T = 3$	0.448	0.448	0.296	0.296	0.369	0.369	-0.049	-0.049
INST, $T = 1$	0.445	0.445	0.255	0.254	0.325	0.332	0.111	0.101
INST, $T = 2$	0.459	0.479	0.288	0.296	0.361	0.379	0.019	0.019
INST, $T = 3$	0.441	0.469	0.293	0.303	0.361	0.381	-0.043	-0.036
IFT-C1, $T = 1$	0.431	0.045	0.263	0.191	0.325	0.066	0.114	-0.073
IFT-C1, $T = 2$	0.447	0.173	0.319	0.324	0.338	0.217	-0.059	-0.042
IFT-C1, $T = 3$	0.414	0.188	0.260	0.309	0.364	0.257	-0.144	-0.111
IFT-C2	0.335	0.393	0.267	0.226	0.333	0.316	-0.082	-0.044
IFT, $T = 1$	0.422	0.211	0.252	0.156	0.309	0.149	0.129	0.093
IFT, $T = 2$	0.443	0.312	0.308	0.264	0.328	0.237	-0.018	0.100
IFT, $T = 3$	0.420	0.319	0.265	0.273	0.358	0.257	-0.114	0.044
gmean	0.421	0.338	0.261	0.249	0.333	0.284	–	–

Table 5.5: Correlation coefficients (Pearson's r) between SERP-level satisfaction ratings and C/W/L-based metric scores. Metric scores are computed using the gain mapping function $g_4(r_i) = r_i/r_{max}$. Blue color represents the three largest coefficients in each column.

static metrics RBP with $\phi = 0.80$, SDCG@K with $K \in \{5, 9\}$, and INSQ with $T \in \{2, 3\}$ also provide comparable coefficients with those adaptive metrics. However, when the majority of queries are navigational, such in the MS dataset, shallow adaptive metrics, INST, IFT (both with $T = 1$), and iRBU@K with $\phi = 0.1$ are no better than Prec@1, SDCG@1, AP1@K, and RR. Similar behaviour was observed when the non-navigational queries were excluded from the THUIR1 dataset. In this case RBP with $\phi = 0.1$ gives a higher coefficient than RBP with $\phi = 0.8$. When the overall *gmean* scores are considered (see the last row in both Tables 5.4 and 5.5), the exponential gain mapping function $g_3(.)$ tends to yield metric scores that are more correlated with query ratings than the scores generated using the

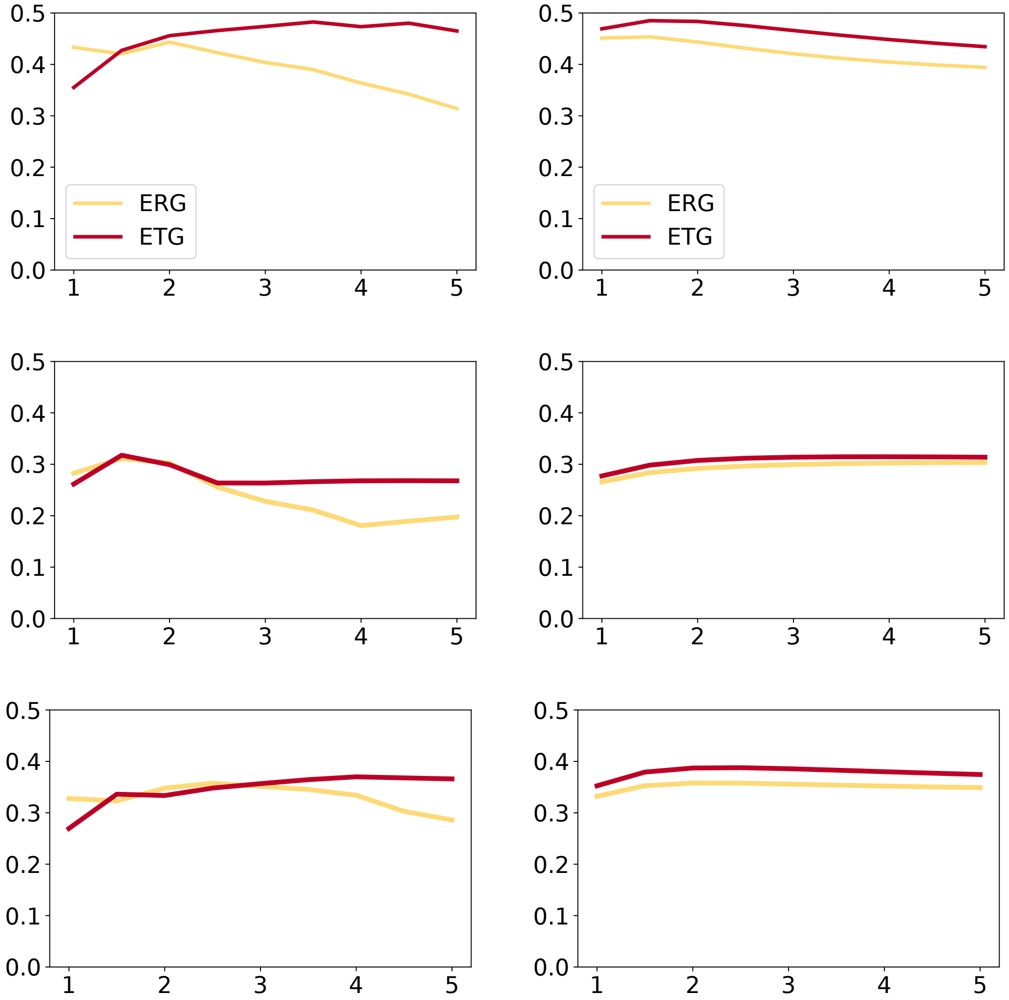


Figure 5.4: Correlation coefficients (y-axis) as a function of $T \in \{1, 1.5, \cdots, 5\}$ (x-axis), and between query-level satisfaction ratings and both ERG and ETG versions of two adaptive metrics: IFT (left column) and INST (right column) for THUIR2 (second row), and THUIR3 (third row).

linear gain mapping function $g_4(.)$.

Particular attention has to be given to the iRBU@K. This adaptive metric is one of the three metrics with the highest correlation coefficients with query satisfaction ratings for all four datasets, and yields the highest coefficient in the THUIR1 dataset with $r \approx 0.50$. Interestingly, Sakai and Zeng [182] demonstrate that iRBU@K with $\phi = 0.99$ also has strong agreement with user preferences (Kendall's $\tau \approx 0.80$). An investigation of why iRBU@K correlates relatively well with query satisfaction and user preferences is left for future work.

Comparing Correlation Coefficients. Hotteling's t test [90] is employed to compare two correlation coefficients with one variable in common, and is computed as follows:

$$t = \frac{(r_{jk} - r_{jh}) \cdot \sqrt{(n-3) \cdot (1 + r_{kh})}}{\sqrt{2 \cdot (1 + 2r_{jk}r_{jh}r_{kh} - r_{jk}^2 - r_{jh}^2 - r_{kh}^2)}} \,,$$

where t follows a Student's t distribution with $n-3$ degrees of freedom; n is the number of points in the data for computing a correlation coefficient; r_{jk} is the correlation coefficient between user-reported ratings and scores from metric k; r_{jh} is the correlation coefficient between user ratings and scores from metric h; and r_{kh} is the correlation coefficient between both metric scores. A tool implemented by Diedenhofen and Musch [62] is used to compare the coefficents in Tables 5.2, 5.3, 5.4, and 5.5. [1].

Consider a significance level (α) of 0.01 with Bonferroni correction and a two-sided Hotelling's t test. In Tables 5.2, 5.3, and 5.4, the highest correlation for the THUIR1 dataset is iRBU@K with $\phi = 0.99$, which is significantly better than Prec@10 ($p = 0.0013$), but not significant relative to RR ($p = 0.0386$), Prec@1 ($p = 0.0636$), ERR ($p = 0.2328$), and AP1@K ($p = 0.0707$); and an adaptive metric, INST (ETG) with $T = 2$, also significantly outperforms Prec@1 ($p = 0.0161$), Prec@10 ($p = 0.0003$), but is not significantly better than RR ($p = 0.0181$), ERR ($p = 0.1333$) and AP1@K ($p = 0.0537$); and the highest correlation for the THUIR2 dataset is D-BPM, which significantly outperforms all of Prec@1, RR, and AP1@K ($p < 0.01$ in all three cases). Table 5.6 summarises several p values computed from THUIR3 – the dataset with the lowest residual values and the highest number of points among the others. The INST (ETG) with $T = 2$, RBP with $\phi = 0.80$, and SDCG@9 are all significantly better than the five conventional metrics: Prec@1, Prec@10, ERR, RR, and AP1@K. We see strong evidence from two datasets that adaptive metric INST (ETG) with $T = 2$ is significantly better correlated with query-level satisfaction ratings than the traditional Prec@10.

Surrogates for Satisfaction Ratings. The fact that negative coefficients dominate the MS columns in Tables 5.4 and 5.5 suggests that query non-reformulation rates are a poor surrogate for query-level satisfaction ratings. The THUIR2 and THUIR3 datasets are employed to evaluate the use of observable query-level user actions for the approximation of the satisfaction ratings. The query non-reformulation rate can be represented as a binary indicator metric nreform which is 1 if the corresponding query is the last one in the session

[1] The package for R programming language is available at `http://comparingcorrelations.org/`

	iRBU@K ($\phi = 0.99$)	INST(ETG) ($T = 2$)	RBP ($\phi = 0.80$)	SDCG@9
Prec@1	0.003	0.000	0.000	0.000
Prec@10	0.729	0.001	0.000	0.000
ERR	0.518	0.000	0.003	0.002
RR	0.523	0.000	0.004	0.003
AP1@K	0.628	0.000	0.003	0.003

Table 5.6: Resultant p values computed using Hotelling's t test for comparing correlation coefficients between each of four metrics (column header) and five conventional ad-hoc metrics (first column in each row) in the THUIR3 dataset. A significance level (α) of 0.01 is employed with Bonferroni correction. Blue color represents a significant difference.

and 0 if not. Other click-based signals are also tested. Suppose $\overrightarrow{c} = \langle c_1, c_2, c_3, \cdots, c_{n(\overrightarrow{c})} \rangle$ denotes a sequence of clicked ranks observed for a SERP. The first click-based signal tested is the number of distinct items clicked ($\mathsf{numclick} = |\{c \mid c \in \overrightarrow{c}\}|$). The second signal is the precision at lowest click (PLC), computed as:

$$\mathsf{PLC}(\overrightarrow{c}) = \begin{cases} n(\overrightarrow{c})/c_{n(\overrightarrow{c})} & \text{if } n(\overrightarrow{c}) > 0 \\ 0 & \text{if } n(\overrightarrow{c}) = 0\,. \end{cases}$$

The other signals are the maximum, minimum, and mean reciprocal ranks of clicked items (minRC, maxRC, meanRC). For example, maxRC is computed as:

$$\mathsf{maxRC}(\overrightarrow{c}) = \begin{cases} \max(\{1/c \mid c \in \overrightarrow{c}\}) & \text{if } n(\overrightarrow{c}) > 0 \\ 0 & \text{if } n(\overrightarrow{c}) = 0\,. \end{cases}$$

These click-based signals were originally proposed by Radlinski et al. [167], and then employed by Chapelle et al. [44] to meta-evaluate ERR. Table 5.7 shows the correlation between these action-based signals and query-level satisfaction ratings, and suggests that the three click-based metrics PLC, maxRC, and meanRC are better than nreform in the two datasets. These differences are significant in the THUIR2 dataset (Hotelling's t test, $p < 0.01$), but not significant in the THUIR3 dataset (Hotelling's t test, $p > 0.10$).

Consistency Across Users. It is also possible to treat each user differently and compute distributions of correlation coefficients. In the THUIR1 dataset, a group of 25 users (user IDs 71–80, 82–95, and 97) inspected and evaluated the same set of 21 SERPs. The graphs in Figures 5.5 (page 195) and 5.6 (page 196) represent distributions of correlation coefficients (computed using kernel density estimation) between metric scores (ERG) and

	nreform	numclick	PLC	minRC	maxRC	meanRC
THUIR2	0.246	0.377	0.392	0.261	0.397	0.358
THUIR3	0.298	0.220	0.317	0.235	0.339	0.327

Table 5.7: Correlation between signals based on user actions (clicks and reformulations) and query-level satisfaction ratings. The values listed are Pearson's correlation coefficients between the signals and query-level satisfaction ratings.

satisfaction ratings across those 25 users, in each case computing the correlation coefficient for a user from their 21 data pairs. The distributions for IFT with $T = 2$, INST (ETG) with $T = 2$, and iRBU@K with $\phi = 0.99$ are more skewed toward the high end than the other three traditional metrics: Prec@10, RR, and ERR (paired t test, $p < 0.01$ in all cases). This indicates that user are more inclined to agree with the adaptive metrics than with the traditional ones. The other metrics whose distributions are more skewed to the right compared to those three traditional metrics are IFT-C2 and RBP with $\phi = 0.80$ (paired t test, $p < 0.025$ in all cases). On the other hand, the distributions for Prec@1 and SDCG@1 are skewed to the left (paired t test, $p < 0.01$ in all cases except the relative difference to Prec@10 with $p \approx 0.10$).

To conclude, this section has shown evidence that scores from adaptive metrics, INST (ETG) with $T = 2$ and iRBU@K with $\phi = 0.99$, have a better correlation with query-level satisfaction ratings than those from metrics that depend on R (AP1@K, AP2@K, QM1@K, QM2@K, and NDCG@K) and those from two traditional metrics, Prec@10 and RR. Other static metrics, RBP with $\phi = 0.80$, SDCG@9, DCG@9, and INSQ, are comparable with those two adaptive metrics in terms of correlation with query-level satisfaction. The correlation also tended to be confounded by the query taxonomy (navigational or non-navigational). When queries are mostly navigational, such as those in the MS dataset, shallow metrics, such as RBP with $\phi = 0.10$, Prec@1, and SDCG@1 have a better correlation with satisfaction than their deeper versions. Other key findings are that the exponential gain mapping function leads to a better correlation with user satisfaction ratings than the linear function, and that the adaptive ETG metrics tended to be better correlated than the ERG metrics for a high-T search.

5.4.2 Session-Level Satisfaction

Three of the datasets (J&A, THUIR2, and THUIR3) contain sequences of query reformulations observed from users as well as their corresponding session-level satisfaction ratings.

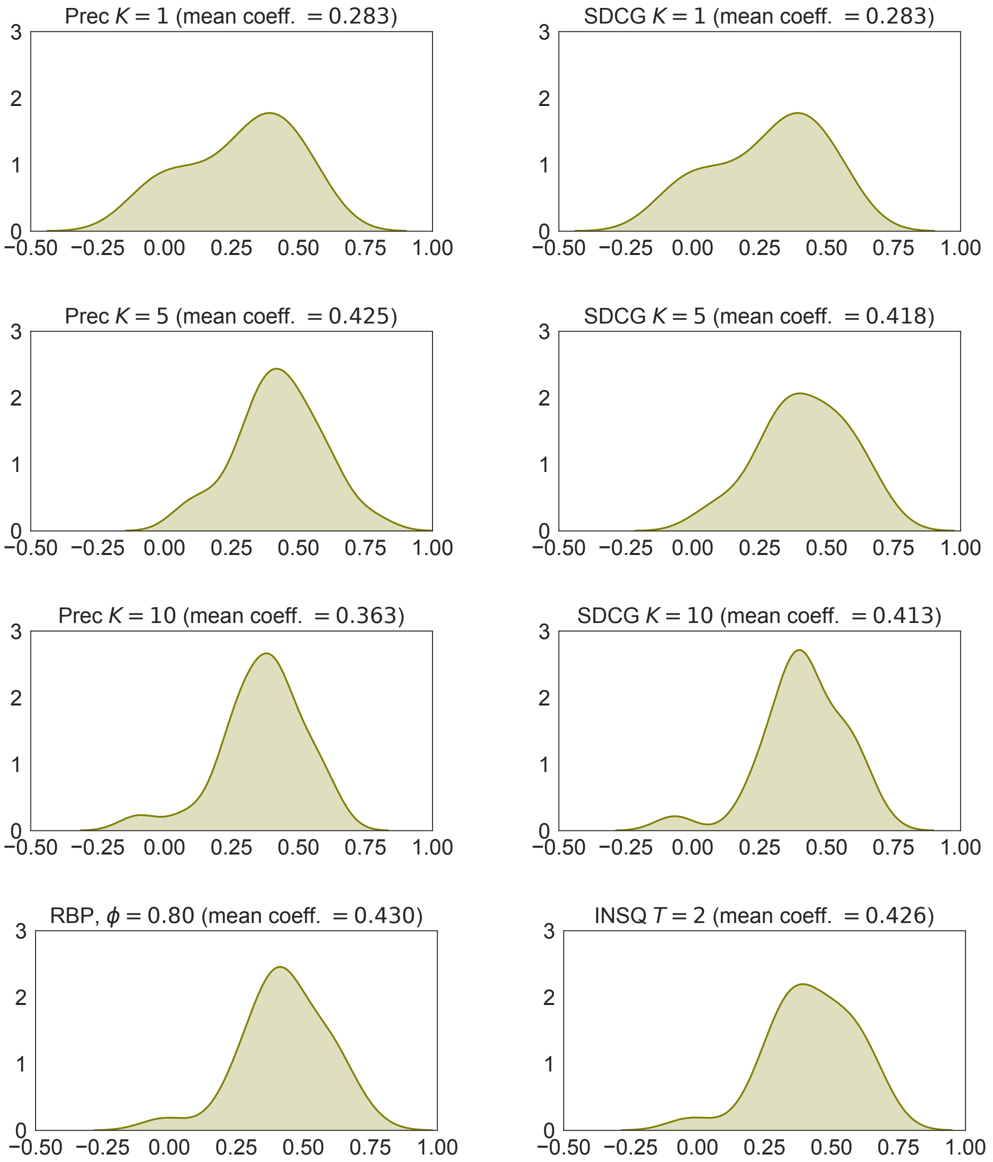


Figure 5.5: Distributions of correlation coefficients (as computed using kernel density estimator) between satisfaction ratings and query scores generated from eight static metrics, computed across 25 users, each of which evaluated the same set of 21 SERPs. Correlation coefficient (Pearson's r) is denoted on x-axis, while density is on y-axis.

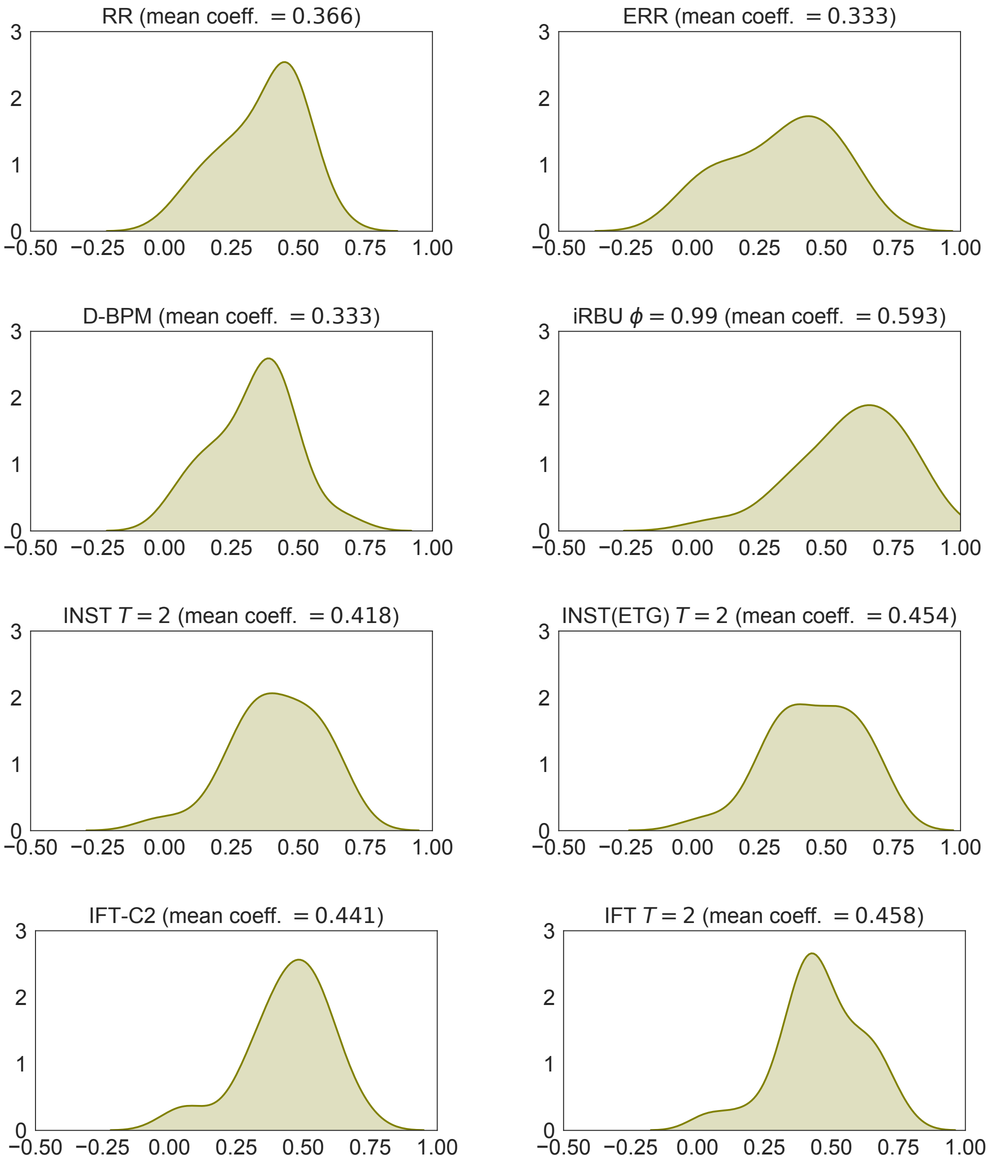


Figure 5.6: Distributions of correlation coefficients (as computed using kernel density estimator) between satisfaction ratings and query scores generated from eight adaptive metrics, computed across 25 users, each of which evaluated the same set of 21 SERPs. Correlation coefficient (Pearson's r) is denoted on x-axis, while density is on y-axis.

Chapter 4 has explored various query-to-session aggregation methods, investigating the relationship between session satisfaction ratings and individual query scores in the session. This section utilises several aggregation functions described in Chapter 4 to meta-evaluate session scores generated from a wide range of SERP-level metrics. In addition to the simple aggregation functions, such as mean, max, and last, the method based on a weighted mean of positional- and quality-based factors (denoted as wam) is also employed (see Equation 4.18 on page 165). Section 4.8.1 has shown that wam is better than the other three aggregation methods, u-shaped function (see Equation 4.20 on page 165), the exponential weighting method [139] (denoted as Liu), and memory model [242] (denoted as Zhang).

In the case of the J&A dataset, the relevance of a document is judged with respect to a task description, not with respect to a query. Even though each SERP is truncated at depth 9, the set of documents beyond depth 9 that was not displayed on that SERP is still available (see Figure 5.2 on page 182). This allows for a third alternative of computing the normalisation factor for average precision at depth K: $Z = \min(K, \hat{R})$, where $\hat{R}$ is the total number of relevant documents for the task, whether or not they were included in the SERP [227]. Sakai and Zeng [182] also use the same normalisation factor for Q-Measure at depth K. Let these two versions of average precision and Q-Measure be denoted by, respectively, AP3@K and QM3@K. Time-biased gain (TBG) is also computed using the implementation of Jiang and Allan [105], which is different from the original version [195]. Jiang and Allan suggest that the time to read a document be defined by a function of the relevance of the document; while the original version by Smucker and Clarke [195] computes this quantity using the length of the document, measured in the number of words.

Table 5.8 (page 198), Table 5.9 (page 199), and Table 5.10 (page 200) show correlation coefficients between session-level satisfaction ratings and aggregated metric scores using sessions in the J&A, THUIR2, and THUIR3 datasets, with blue color representing five metrics with the greatest geometric mean values on the rightmost column. For THUIR2 and THUIR3, the max column is excluded from the tables because the majority of the coefficients in this column are very low, and some of them are negative. Gains are computed from relevance grades using the exponential gain mapping function $g_3(r_i) = (2^{r_i} - 1)/(2^{r_{max}} - 1)$ for all metrics except AP1@K, AP3@K, RR, and ERR. Section 5.4.1 has provided an empirical evidence that the exponential gain function is better than the linear counterpart. For AP1@K, AP3@K, and RR, this experiment applies the binary gain mapping $g_2(.)$, which returns 1 if the corresponding document is fully relevant, or otherwise yields 0.

	mean	max	min	first	last	u-shape	Liu	Zhang	wam	gmean
Prec, $K = 1$	0.263	0.077	0.266	0.110	0.225	0.235	0.247	0.241	0.231	0.196
Prec, $K = 5$	0.429	0.308	0.391	0.312	0.443	0.445	0.438	0.437	0.454	0.402
SDCG, $K = 1$	0.263	0.077	0.266	0.110	0.225	0.235	0.247	0.241	0.231	0.196
SDCG, $K = 5$	0.423	0.318	0.381	0.291	0.435	0.433	0.427	0.425	0.449	0.394
RBP, $\phi = 0.10$	0.290	0.113	0.278	0.137	0.255	0.268	0.276	0.271	0.293	0.231
RBP, $\phi = 0.80$	0.409	0.315	0.391	0.297	0.427	0.422	0.415	0.413	0.451	0.390
INSQ, $T = 2$	0.409	0.319	0.382	0.286	0.422	0.419	0.413	0.410	0.449	0.386
INSQ, $T = 3$	0.408	0.315	0.388	0.295	0.424	0.419	0.413	0.411	0.448	0.388
INST, $T = 3$	0.396	0.313	0.373	0.291	0.414	0.409	0.403	0.401	0.430	0.378
INST (ETG), $T = 3$	0.414	0.320	0.389	0.291	0.426	0.422	0.416	0.414	0.458	0.391
IFT-C1, $T = 3$	0.327	0.265	0.326	0.262	0.333	0.336	0.332	0.332	0.348	0.316
IFT-C1 (ETG), $T = 3$	0.324	0.219	0.336	0.204	0.294	0.293	0.308	0.309	0.347	0.289
IFT-C2	0.398	0.285	0.380	0.230	0.358	0.373	0.385	0.383	0.443	0.353
IFT-C2 (ETG)	0.348	0.240	0.349	0.254	0.372	0.370	0.358	0.354	0.388	0.333
IFT, $T = 3$	0.334	0.242	0.344	0.253	0.344	0.343	0.338	0.338	0.362	0.319
IFT (ETG), $T = 3$	0.313	0.161	0.315	0.184	0.324	0.317	0.311	0.305	0.362	0.279
AP1@K, $K = 9$	0.487	0.268	0.387	0.296	0.430	0.460	0.471	0.471	0.491	0.409
AP3@K, $K = 9$	0.329	0.268	0.303	0.267	0.359	0.344	0.337	0.336	0.355	0.320
QM3@K, $K = 9$, $\beta = 10$	0.348	0.286	0.335	0.261	0.361	0.364	0.357	0.355	0.376	0.336
NDCG	0.353	0.269	0.346	0.265	0.372	0.362	0.355	0.353	0.388	0.338
S-BPM	0.316	0.220	0.347	0.267	0.321	0.320	0.312	0.313	0.352	0.305
D-BPM	0.347	0.273	0.352	0.240	0.323	0.331	0.337	0.340	0.370	0.321
ERR	0.385	0.186	0.357	0.197	0.339	0.358	0.366	0.362	0.400	0.317
RR	0.392	0.169	0.345	0.240	0.334	0.358	0.367	0.369	0.368	0.318
iRBU, $\phi = 0.99$	0.337	0.306	0.329	0.140	0.217	0.256	0.301	0.298	0.317	0.269
TBG	0.363	0.290	0.370	0.270	0.349	0.362	0.364	0.363	0.371	0.343
gmean	0.358	0.232	0.345	0.231	0.344	0.350	0.353	0.351	0.376	

Table 5.8: Correlation coefficients (Pearson's r) between session-level satisfaction ratings and metric scores for set of sessions in the J&A dataset. Blue color is based on the row *gmean*, representing five metrics with the highest geometric mean values on the last column.

	mean	min	first	last	u-shape	Liu	Zhang	wam	gmean
Prec, $K = 1$	0.228	0.262	0.116	0.099	0.194	0.227	0.228	0.296	0.194
Prec, $K = 5$	0.354	0.440	0.230	0.296	0.341	0.354	0.354	0.432	0.344
SDCG, $K = 1$	0.228	0.262	0.116	0.099	0.194	0.227	0.228	0.296	0.194
SDCG, $K = 5$	0.335	0.411	0.194	0.277	0.324	0.335	0.335	0.397	0.319
RBP, $\phi = 0.10$	0.236	0.285	0.114	0.122	0.207	0.235	0.236	0.269	0.203
RBP, $\phi = 0.80$	0.358	0.440	0.185	0.288	0.341	0.358	0.358	0.434	0.335
INSQ, $T = 2$	0.340	0.422	0.166	0.271	0.324	0.340	0.340	0.411	0.316
INSQ, $T = 3$	0.351	0.435	0.169	0.280	0.333	0.351	0.351	0.428	0.326
INST, $T = 3$	0.338	0.422	0.181	0.282	0.327	0.338	0.338	0.414	0.321
INST (ETG), $T = 3$	0.356	0.438	0.177	0.279	0.337	0.356	0.356	0.430	0.330
IFT-C1, $T = 3$	0.265	0.272	0.037	0.261	0.264	0.266	0.265	0.269	0.208
IFT-C1 (ETG), $T = 3$	0.346	0.441	0.023	0.232	0.299	0.346	0.346	0.433	0.244
IFT-C2	0.375	0.458	0.050	0.221	0.310	0.375	0.375	0.446	0.279
IFT-C2 (ETG)	0.330	0.422	0.053	0.276	0.312	0.331	0.330	0.415	0.271
IFT, $T = 3$	0.292	0.337	0.099	0.266	0.285	0.292	0.292	0.323	0.259
IFT (ETG), $T = 3$	0.338	0.425	0.019	0.265	0.309	0.339	0.338	0.418	0.239
AP1@K, $K = 10$	0.237	0.278	0.218	0.131	0.210	0.235	0.237	0.175	0.210
S-BPM	0.386	0.460	0.180	0.238	0.330	0.384	0.386	0.453	0.338
D-BPM	0.434	0.481	0.340	0.200	0.363	0.432	0.434	0.474	0.383
ERR	0.296	0.382	0.134	0.171	0.258	0.296	0.296	0.362	0.260
RR	0.245	0.268	0.220	0.130	0.214	0.244	0.245	0.181	0.214
iRBU, $\phi = 0.99$	0.385	0.441	0.183	0.167	0.295	0.384	0.385	0.442	0.316
gmean	0.315	0.377	0.117	0.208	0.284	0.315	0.315	0.360	

Table 5.9: Correlation coefficients (Pearson's r) between session-level satisfaction ratings and metric scores for set of sessions in the THUIR2 dataset. Blue color is based on the row *gmean*, representing five metrics with the highest geometric mean values on the last column. The *max* column is not included, since it contains negative coefficients.

	mean	min	first	last	u-shape	Liu	Zhang	wam	gmean
Prec, $K = 1$	0.188	0.262	0.154	0.143	0.178	0.188	0.188	0.257	0.191
Prec, $K = 5$	0.260	0.266	0.247	0.205	0.259	0.260	0.260	0.344	0.260
SDCG, $K = 1$	0.188	0.262	0.154	0.143	0.178	0.188	0.188	0.257	0.191
SDCG, $K = 5$	0.257	0.283	0.237	0.213	0.258	0.257	0.257	0.351	0.261
RBP, $\phi = 0.10$	0.199	0.271	0.162	0.156	0.191	0.199	0.199	0.268	0.202
RBP, $\phi = 0.80$	0.270	0.278	0.248	0.235	0.278	0.270	0.270	0.351	0.273
INSQ, $T = 2$	0.267	0.286	0.243	0.233	0.274	0.267	0.267	0.354	0.272
INSQ, $T = 3$	0.271	0.280	0.249	0.235	0.279	0.271	0.271	0.351	0.274
INST, $T = 3$	0.263	0.278	0.251	0.223	0.267	0.263	0.263	0.339	0.267
INST (ETG), $T = 3$	0.271	0.284	0.238	0.248	0.285	0.271	0.271	0.365	0.277
IFT-C1, $T = 3$	0.277	0.278	0.268	0.231	0.278	0.277	0.277	0.324	0.275
IFT-C1 (ETG), $T = 3$	0.242	0.264	0.155	0.264	0.292	0.242	0.242	0.359	0.251
IFT-C2	0.222	0.214	0.160	0.250	0.268	0.222	0.222	0.344	0.233
IFT-C2 (ETG)	0.281	0.264	0.255	0.223	0.281	0.281	0.281	0.308	0.270
IFT, $T = 3$	0.258	0.252	0.249	0.224	0.262	0.258	0.258	0.302	0.257
IFT (ETG), $T = 3$	0.283	0.298	0.195	0.270	0.302	0.283	0.283	0.369	0.282
AP1@K, $K = 10$	0.216	0.334	0.199	0.165	0.209	0.216	0.216	0.246	0.221
S-BPM	0.165	0.230	0.119	0.198	0.202	0.165	0.165	0.318	0.188
D-BPM	0.164	0.223	0.113	0.168	0.191	0.164	0.164	0.335	0.182
ERR	0.216	0.276	0.163	0.191	0.220	0.216	0.216	0.322	0.223
RR	0.237	0.323	0.198	0.187	0.231	0.237	0.237	0.244	0.234
iRBU, $\phi = 0.99$	0.072	0.141	0.063	0.189	0.119	0.072	0.072	0.253	0.108
gmean	0.222	0.262	0.187	0.205	0.235	0.222	0.222	0.313	

Table 5.10: Correlation coefficients (Pearson's r) between session-level satisfaction ratings and metric scores for set of sessions in the THUIR3 dataset. Blue color is based on the row *gmean*, representing five metrics with the highest geometric mean values on the last column. The *max* column is not included, since it contains negative coefficients.

To draw some general patterns across metrics and across the aggregation techniques, each row and column in Tables 5.8, 5.9, and 5.10 has a geometric mean (*gmean*) associated with it to provide an overall perspective on that row or column. This is particularly useful for the J&A datasets, since the differences between pairs of coefficients are not significant under the Hotelling's t test. Recall that the J&A dataset only has 80 data points, which is the lowest among all datasets used in this study. Looking down the final columns in all three tables indicate that RBP with $\phi = 0.8$ and INST (ETG version) with $T = 3$ are consistently in the list of five best metrics across all datasets, providing better correlations when combined across the suite of aggregation columns compared to metrics such as ERR, RR, iRBU@K with $\phi = 0.99$, Prec@K with $K = 1$, and SDCG@K with $K = 1$.

In the J&A dataset, metrics that depends on $\hat{R}$ (the estimated number of relevant documents for a task) such as NDCG, QM3@K and AP3@K (both with $Z = \max(K, \hat{R})$) are no better than INST (ETG version) with $T = 3$, TBG, INSQ with $T = 3$, Prec@K with $K = 5$, and SDCG@K with $K = 5$. However, when the normalisation factor $Z = \sum_{i=1}^{K} r_i$ is employed, average precision at depth K performs the best among the other metrics. In the case of the THUIR2 dataset, D-BPM provides the highest *gmean* score. A simple metric, Prec@K with $K = 5$, has better correlations than more complex ones such as, ERR, IFT, and iRBU@K with $\phi = 0.99$. Finally, IFT (ETG) with $T = 3$ performs poorly on THUIR2, but gives the highest correlation coefficient on THUIR3.

The final rows in the three tables, each of which records the geometric means computed over the values in the columns above them provide several further observations: that the aggregation method based on a weighted mean of position- and quality-based factors (wam) is better than two recently proposed techniques, Zhang [242] and Liu [139], and provides the highest correlation coefficients with session ratings in the J&A and THUIR3 datasets; that taking the average across the query scores in a session appears to be comparable with the two methods, Zhang and Liu; that min appears to be better than max; and that the last query in each session seems to be more influential than the first one.

Section Summary. This section has explored correlations between various types of offline metrics and satisfaction ratings using four datasets constructed from lab-based experiments and one dataset from a commercial search engine. It has been shown that the correlation between metric and user satisfaction is confounded by the query taxonomy, such as navigational or non-navigational. In the MS dataset, where most of the queries are navigational, shallow metrics, such as RR, Prec@K with $K = 1$, SDCG@K with $K = 1$, RBP with $\phi = 0.1$, and iRBU@K with $\phi = 0.10$ correlate better with satisfaction than their deeper versions. A similar phenomenon is also observed in the subset of THUIR1, in which all queries are navigational. In a collection of data with a diverse task complexity, adaptive metrics, such as INST (ETG version with $T = 2$) and iRBU@K with $\phi = 0.99$ tend to be better than the other metrics, such as Prec@10, RR, QM1@K, AP1@K, and NDCG@K. Static metrics, such as RBP with $\phi = 0.80$, SDCG@9, DCG@9, and INSQ also correlate relatively well with query-level satisfaction ratings. This suggests that metrics should be parameterised in accordance with the query taxonomy, or with the user's goal, controlling the expected number of items that are inspected by the user. That is, the argument of Moffat et al. [153, 155] in regard to parameterisation of metrics according to the complexity of the information need can be seen to have empirical support.

At the session-level, we use several aggregation functions described in Chapter 4 to compute correlation coefficients between session scores and session ratings. The results again show that RBP with $\phi = 0.80$ and adaptive metrics, such as INST, perform relatively well. In the J&A dataset, the metrics that depend on the knowledge of R, such as QM3@K and NDCG do not perform any better than R-agnostic metrics, such as INSQ and INST.

Other key observations in this section are that the exponential gain mapping function yields metric scores that have a better relationship with satisfaction than the scores generated using the linear gain mapping function; that scores generated by the ETG versions of adaptive metrics tend to be better correlated, as T becomes larger; and that click-based actions, such as *precision at lowest click* and *maximum reciprocal clicked rank*, provide a better surrogate for query-level satisfaction ratings than the query reformulation binary indicator.

5.5 User Models and User Behaviour

Section 5.4 explored correlation coefficients between scores from a wide range of metrics and both query- and session- satisfaction ratings (link 5 in Figure 5.1 on page 177). This section investigates the dual of that relationship – the relationship between user models (corresponding to metrics in the C/W/L framework), and observed user behaviour (link 6 in Figure 5.1). Here, we propose a method for measuring user model accuracy from the perspective of the C/W/L framework, and investigate the effect of adaptivity for improving metric accuracy.

5.5.1 Measuring User Model Accuracy

Model accuracy is measured by calculating the extent to which the observed user behaviour matches the behaviour predicted by the model. With the C/W/L framework, this is done by computing the distances between distributions $C(\cdot)$, $W(\cdot)$, and $L(\cdot)$ for each given metric, and their corresponding observed distributions, denoted by $\hat{C}(\cdot)$, $\hat{W}(\cdot)$, and $\hat{L}(\cdot)$. In the model, $C(\cdot)$, $W(\cdot)$, and $L(\cdot)$ are interrelated, and can be computed from each other under the assumption that the user sequentially examines search results in each SERP. In the user observation data, that interrelationship cannot be assumed for $\hat{C}(\cdot)$, $\hat{W}(\cdot)$, and $\hat{L}(\cdot)$, since inspection traces observed from users might not be sequential. Hence, it make sense for each of them to be independently estimated from interaction logs.

Suppose a dataset covers a set of user IDs $U = \{u_1, u_2, \ldots, u_{|U|}\}$, and each user u_i is associated with a set of *view sequences* $\mathcal{V}(u_i) = \{\overrightarrow{v_1}, \overrightarrow{v_2}, \ldots, \overrightarrow{v}_{|\mathcal{V}(u_i)|}\}$, where each view sequence $\overrightarrow{v_i}$ is an ordered sequence of rank positions inspected for a particular SERP, and is a sub-sequence of an *action sequence* (see Section 3.2.1 on page 76). Ideally, the view sequences are eye-fixation sequences obtained from an eye-tracking experiment, or are recorded from a particular Web interface, such as impression sequences in the `Seek.com` dataset used in Chapters 3 and 4[2].

Evaluating W. The observed $\hat{W}(i)$ is estimated by maximising the data likelihood. That is, $\hat{W}(i)$ is computed as:

$$\hat{W}(i) = \frac{\sum_{u \in U} \sum_{\overrightarrow{v} \in \mathcal{V}(u)} \mathbf{I}(i \in \overrightarrow{v})}{\sum_{u \in U} \sum_{\overrightarrow{v} \in \mathcal{V}(u)} D(\overrightarrow{v})} ,$$

where $D(\overrightarrow{v})$ is the number of distinct elements in $\overrightarrow{v}$, and $\mathbf{I}(A)$ is an indicator function that returns 1 if event A occurs, and 0 otherwise. To give an illustration, consider the following sets of view sequences observed from three different users, $U = \{u_1, u_2, u_3\}$:

$$\begin{aligned} \mathcal{V}(u_1) &= \{\langle 1,2,1,3\rangle, \langle 1,3\rangle, \langle 1\rangle, \langle 1,2,1\rangle\} , \\ \mathcal{V}(u_2) &= \{\langle 1,4,2\rangle, \langle 1,2,3,4\rangle\} , \\ \mathcal{V}(u_3) &= \{\langle 1,2,1,4,6\rangle, \langle 2,3,5\rangle, \langle 1\rangle, \langle 1\rangle\} . \end{aligned} \tag{5.1}$$

Using these collection of view sequences, the summation over all numbers of distinct items across all view sequences is $\sum_{u \in U} \sum_{\overrightarrow{v} \in \mathcal{V}(u)} D(\overrightarrow{v}) = 24$. Hence, $\hat{W}(i)$ values for $i > 0$

[2] The "view sequence" is a more general term than the "impression sequence" in the `Seek.com` data and than the "eye-fixation sequence" captured by an eye-tracking tool.

are computed as follows: $\hat{W}(1) = 9/24$, $\hat{W}(2) = 6/24$, $\hat{W}(3) = 4/24$, $\hat{W}(4) = 3/24$, $\hat{W}(5) = 1/24$, $\hat{W}(6) = 1/24$, and $\hat{W}(i) = 0$ for $i > 6$.

To measure the closeness between the $W(\cdot)$ generated from a model of a particular metric and the observed $\hat{W}(\cdot)$ estimated from a dataset, this study employs a mean squared error (MSE) function:

$$\mathsf{MSE}(\hat{\mathbf{W}}, \mathbf{W}) = \frac{1}{N} \cdot \sum_{i=1}^{N} \left[W(i) - \hat{W}(i) \right]^2 ,$$

where N is the SERP length. The closer the $\mathsf{MSE}(\hat{\mathbf{W}}, \mathbf{W})$ value is to zero, the stronger the empirical evidence for that $W(\cdot)$ formulation.

Evaluating L. Assuming that users sequentially scan down the ranking, the position of the last item inspected is also the deepest rank position examined. A maximum likelihood estimator for $\hat{L}(i)$ is then:

$$\hat{L}(i) = \frac{\sum_{u \in U} \sum_{\overrightarrow{v} \in \mathcal{V}(u)} \mathbf{I}(i = \max\{\overrightarrow{v}\})}{\sum_{u \in U} |\mathcal{V}(u)|} .$$

Considering again the example described in Equation 5.1, values of $\hat{L}(i)$ for $i > 0$ are determined as follows: $\hat{L}(1) = 3/10$, $\hat{L}(2) = 1/10$, $\hat{L}(3) = 2/10$, $\hat{L}(4) = 2/10$, $\hat{L}(5) = 1/10$, $\hat{L}(6) = 1/10$, and $\hat{L}(i) = 0$ for $i > 6$. To compute the distance between the model $L(\cdot)$ and the observed distribution $\hat{L}(\cdot)$, use is again made of $\mathsf{MSE}(\hat{\mathbf{L}}, \mathbf{L})$.

Evaluating C. In contrast to $W(\cdot)$ and $L(\cdot)$, which are both probability distributions (that is, $\sum_{i=1}^{\infty} W(i) = \sum_{i=1}^{\infty} L(i) = 1$), the continuation function $C(\cdot)$ is a set of independent values between zero and one. Chapter 3 describes three heuristics for computing empirical $\hat{C}(\cdot)$ from view sequences across all users and queries. One of those heuristics is the rule G, which states that a *continuation* is deemed to occur at rank i if an examination at rank i is followed by another at a higher ranking postion. An empirical $\hat{C}(i)$ function is then determined by aggregating the continuation indicators over all view sequences. The distance between $C(\cdot)$ and $\hat{C}(\cdot)$ is measured using a *weighted* mean squared error $\mathsf{WMSE}(\hat{\mathbf{C}}, \mathbf{C})$, where the value at rank i is weighted by the relative frequency with which documents at rank i were inspected. This weighting scheme is required, since $C(\cdot)$ itself is not a probability distribution.

Model-Generated $C(\cdot)$ for Adaptive Metrics. Recall that in adaptive user models $C(\cdot)$ is affected by the gain vector, which means that different gain vectors lead to different distributions for $C(i)$, $W(i)$, and $L(i)$. To allow for comparison between $C(\cdot)$, $W(\cdot)$, and $L(\cdot)$ and their observed distributions in a dataset with a diverse set of gain vectors, representative model-generated distributions are necessary. Of the available choices, it was felt appropriate to average their values across all gain vectors, $\mathbf{\Gamma} = \{\overrightarrow{r_1}, \overrightarrow{r_2}, \ldots\}$, in the dataset. As an instance, representative values of $C(i)$ for INST, an adaptive metric, are computed as follows:

$$C_{\mathsf{INST}}(i) = \frac{1}{|\mathbf{\Gamma}|} \cdot \sum_{\overrightarrow{r} \in \mathbf{\Gamma}} C_{\mathsf{INST}}(i, \overrightarrow{r}) \,,$$

where $C_{\mathsf{INST}}(i, \overrightarrow{r})$ is the INST continuation probability at rank i with respect to the relevance vector $\overrightarrow{r}$.

5.5.2 Measuring Accuracy Using View Distributions

The J&A dataset (see Table 5.1 on page 181) contains view sequences captured by an eye-tracking tool, but the other four datasets do not. In the absence of view sequences, Chapter 3 (Section 3.5 on page 3.5) has proposed a model for inferring view distributions from click sequences by assuming that the user inspects the ranking one-by-one from top to bottom, and that if they click the document at rank i, they have seen ranks 1 to i previously. Suppose that $V(i \mid u, q)$ is the probability that user u views the item listed at rank i for query q. Using click data, $V(i \mid u, q)$ is estimated as follows:

$$\hat{V}(i \mid u, q) = \begin{cases} 1 & \text{if } i \leq DC(u,q) \\ \exp\left[(DC(u,q) - i)/\omega\right] & \text{if } i > DC(u,q)\,, \end{cases} \tag{5.2}$$

where ω is the persistence beyond the deepest click, and needs to be estimated from the data; $DC(u,q)$ is the deepest rank position clicked; and $NC(u,q)$ is the number of distinct items clicked. Let empirical distributions of $C(\cdot)$, $W(\cdot)$, and $L(\cdot)$ that are estimated using view distributions be denoted by $\hat{C}(i; \hat{\mathbf{V}})$, $\hat{W}(i; \hat{\mathbf{V}})$, and $\hat{L}(i; \hat{\mathbf{V}})$.

Suppose that each user u_i can also be thought of as having an association with a set of queries $Q(u_i) = \{q_1, q_2, \ldots, q_{|Q(u_i)|}\}$. Considering $V(i \mid u, q)$ as an *expected count*, empirical continuation probabilities can be computed as follows:

$$\hat{C}(i;\hat{\mathbf{V}}) = \frac{\sum_{u\in U}\sum_{q\in Q(u)}\hat{V}(i+1 \mid u,q)}{\sum_{u\in U}\sum_{q\in Q(u)}\hat{V}(i \mid u,q)}.$$

Similarly, $\hat{W}(i)$ is the fraction of attention paid to rank i, and is determined as follows:

$$\hat{W}(i;\hat{\mathbf{V}}) = \frac{\sum_{u\in U}\sum_{q\in Q(u)}\hat{V}(i \mid u,q)}{\sum_{u\in U}\sum_{q\in Q(u)}\sum_{j=1}^{N}\hat{V}(j \mid u,q)}.$$

Finally, to estimate $\hat{L}(i)$, we make use of the assumption that users sequentially inspect the ranking, and thus the following relationship holds: $L(i) = V(i) - V(i+1)$. Recall that $V(i) = \prod_{j=1}^{i-1} C(j)$ is the examination probability at rank i. For example, if 50% of users in the population examine rank 2 and only 20% of users from the same population view rank 3, a random user in that universe has a probability of 0.30 to stop at rank 2. Hence, $\hat{L}(i)$ can be estimated via:

$$\hat{L}(i;\hat{\mathbf{V}}) = \frac{\sum_{u\in U}\sum_{q\in Q(u)}\left[\hat{V}(i \mid u,q) - \hat{V}(i+1 \mid u,q)\right]}{\sum_{u\in U}\sum_{q\in Q(u)}\sum_{j=1}^{N}\left[\hat{V}(j \mid u,q) - \hat{V}(j+1 \mid u,q)\right]}.$$

These alternatives for computing $\hat{C}(i)$, $\hat{W}(i)$ and $\hat{L}(i)$ can be utilised if gaze information is not available, but click information is (the THUIR1, THUIR2, and THUIR3 datasets).

5.5.3 User Model Evaluation

The accuracy of a range of C/W/L user models is now assessed using the J&A, THUIR1, and THUIR3 datasets. The THUIR2 dataset is not used because the relevance judgements are shallow, with only top 5 documents annotated. The THUIR1 and THUIR3 datasets do not include view sequences, and hence the alternative formulations based on estimated view distributions $\hat{V}(i \mid u,q)$ for $\hat{C}(\cdot)$, $\hat{W}(\cdot)$, and $\hat{L}(\cdot)$ are employed.

Fitting The Parameter. The computation of view distributions (as described in Equation 5.2) employs a parameter ω fitted using view sequences and click sequences associated with the J&A dataset. This is a choice being made, not a compulsory action. Recall that J&A is the only dataset used in this study that contains view sequences. One way to fit the parameters is by minimising the distance between $\hat{C}(i)$ (empirical $C(\cdot)$ computed from view sequences) and $\hat{C}(i;\hat{\mathbf{V}})$ (empirical $C(\cdot)$ computed using view distributions). That is, the following problem needs to be solved:

$$\underset{\omega}{\operatorname{argmin}} \frac{1}{N} \cdot \sum_{i=1}^{N} w_i \cdot \left[\hat{C}(i) - \hat{C}(i; \hat{\mathbf{V}}, \omega)\right]^2 ,$$

where $N = 9$ in the case of the J&A dataset (that is, SERP length); w_i is the weight that is proportional to the frequency of user attentions at rank i; and $\hat{C}(i; \hat{\mathbf{V}}, \omega)$ is the observed $\hat{C}(i; \hat{\mathbf{V}})$ calculated using the parameter ω. This fitting process results in $\omega = 1.4$, which means that $\hat{V}(i \mid u, q) = \exp\left[(DC(u, q) - i)/1.4\right]$ when i is beyond the deepest click rank.

Evaluation Results. Table 5.11 (page 208) shows the results of measuring user model accuracy for several metrics. The three datasets exhibit the same general pattern, in the sense that the accuracy of a particular metric in one dataset is also reflected in the other two datasets. In general, RBP with $\phi = 0.80$, INSQ with $T = 3$, and the three adaptive metrics INST with $T = 3$, IFT-C1 with $T = 3$, and IFT with $T = 3$ are more accurate than several shallower metrics, such as RBP with $\phi \le 0.65$, Prec@K with $K \in \{1, 5\}$, SDCG@K with $K \in \{1, 5\}$, and RR, for all of $C(i)$, $W(i)$, and $L(i)$. In the case of RBP, increasing ϕ beyond 0.80 again decreases its accuracy.

Table 5.13 (page 211) shows the accuracy of several static and adaptive user models, computed using 1,060,216 queries drawn from `Yandex.ru`, a Russian Web search engine. It can be seen that shallower metrics, such as RBP with $\phi = 0.65$ and INSQ with $T = 1.5$ better predict observed $W(\cdot)$ and $C(\cdot)$ than their deeper versions, RBP with $\phi = 0.80$ and INSQ with $T = 3$. In particular, Prec@K with $K = 5$ and SDCG with $K = 7$ are more accurate than Prec@K with $K = 10$ and SDCG with $K = 10$ in terms of $W(\cdot)$. This result indicates that user behaviours on `Yandex.ru` are more top-weighted than those recorded in the lab-based datasets with a diverse task complexity (J&A, THUIR1, and THUIR3). For RBP, the most accurate $W(i)$ on `Yandex.ru` was achieved when $\phi = 0.70$. This result is a confirmation of what other authors have measured. Chapelle et al. [44] carried out experiments using commercial click logs, and found that the examination probabilities of RBP with $\phi = 0.70$ is close to the observed examination probabilities. Using an impression model based on click-gap distributions (see Section 3.5.2 on page 108), Zhang et al. [244] reported that the best fit parameter for RBP observed from `MSN` click logs is $\phi = 0.73$. Using maximum likelihood estimation on last probability, $L(i)$, Park and Zhang [163] demonstrated that the best fit parameter for RBP is $\phi = 0.78$. Table 5.12 summarises the previous comparisons between user behaviour as modelled by RBP and observed behaviour.

Metric	$\mathsf{WMSE}(\hat{C}, C)$ $(\times 10^{-3})$			$\mathsf{MSE}(\hat{W}, W)$ $(\times 10^{-3})$			$\mathsf{MSE}(\hat{L}, L)$ $(\times 10^{-3})$		
	J&A	TH1	TH3	J&A	TH1	TH3	J&A	TH1	TH3
Prec, $K = 1$	674.2	689.4	718.0	790.1	750.2	788.2	1089.3	827.7	985.3
Prec, $K = 5$	243.1	266.2	231.2	59.5	43.4	38.8	1113.7	959.0	883.2
Prec, $K = 9$	24.2	46.0	36.4	6.2	18.1	16.7	347.7	963.2	915.9
Prec, $K = 10$	24.2	24.1	27.4	7.3	21.2	22.3	347.7	726.2	808.1
SDCG, $K = 1$	674.2	689.4	718.0	790.1	750.2	788.2	1089.3	827.7	985.3
SDCG, $K = 5$	242.0	263.8	244.4	71.6	49.9	54.1	419.2	203.6	242.7
SDCG, $K = 9$	16.6	38.9	42.3	6.1	5.1	8.5	129.7	149.7	196.0
SDCG, $K = 10$	16.6	15.8	32.1	5.2	3.9	8.7	129.7	74.3	157.9
RBP, $\phi = 0.10$	524.5	536.1	561.2	612.9	574.3	608.6	912.2	649.8	799.4
RBP, $\phi = 0.50$	113.9	115.8	128.9	158.6	130.8	146.2	439.6	195.9	290.0
RBP, $\phi = 0.65$	37.5	37.7	47.3	61.3	42.2	49.9	300.3	92.9	153.3
RBP, $\phi = 0.80$	3.6	2.9	9.5	8.9	2.3	4.5	118.7	11.0	40.7
RBP, $\phi = 0.95$	12.0	11.6	15.6	48.2	52.0	53.0	57.2	215.0	261.6
INSQ, $T = 1$	57.3	56.9	77.8	75.6	57.5	69.1	434.8	215.4	314.7
INSQ, $T = 2$	18.3	17.1	31.8	20.2	11.8	16.8	229.9	69.1	135.2
INSQ, $T = 3$	7.4	6.1	17.5	17.5	13.7	16.6	133.2	22.9	73.2
RR	244.7	213.5	396.8	210.2	145.9	307.3	218.2	75.5	285.9
INST, $T = 1$	117.3	126.4	165.7	205.5	197.2	244.0	563.2	327.7	469.1
INST, $T = 2$	32.3	33.6	51.9	42.0	31.7	45.6	298.2	108.4	188.9
INST, $T = 3$	11.1	10.6	22.5	16.3	8.1	12.8	180.3	40.0	93.7
IFT-C1, $T = 1$	250.3	395.5	458.0	160.0	176.2	252.2	211.9	123.7	256.0
IFT-C1, $T = 2$	90.3	175.9	206.6	36.3	27.1	49.2	74.1	66.5	87.1
IFT-C1, $T = 3$	27.8	66.3	74.7	21.8	7.0	8.9	21.8	61.4	34.7
IFT-C2	5.5	14.0	22.0	22.6	44.4	51.9	60.3	374.5	535.0
IFT, $T = 1$	333.2	423.7	481.2	232.1	207.1	276.8	431.6	178.4	314.7
IFT, $T = 2$	131.2	186.1	218.0	63.7	36.8	57.8	230.2	67.6	100.9
IFT, $T = 3$	40.8	67.7	79.8	19.3	5.3	10.2	112.5	37.5	34.1

Table 5.11: User model accuracy using the J&A, THUIR1 (TH1), and THUIR3 (TH3) datasets. In the case of the THUIR1 and THUIR3 datasets, view sequences are inferred from click data. Lower values are more accurate. Blue color represents the best value from each group of values as a metric parameter is altered.

Author	Methodology	Best fit ϕ
Park and Zhang [163]	Using maximum likelihood estimation and click logs to find the best fit parameter for predicted $L(i)$.	$\phi = 0.78$
Chapelle et al. [44]	Visually comparing predicted view probabilities, $W(i)/W(1)$, with observed view probabilities computed from click logs.	$\phi = 0.70$
Yilmaz et al. [240]	Computing mean squared error between predicted view probabilities and observed view probabilities computed from click logs.	No parameter value was reported. However, RBP is more accurate than DCG, but is less accurate than EBU.
Zhang et al. [244]	Computing KL-Divergence between predicted view probabilities and observed view probabilities computed from click logs via an *impression model*.	$\phi = 0.73$
Carterette [39]	Visually comparing predicted $L(i)$ with observed $\hat{L}(i)$ computed from click logs.	No parameter value was reported. However, RBP is less accurate than DCG and RR.

Table 5.12: Comparisons between RBP user model and observed behaviour reported by other authors.

Note also that being accurate in one characteristic tends to correspond to accuracy in the other two characteristics. That is, the relationship between $C(i)$, $W(i)$, and $L(i)$ is reflected in their observed versions $\hat{C}(i)$, $\hat{W}(i)$, and $\hat{L}(i)$. The interrelationship among these three observed functions is stronger for unbounded effectiveness metrics, such as RBP and INSQ.

5.5.4 Empirical Evidence for Adaptive Models

Several metrics used in this study are adaptive, such as AP, RR, ERR, iRBU@K, INST, and IFT-C2, which means that the viewing behaviours of their simulated users are affected by the relevance of the documents. Reciprocal rank, ERR, and iRBU@K assume that the user's decision to shift their attention from rank i to rank $i+1$ is influenced only by the relevance of the document at rank i; the total gain accumulated so far, $\sum_{j=1}^{i} r_j$, influences the continuation function of INST; the rate of gain to date, $(\sum_{j=1}^{i} r_j)/i$, affects the viewing behaviours of users modelled by IFT-C2; finally, continuation function of AP depends on part of the gain vector beyond rank i that has not yet been examined by the user.

Moffat et al. [153] hypothesise that, all other things being equal, $C(i)$ decreases as the user encounters part of the ranking that is information-heavy. If this is correct, adaptivity is then a key to the development of a more accurate metric-based user model. Chapter 4 already provided an empirical support for this hypothesis using a logistic regression analysis, demonstrating that the odds of continuing decrease as the user gathers relevant documents to achieve their target (see Table 4.6 on page 150). This section revisits this hypothesis and seeks empirical support using a different experiment.

In the J&A, THUIR1, THUIR2, and THUIR3 datasets, a SERP is associated with relevance judgements, and with behaviours (click or view sequences) recorded from one or more users who viewed that SERP. This experiment builds two bins of SERPs, so that two rankings from different bins have a constrasting quality in their top-5 positions. First, SERPs are sorted by their total gains over their top-5 results, denoted by $S_5 = \sum_{i=1}^{5} g(r_i)$ with $0 \leq g(r_i) \leq 1$ being the gain mapping function. The 50 highest-scoring SERPs are placed in a *good* bin; and 50 SERPs with the lowest S_5 are placed in a *poor* bin. The size of the J&A dataset is small in terms of the number of (SERP ID, User ID) pairs. Instead, top-25 and bottom-25 are employed for J&A. Second, click or view sequences on these two bins are then examined to investigate whether there is a notable difference in the viewing behaviours. The expected outcome suggested by the hypothesis from Moffat et al. [153] is that expected search lengths (ESL) observed from the set of good SERPs are lower than those inferred from the set of poor ones.

In the case of the J&A dataset, empirical ESL is computed as the deepest viewed rank, since view sequences are available. For the other three datasets, empirical ESL is estimated using view distributions as $\mathsf{ESL} = \sum_{i=1}^{K} \hat{V}(i \mid u, q)$, where K is the SERP size. Recall that SERPs on THUIR1 are evaluated independently from each other; while those from the other datasets are grouped based on session IDs, and are evaluated with respect to the sessions. Chapter 4 has suggested that the viewing behaviour of the user when they examine the j th SERP is affected by the quality of the previously seen SERPs in the same session. To reduce the effect of this bias, a subset of SERPs for the first queries in the sessions are also considered, in which each SERP is deemed to be assessed independently of the others.

	WMSE($\hat{C}, C$) ($\times 10^{-3}$)	MSE($\hat{W}, W$) ($\times 10^{-3}$)	MSE($\hat{L}, L$) ($\times 10^{-3}$)
Prec, $K = 1$	475.2	551.8	517.9
Prec, $K = 3$	299.6	73.6	931.1
Prec, $K = 5$	197.9	49.7	1077.8
Prec, $K = 7$	136.4	61.9	1132.9
Prec, $K = 9$	101.7	77.6	1151.8
Prec, $K = 10$	93.3	85.3	1098.7
SDCG, $K = 1$	475.2	551.8	517.9
SDCG, $K = 3$	244.3	60.6	160.4
SDCG, $K = 5$	129.1	13.1	127.5
SDCG, $K = 7$	63.6	10.9	109.6
SDCG, $K = 9$	27.4	18.3	95.5
SDCG, $K = 10$	18.4	23.1	73.2
RBP, $\phi = 0.10$	348.4	392.9	362.7
RBP, $\phi = 0.50$	38.1	42.3	32.9
RBP, $\phi = 0.65$	2.9	2.3	1.6
RBP, $\phi = 0.80$	12.1	17.1	31.8
RBP, $\phi = 0.95$	65.6	111.1	460.0
INSQ, $T = 1$	18.8	11.9	53.0
INSQ, $T = 1.5$	4.4	5.8	11.7
INSQ, $T = 2$	2.0	14.5	3.4
INSQ, $T = 3$	7.8	36.0	20.0
RR	440.4	490.2	414.8
INST, $T = 1$	107.5	159.4	179.7
INST, $T = 2$	6.7	5.6	10.5
INST, $T = 3$	1.0	6.0	2.4
IFT-C1, $T = 1$	330.7	261.3	193.9
IFT-C1, $T = 2$	150.1	21.0	172.9
IFT-C1, $T = 3$	117.4	17.1	247.9
IFT-C2	82.7	118.0	853.9
IFT, $T = 1$	342.4	272.3	218.5
IFT, $T = 2$	151.3	22.7	156.9
IFT, $T = 3$	115.9	15.6	228.4

Table 5.13: User model accuracy using the `Yandex.ru` dataset. View sequences are inferred from click data. Lower values are more accurate. Blue color represents the best value from each group of values as a metric parameter is altered.

Dataset	Good		Poor		Difference	
	$\bar{S}_5$	$\overline{\mathsf{ESL}}$	$\bar{S}_5$	$\overline{\mathsf{ESL}}$	p value	Cohen's d
THUIR1 (Navigational)	2.55	4.49	0.64	6.53	**0.00**	**−0.74**
THUIR1 (Non-Nav.)	5.00	6.59	0.22	8.56	**0.00**	**−0.80**
THUIR1 (Remember)	5.00	6.09	0.32	7.99	**0.00**	**−0.76**
THUIR1 (Understand)	4.79	6.33	0.41	8.25	**0.00**	**−0.63**
J&A (All)	4.97	4.16	0.00	3.96	0.85	0.05
J&A (First Query)	4.45	6.00	0.96	6.84	0.39	−0.24
THUIR2 (All)	4.00	3.46	0.41	3.41	0.93	0.02
THUIR2 (First Query)	3.29	3.45	1.00	5.43	**0.00**	**−0.67**
THUIR3 (All)	5.00	4.53	0.52	3.61	0.12	0.31
THUIR3 (First Query)	4.57	5.18	1.07	4.56	0.29	0.21

Table 5.14: Expected search length (ESL) differences in good and poor bins of SERPs. Effect sizes (Cohen's d) and p values (independent two-sample t test) are also reported. A significance level (α) of 0.005 is employed with Bonferroni correction.

The results of this experiment are shown in Table 5.14. Assuming a null hypothesis that there is no difference between the two bins, and expected search lengths are just due to chance. Using an independent two-sample t test, it can be seen that $p < 0.005$ (with Bonferroni correction) for all subsets of THUIR1, and for a subset of THUIR2 which contains SERPs from the first queries. The effect sizes for cases where $p < 0.005$ are all above the *medium* level ($|d| > 0.50$). However, no evidence is found in the J&A and THUIR3 datasets. Figure 5.7 further shows the distributions of ESL for all four subsets of THUIR1. Two general patterns emerge. First, the median and mean ESL computed from good SERPs are lower than those computed from poor ones. Second, users who performed navigational search tasks have a lower expected search length than those who performed non-navigational counterparts.

The difference in ESL between good and poor SERPs in the THUIR1 dataset is also reflected in the behaviour of empirical $\hat{C}(i)$ and $\hat{W}(i)$. Figure 5.8 shows $\hat{C}(i)$ values computed from the two bins of SERPs for $1 \leq i \leq 5$. The values for $5 < i \leq 10$ are not of interest. As can be seen, users who examined good SERPs have a more reduced $\hat{C}(i)$ than those who inspected poor ones; and $\hat{C}(i)$ values observed from users who performed navigational tasks tend to be lower than those inferred from users who carried out informational or transactional tasks. In addition, Figure 5.9 presents empirical $\hat{W}(i)$ values for all rank positions in the SERP. It is clear that the $\hat{W}(i)$ function inferred from the collection of good SERPs is top-heavier than that inferred from the bin of poor ones.

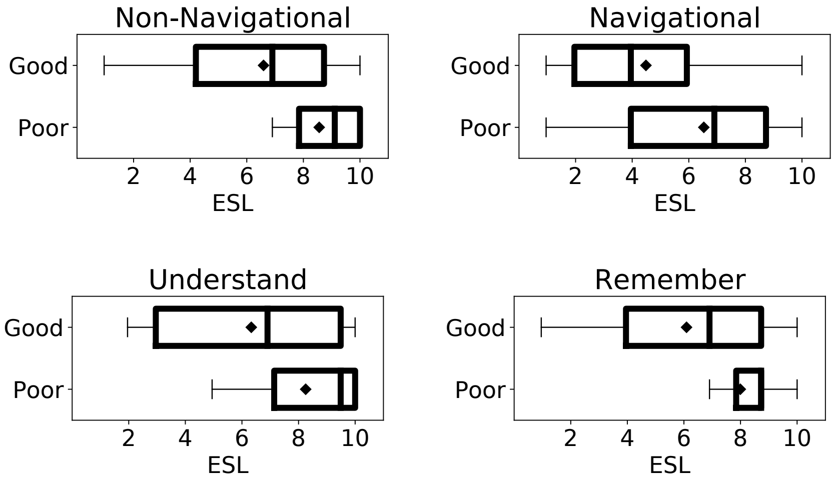


Figure 5.7: Distributions of expected search lengths for the good and poor SERPs observed from the THUIR1 dataset, stratified by query taxonomy (non-navigational and navigational) and by task cognitive level (understand and remember).

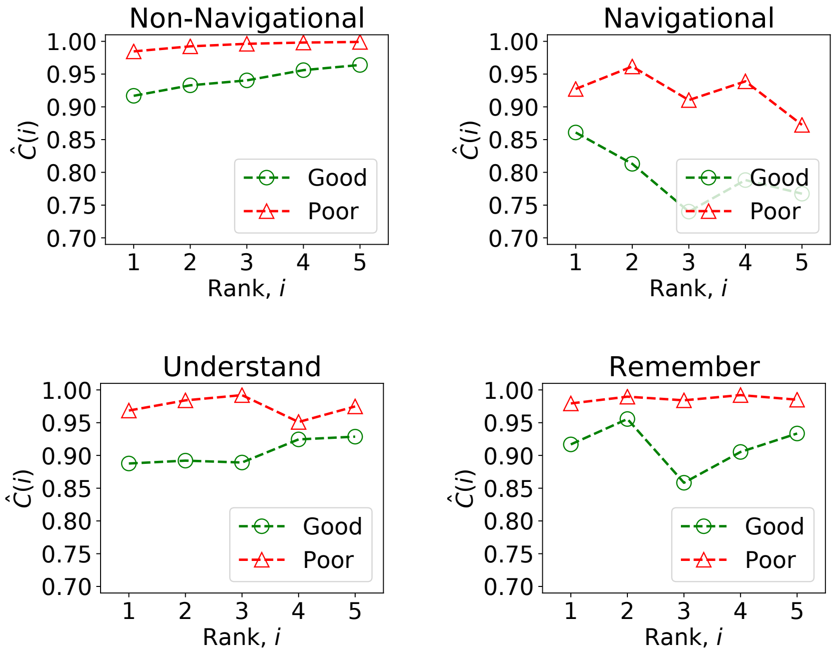


Figure 5.8: Empirical $\hat{C}(i)$ computed from the two groups of SERPs in the THUIR1 dataset for $1 \leq i \leq 5$, stratified by query taxonomy (non-navigational and navigational) and by task cognitive level (understand and remember).

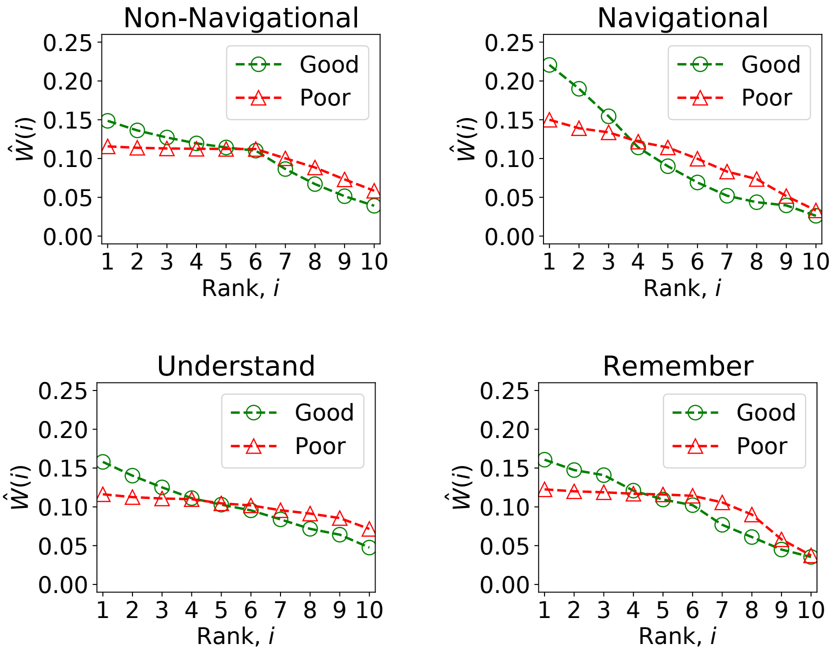


Figure 5.9: Empirical $\hat{W}(i)$ computed from the two groups of SERPs in the THUIR1 dataset, again stratified by query taxonomy (non-navigational and navigational) and by task cognitive level (understand and remember). Note that $\sum_{i=1}^{10} \hat{W}(i) = 1$.

Evidence from Web Search. The `Yandex.ru` dataset which contains 1,060,216 fully-judged SERPs is also used to see the difference in ESL between good and poor rankings. Two opposing extreme cases are considered for defining good and poor SERPs, since the `Yandex.ru` dataset contains many more queries compared to other datasets. SERPs with all items being relevant ($r_i = 1, 1 \leq i \leq 10$) are placed in a *good* bin; and SERPs with all items being non-relevant ($r_i = 0, 1 \leq i \leq 10$) are placed in a *poor* bin. As with THUIR2 and THUIR3, queries in the `Yandex.ru` dataset are also grouped based on session IDs. Hence, a subset of SERPs for the first queries in the sessions is also considered. As with the observations from THUIR1, Table 5.15 shows evidence from the `Yandex.ru` dataset that users tend to stop earlier when inspecting good SERPs than when inspecting poor SERPs. The function $\hat{W}(i)$ computed from good SERPs is also more top-weighted than that computed from poor SERPs.

To conclude, this section has found evidence that user behaviour is influenced by the relevance of the documents, and that users tends to stop earlier when inspecting an information-heavy ranking than when inspecting a poor ranking. This further suggests that adaptivity is a critical aspect for an accurate user model. However, the issue of how

Subset	Good		Poor		Difference	
	#SERPs	$\overline{\mathsf{ESL}}$	#SERPs	$\overline{\mathsf{ESL}}$	p value	Cohen's d
All	$104,040$	2.78	$4,083$	4.39	**0.00**	$\mathbf{-0.62}$
First Query	$51,018$	3.03	$1,780$	4.30	**0.00**	$\mathbf{-0.48}$

Table 5.15: Expected search length (ESL) differences in good and poor bins of SERPs in the `Yandex.ru` dataset. The number of SERPs in each bin, effect sizes (Cohen's d), and p values (independent two-sample t test) are also reported. A significance level (α) of 0.025 is employed with Bonferroni correction.

the adaptivity itself should be realised (such as, INST versus IFT-C1) has not yet been investigated, and will be addressed from the next paragraph.

Comparing Adaptive Factors. We now investigate which version of adaptivity best predicts continuation probability among AP, ERR, RR, iRBU, INST, IFT-C1, IFT-C2, IFT. First, a logistic regression is employed by optimising the following linear model:

$$\ln(\mathbf{c}_i/(1-\mathbf{c}_i)) = w_0 + w_1 \cdot C^*(i)\,, \tag{5.3}$$

where $\mathbf{w} = \{w_0, w_1\}$ is the set of coefficients that need to be estimated, $\mathbf{c}_i$ is a continuation variable computed from a view distribution (see Equation 4.11 on page 143), and $C^*(i)$ is a continuation probability function that only considers adaptive factors. For AP, IFT-C1, IFT-C2, and IFT, $C^*(i)$ functions are the same as their original $C(i)$ functions (see Equation 2.41 on page 54, Equation 2.52 on page 59, Equation 2.53 on page 59, and Equation 2.54 on page 60). Note that $C_{\mathsf{IFT}}(i) = C_{\mathsf{IFT\text{-}C1}}(i) \cdot C_{\mathsf{IFT\text{-}C2}}(i)$. In the case of INST, non-adaptive factors also exist. For example, all other factors being equal, $C_{\mathsf{INST}}(i)$ increases with rank position i. To exclude non-adaptive factors on $C_{\mathsf{INST}}(i)$, the following specification is considered: $C^*_{\mathsf{INST}}(i) = [T_i/(T_i+1)]^2$, where $T_i = \max(0, T - \sum_{j=1}^{i} r_j)$. For RR, ERR, and iRBU, this study uses $C^*(i) = 1 - r_i$. Second, the log-likelihood value computed from the best fit model is used to measure the goodness of fit of $C^*(i)$ to the observed behaviour given the best fit values for w_0 and w_1.

Note that the value of log-likelihood is negative, and that the closer the log-likelihood is to zero, the more accurate the model is. To decide whether any two adaptive models have a significant difference in the log-likelihood value, Vuong's z test is employed. This statistical test is a model selection tool based on likelihood ratio for making probabilistic statement about two *non-nested* models [228].

Model, $C^*(i)$	w_0		w_1		log-L	p
	value	p	value	p		(Vuong's z)
INST	0.462	0.000	6.575	0.000	-1.182×10^4	**0.001**
IFT-C1	0.401	0.000	1.435	0.000	-1.184×10^4	**0.000**
IFT	0.514	0.000	1.577	0.000	-1.211×10^4	**0.000**
AP	0.191	0.000	0.887	0.000	-1.249×10^4	**0.000**
RR, ERR, iRBU	0.689	0.000	0.200	0.000	-1.256×10^4	**0.000**
IFT-C2	0.729	0.000	0.095	0.000	-1.257×10^4	–

Table 5.16: Log-likelihood values (log-L) computed using six linear models (Equation 5.3 on page 215) based on the six $C^*(i)$ functions, and identified by the THUIR1 regressions. The p values on the right-most column relate to the difference between each row and its successor, computed using Vuong's z test for comparing two log-likelihood values from two non-nested models [228]. Note that the rows are sorted based on the log-likelihood values in decreasing order.

The THUIR1 and `Yandex.ru` datasets are employed for fitting the models and computing the log-likelihood values. In contrast to the THUIR2 and THUIR3 datasets, each query in both THUIR1 and `Yandex.ru` corresponds to more than one user, allowing for estimating two important quantities: the expected volume of relevance for undertaking the search, T; and the *expected minimum rate of gain* that keeps the user inspecting the SERP, A. Three metrics, INST, IFT-C1, and IFT have T as their parameter, while IFT-C2 and IFT are two metrics that depends on A.

To estimate T for a particular query, this experiment uses the approach described in Chapter 4 (see Equation 4.13 on page 147) by taking the average number of distinct relevant items clicked across all users that submitted that query. Suppose $U(q)$ is the set of user IDs that correspond to the query q. The expected minimum rate of gain A is estimated as follows:

$$\hat{A} = \sum_{i=i}^{N} \left[V(i \mid q) - V(i+1 \mid q)\right] \cdot \mathit{rate}(i)\,, \tag{5.4}$$

where N is the length of SERP; $V(i \mid q)$ is determined by averaging $V(i \mid u, q)$ across all users $u \in U(q)$, and is defined as $V(i \mid q) = 0$ for $i > N$; and $\mathit{rate}(i) = \left(\sum_{j=1}^{i} r_j\right)/i$, the rate of gain at rank i.

Tables 5.16 and 5.17 show the result of this experiment using, respectively, the THUIR1 and `Yandex.ru` datasets. The magnitudes of log-likelihood observed from `Yandex.ru` are much larger than those observed from THUIR1, since `Yandex.ru` is larger than THUIR1. In general, the p values for w_1 (the coefficient that corresponds to $C^*(i)$) are all near zero,

Model, $C^*(i)$	w_0		w_1		log-L	p
	value	p	value	p		(Vuong's z)
IFT-C2	-0.028	0.000	0.372	0.000	-6.476×10^6	**0.000**
IFT	0.153	0.000	0.447	0.000	-6.482×10^6	**0.000**
RR, ERR, iRBU	0.282	0.000	-0.196	0.000	-6.495×10^6	**0.000**
INST	0.159	0.000	1.419	0.000	-6.496×10^6	**0.000**
IFT-C1	0.179	0.000	0.121	0.000	-6.508×10^6	**0.000**
AP	0.130	0.000	0.128	0.000	-6.510×10^6	–

Table 5.17: Log-likelihood values (log-L) computed using six linear models (Equation 5.3 on page 215) based on the six $C^*(i)$ functions, and identified by the Yandex.ru regressions. The p values on the right-most column relate to the difference between each row and its successor, computed using Vuong's z test for comparing two log-likelihood values from two non-nested models [228]. Note that the rows are sorted based on the log-likelihood values in decreasing order.

indicating that any adaptive factor is significant for predicting the user behaviour. As can be seen, INST has the best fit adaptive factor among the six models in the THUIR1 dataset, and is better than AP and IFT-C1 in both datasets (Vuong's z test, $p < 0.001$ in all cases). Adaptive factor in IFT-C2, which is affected by the minimum rate of gain A, exhibits two opposing results. Among the six adaptive models, IFT-C2 has the worst fit in the THUIR1 dataset with a diverse task complexity, but has the closest fit in the Yandex.ru dataset whose observed behaviour tends to be top-heavy compared to the behaviour observed from the THUIR1 dataset (see Table 5.11 on page 208 and Table 5.13 on page 211).

5.6 Model Accuracy and Satisfaction

After exploring correlation coefficients between scores generated from metrics with various user models and satisfaction ratings at both query- and session-level in Section 5.4 (link 5 in Figure 5.1 on page 177), and after measuring the extent to which those user models predict observed behaviour in Section 5.5 (link 6 in Figure 5.1), we ask whether this dualism has a connection at least to some extent (link 8 in Figure 5.1), and further hypothesise that the metrics with user models that fit observed user behaviour also tend to be the metrics that correlate well with user satisfaction ratings. The goal of this section is to develop evidence for or against this hypothesis.

5.6.1 Tuning Parameters via Model Accuracy and Satisfaction

This section examines whether improvements in model accuracy tend to be followed by increases in correlation between metric scores and satisfaction ratings. Figure 5.10 shows joint plots between correlation with satisfaction ratings (query-level ratings for both THUIR1 and THUIR3, and session-level ratings for the J&A datasets) and user model accuracy on the two vertical scales, as the parameters for those metrics are varied on the horizontal scale, for RBP and INST. In the case of the J&A dataset, query scores are aggregated using the weighted mean approach described in Chapter 4 (see Equation 4.18 on page 165).

As a general pattern, the accuracy of the user model (via either $C(i)$, $W(i)$, or $L(i)$) closely reflects the correlation between metric score and satisfaction ratings. That is, increased accuracy of any of $C(i)$, $W(i)$, or $L(i)$ tends to correspond to an increased correlation between metric score and user satisfaction. Other metrics, such as Prec@K, SDCG@K, and INSQ, also exhibit the same general pattern. In regard to model accuracy, it is clear that $\hat{C}(i)$, $\hat{W}(i)$, or $\hat{L}(i)$ each tend to define the others. This observation signals a clear relationship between a metric fit in terms of its ability to predict user behaviour (via the corresponding user model), and metric fit in terms of its ability to act as a surrogate for explicit user satisfaction ratings (via the assessed score of the SERP).

Note that collection of user-reported satisfaction ratings can be expensive, since it requires laboratory-based user studies (such as those logged in the J&A, THUIR1, THUIR2, and THUIR3 datasets), and involves a non-trivial number of participants. The results in Figure 5.10 suggest an alternative for estimating parameters, such as K for SDCG@K, ϕ for RBP, and T for INST. The tuning process for those model-based metrics can be done via a set of click sequences, as opposed to the set of satisfaction ratings, which can be collected in real time from an operational search engine.

5.6.2 Metrics Based on What Users Have Seen

In the previous section, it has been shown that tuning the parameter of a particular metric with the goal of increasing the correlation between scores and satisfaction ratings can, to some extent, be done using logged behaviours, such as click data. This section addresses a more general issue, asking whether building a more accurate user model will be rewarded by a metric whose scores are more correlated with satisfaction ratings.

We carry out an experiment to see whether metrics that utilise what information users have looked at, such as clicks, mouse-hovers, or the view distributions $\hat{V}(i \mid u, q)$ have a

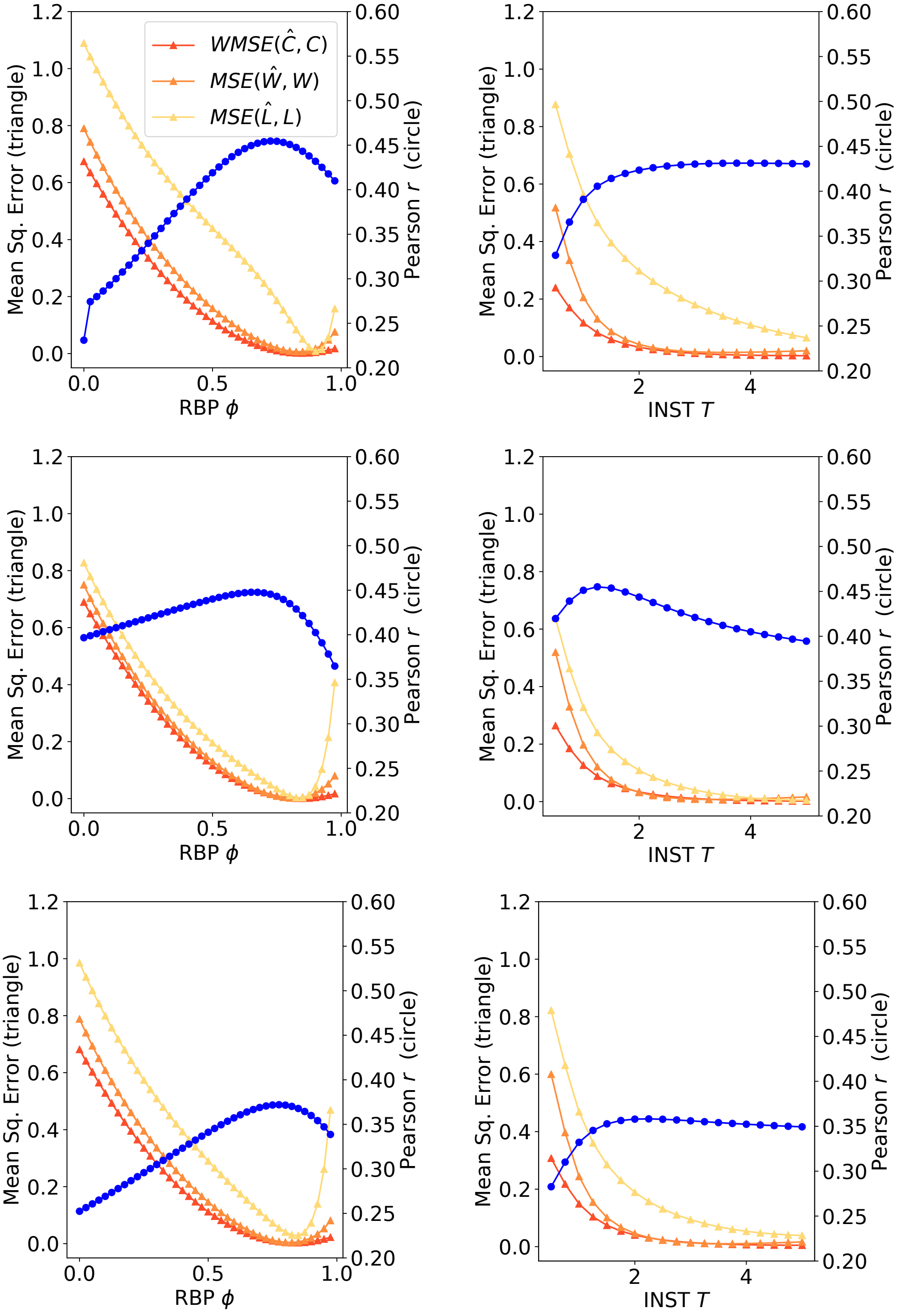


Figure 5.10: Joint plots between correlation with session satisfaction ratings and user model accuracy for several parameter values. The plots are for RBP (parameter ϕ) and INST (parameter T), respectively, using the J&A (first row), THUIR1 (second row), and THUIR3 (third row) datasets. Blue dotted line represents Pearson's correlation coefficients that are associated with the right-hand y-axis.

better correlation with satisfaction than conventional metrics that do not directly have this knowledge. Recall that the distributions $\hat{V}(i \mid u, q)$ serve as gold standards for computing user model accuracy in the THUIR1, THUIR2, and THUIR3 datasets.

Metrics that use view distributions are computed as $\sum_{i=1}^{K} r_i \cdot \left(\hat{V}(i \mid u, q)/Z\right)$, where $Z = \sum_{j=1}^{K} \hat{V}(j \mid u, q)$ is the normalisation factor. Note that $\hat{V}(i \mid u, q)$ has the parameter ω that represents user persistence beyond the deepest click rank, and this experiment uses $\omega \in \{0.5, 1.0, 1.5, 2.0\}$. The other metrics used in this experiment are based on rank sequences generated from click and mouse-hover actions. These metrics are the *mean*, *max*, and *min* values from the corresponding gain vectors; and are denoted by cMEAN, cMAX, and cMIN for click-based ones, and by hMEAN, hMAX, and hMIN for mouse-based ones. In regard to the use of mouse-hover information, Guo and Agichtein [70] have demonstrated that mouse positions can be used to infer gaze positions. They propose a machine learning approach to predict when eye and mouse movements are closely coordinated in the screen.

This experiment uses all SERPs in the THUIR1 dataset (291 SERPs). For THUIR2 and THUIR3, cases in which the deepest click rank positions are beyond the relevance judgement pool depths are excluded. These result in 666 SERPs for THUIR2, and 1,259 SERPs for THUIR3. Table 5.18 shows the resultant correlation coefficients. The coefficients computed from five ad-hoc metrics that perform relatively well on these three datasets (SDCG@K with $K = 9$, RBP with $\phi = 0.80$, iRBU@K with $\phi = 0.99$, D-BPM, and INST (ETG) with $T = 2$) are also reported for comparisons (see Tables 5.2, 5.3, 5.4 on pages 187, 188, 189, respectively). It can be seen that *ideal* metrics defined based on clicks and view distributions perform better than the five offline metrics. Metrics based on view distributions provide the highest correlation coefficients on the THUIR1 dataset, but not on the other datasets. Similarly, click-based metrics are superior compared to the others on THUIR2 and THUIR3. This provides further evidence that user satisfaction should be modelled as a function of based on seen items; and increasing the accuracy of a user model so that it reflects what the user examines would also lead to an increased correlation between its scores and satisfaction ratings.

5.7 Summary

Many effectiveness metrics have dual aspects: numeric scores for SERPs, used in evaluation as a surrogate for user satisfaction; and models that (should) reflect how users interact with SERPs. This chapter has used five pre-existing resources, and the C/W/L approach to metric definition, to shed fresh light on that duality.

	THUIR1	THUIR2	THUIR3
View Distribution, $\omega = 0.5$	0.515	0.372	0.330
View Distribution, $\omega = 1.0$	0.508	0.367	0.354
View Distribution, $\omega = 1.5$	0.494	0.360	0.366
View Distribution, $\omega = 2.0$	0.480	0.356	0.372
cMEAN	0.362	0.569	0.434
cMAX	0.253	0.576	0.423
cMIN	0.361	0.513	0.321
hMEAN	0.484	–	0.333
hMAX	0.304	–	0.299
hMIN	0.450	–	0.239
SDCG, $K = 9$	0.448	0.351	0.376
RBP, $\phi = 0.80$	0.435	0.348	0.371
iRBU@K $\phi = 0.99$	0.502	0.370	0.340
D-BPM	0.307	0.390	0.294
INST (ETG), $T = 2$	0.484	0.354	0.386

Table 5.18: Correlation coefficients between query scores and query-level satisfaction ratings, with the query scores computed from view distributions $\hat{V}(i \mid u, q)$, from click- and hover-based metrics, and from five conventional metrics. Blue values represent three highest Pearson's correlation coefficients in each column. A horizontal line in the middle separates the *ideal* metrics that are based on what users have seen (based on $\hat{V}(i \mid u, q)$, click, and hover) from those that are not.

Section 5.4 has calculated correlation coefficients between metric scores and satisfaction ratings, and found that the relationship between score and satisfaction is confounded by the query taxonomy, that is, by the user's initial rationale for performing their search activity. When queries are mostly navigational, shallow metrics, such as RR, Prec@K with $K = 1$, SDCG@K with $K = 1$, RBP with $\phi = 0.1$, and iRBU@K with $\phi = 0.10$ correlate better with user satisfaction than their deeper versions. On the other hand, when datasets with a diverse task complexity are utilised, adaptive metrics, such as iRBU@K with $\phi = 0.99$ and INST (ETG version with $T = 2$) appear to be better correlated with query satisfaction than Prec@10, RR, and AP1@K. Other metrics, RBP, SDCG, DCG, and INSQ, also correlate relatively well with satisfaction ratings at both query- and session-levels.

Several key findings have emerged from Section 5.4. First, scores generated by the ETG versions of adaptive metrics tend to be better correlated than those generated by their ERG versions, as the user's goal T increases. Second, mapping the relevance vector using an exponential gain mapping function leads to scores that are better correlated with satisfaction ratings, compared to scores calculated using the linear version. Third, click-based actions, such as *precision at lowest click* and *maximum reciprocal clicked rank*, have a better relationship with query-level satisfaction ratings than the query reformulation binary indicator.

After exploring the relationship between metric scores and satisfaction in Section 5.4, Section 5.5 then investigated the dual of that relationship – the correlation between predicted behaviour (user model) and observed behaviour. This section proposed a methodology for inferring observed behaviour from the perspective of the C/W/L structure, and then used that method to measure to the extent to which user models predicted three empirical behaviours, $\hat{C}(\cdot)$, $\hat{W}(\cdot)$, and $\hat{L}(\cdot)$. The results show that metrics that have been shown to be better correlated with satisfaction (as described in Section 5.4), such as RBP with $\phi = 0.80$, INSQ, and INST, also appear to be the metrics that better predict $\hat{C}(\cdot)$, $\hat{W}(\cdot)$, and $\hat{L}(\cdot)$. Further, the last part of Section 5.5 also found evidence that the user's expected search depth is affected by the query taxonomy and the relevance of the documents inspected. These findings create a clear connection between the two sides of the dual relationship shown in Figure 5.1 (page 177).

Section 5.6 then investigated whether the metrics with user models that fit observed user behaviour also tend to be the metrics that correlate well with user satisfaction ratings. The results show that increased accuracy of any of $C(i)$, $W(i)$, or $L(i)$ tends to be followed by an increased correlation between metric score and user satisfaction. Moreover, this section also demonstrated that *ideal* metrics defined using what users have looked at (that are deemed to be the most accurate ones), such as clicks and gaze distributions, are indeed better correlated with satisfaction ratings. Hence, the effort of making a metric more accurate would be, to some extent, rewarded by an increased correlation with satisfaction.

To conclude, we have constructed and demonstrated a new framework for meta-evaluation of metrics, based on comparing predicted user behaviour with measured user actions, and, using that framework, have shown that metrics that correlate well with user satisfaction have as their duals user models that correlate well with observed user actions. This is an important new way of thinking about meta-evaluation of metrics. Note also that the relationship works in both directions – metrics that have accurate user models (in terms of $C(i)$, $W(i)$, and $L(i)$ being good fits to observed behaviour) can then be argued as being the ones that should be used as the most appropriate surrogates for user satisfaction.

Chapter 6

Conclusion and Future Work

Users typically submit multiple queries during the course of each search session. Hence, it is useful to extend the traditional query-based IR evaluation in order to assess a multi-query session as a single unit. By arguing that metric scores should reflect what users have experienced during the course of the session, we used search interaction logs to model user behaviour and satisfaction, and to derive evidence that allows metric comparisons. This chapter summarises those findings, and in Section 6.2, considers directions for possible future work.

6.1 Conclusion

Chapters 3, 4, and 5 have presented and explored the following ideas.

Empirical $C(i)$ and Impression Models. A critical step in the development of user-based metrics is to understand user search behaviour. A way of operationalising observations of behaviour is via the notion of conditional continuation probability, $C(i)$. Chapter 3 addressed the question: *is it possible to use search interaction logs to model user behaviours?* We propose three heuristic rules for computing observed continuation probability using user logged viewing behaviours, such as a collection of impression or gaze sequences. Our experiment showed that these three rules all result in the same behavioural patterns in regard to $C(i)$, namely that it increases with rank i, confirming the "sunk cost" property hypothesised by Moffat et al. [153]. We also demonstrated the use of observed $C(i)$ to fit the parameters of three metrics, SDCG, RBP, and INSQ, and found that INSQ has a more accurate user model, compared to the other two metrics.

When impression sequences are not available but click sequences are, observed $C(i)$ can still be estimated via an impression model, a tool that is useful for inferring impression distributions from click logs. We proposed a new impression model, arguing that the user tends to examine all items prior to the deepest clicked rank (dc), and tends to view a number of items beyond dc. Next, our analysis of user observation data found that the number of items inspected beyond dc is negatively correlated with the number of clicks performed by the user, and increases with dc itself. Finally, our experiments demonstrated that an impression model that considers these behaviours is more accurate than previous approaches, such as the model based on *click gaps* by Zhang et al. [244].

These two tools, impression models and methods for inferring $C(i)$, were then used in Chapter 4 to investigate factors affecting user behaviours in the development of session-based metrics. As described in Chapter 5, impression models are also useful for measuring the accuracy of metric-based user models based on evidence derived from click logs.

Session-Based Metric and Query-to-Session Aggregation Framework. Chapter 4 then presented a second major contribution, the development of methods for scoring sessions. Two goals for session evaluation were considered: (1) the first goal was to develop a session-based effectiveness metrics for session test collections, where a particular topic is associated with a fixed sequence of queries; and (2) the second one was to build a fitted relationship between session satisfaction ratings and individual query scores.

In regard to the first goal, we proposed the session-based C/W/L framework by extending its query-based version. In the new approach the user model is characterised by two behaviours: conditional continuation probability at rank i when inspecting the jth SERP, $C(j, i)$ (query-level behaviour); and conditional reformulation probability, $F(j)$ (session-level behaviour). We then employed search interaction logs from two commercial search engines to investigate factors that affect both query- and session-level behaviours. This analysis required the impression models and methods for computing empirical $C(i)$ described in Chapter 3. Two main findings emerged. First, we confirmed that the INST user model is appropriate for modelling query-level behaviour. Second, the query position j in the session, the user's goal T at the beginning of search, and the unmet number of relevant items to date, are all positively correlated with $F(j)$.

By incorporating those factors affecting both $C(j, i)$ and $F(j)$, we developed sINST, a new metric for session evaluation, the first session-based metric that is adaptive and goal-sensitive. Although three existing session metrics, LCYsRBP, sDCG, and KsDCG, can be made goal-sensitive by setting their parameters to fit a certain value of T (anticipated number of useful documents), they are not adaptive. Further, our experiment showed that sINST gives a closer fit to observed user behaviour than do those previous metrics. Finally, we described a method for approximating sINST that is less expensive than the Monte

Carlo, which requires a large number of randomised trials.

To address the second goal for session evaluation, we utilised several pre-existing datasets from lab-based user studies to explore factors influencing session-level satisfaction ratings. Previous work has suggested that session satisfaction is only affected by individual query positions with the *last* query being the most useful factor. Our study found that a quality-based factor (the spectrum from the best to the worst queries) is also significant, and that combining both positional- and quality-based factors provides a better correlation with session satisfaction than do positional-based factors alone.

Based on these findings, we proposed two novel query-to-session aggregation frameworks. The first one merges query weights derived from positional- and quality-based factors using a linear combination scheme. The second one is based on the notion of *forgetfulness*, where the aggregate score can be interpreted as the rate of remembered query utility per SERP inspected. When query-level satisfaction ratings are used to represent query scores, our proposed aggregation functions provide a better fit with session-level satisfaction than do previous aggregation approaches. The proposed query-to-session aggregation functions were then employed in Chapter 5 for meta-evaluation of metrics at the level of sessions.

Meta-Evaluation Framework. As with IR systems, evaluation metrics also need to be evaluated. In Chapter 5 we proposed a meta-evaluation framework, arguing that metrics have dual aspects: metric scores that are intended to have a relationship with user satisfaction; and user models that are correlated with observed user behaviour.

We calculated correlation coefficients between metric scores and satisfaction ratings at both query- and session-levels. For scoring sessions, we used methods for combining individual query scores described in Chapter 4, since user observation data was available in connection with some of the public datasets that were employed. Several findings emerged. First, the resultant correlation coefficients are confounded by the query taxonomy. When queries are mostly navigational, shallow metrics are better correlated with satisfaction than are deeper metrics. The opposite results are observed, when datasets with a diverse task complexity are employed. Second, the ETG versions of adaptive metrics tend to be better correlated with satisfaction than those computed by their *expected rate of gain* (ERG) versions, as the user's desire T increases. Third, the exponential gain values provide scores that have a better relationship with satisfaction, compared to scores computed from the linear gain values.

After exploring the correlation between metric scores and satisfaction, we also explored the dual of that relationship – the connection between user models and observed behaviours. We proposed a method for measuring the extent to which a user model fits observed behaviour via three functions, $C(\cdot)$, $W(\cdot)$, and $L(\cdot)$, building on the impression model developed in Chapter 3. We found that several metrics that have been shown to be better correlated with satisfaction also tend to be the metrics that better predict $C(\cdot)$, $W(\cdot)$, and $L(\cdot)$. We also suggested that adaptivity, where $C(\cdot)$ changes as the user encounters relevance in the ranking, is a key to the development of an accurate user model, and showed that INST has a better fit adaptive factor than do AP and IFT-C1.

Finally, we found that the metrics with accurate user models tend to be the metrics that correlate relatively well with user satisfaction. We demonstrated that tuning the parameters of several *unbounded* metrics, such as RBP and INST, can be done via recorded user behaviours (such as click logs), providing (at least to a limited extent) the same benefit as the tuning process via datasets with user satisfaction ratings. Note that, in contrast to click logs, user-reported satisfaction ratings are difficult to collect at scale.

6.2 Future Work

This section presents possible future work based on the findings described in Section 6.1.

Other Factors Affecting $F(j)$. The experiments reported in Section 4.5 suggests that $F(j)$ is influenced by three factors: the query position in the session, the expected number of useful documents, and the total number of useful documents accumulated to date. One clear future direction is to seek other possible factors that influence $F(j)$. It has been conjectured that user behaviour is affected by rate of gain [20]. In the session-level behaviour, the rate at which gain has been accrued in query j (rather than through the whole session so far) provides more precise estimation of $F(j)$. Other factors are also possible. For example, De Vries et al. [61] introduce the notion of *tolerance to irrelevance*. That is, the user exits from the current SERP and submits a new query when the total number of non-relevant items inspected by the user has reached their tolerance to irrelevance.

Both rate of gain and tolerance to irrelevance might be useful for improving the estimation of $F(j)$. However, their interaction with current suggested behaviours (such as the sunk cost property) also needs to be investigated. For example, consider a case when the ranking contains non-relevant items throughout. The sunk cost property states that the continuation probability always increases with i. However, if the user has a tolerated number of non-relevant items in their mental state, their continuation probability should decrease after some time. Hence, the tension raised between sunk cost and tolerance to irrelevance should be carefully handled to better predict user search behaviour.

New Session Test Collection. Despite the need for session-based evaluation, session test collections are limited. Test collections from the TREC Session Track 2010 – 2014 are the primary existing resources for offline session assessment. However, none of them considers user variations, such as query and expectation variations. Moffat et al. [155] show that the source of variability caused by users' initial queries is more significant than other sources of variability, such as topics and systems. Therefore, another future direction is to develop a new session test collection that also considers user variations. Fortunately, this can be done by extending UQV100 – an existing query-based test collection that considers query and expectation variations, and that contains *backstories* (that is, "information need statements") [24]. Query variations have also been collected in connection with the recent CC-News-En collection [143].

Note that a session test collection requires a topic to be associated with a sequence of queries. With UQV100, the sequences of queries per backstory could be constructed or simulated in many ways. One way is to group queries based on the likelihood that a query would be in a certain position in the session: the first group contains queries that most likely would serve as initial queries; the second one contains those that most likely appear as the second queries; and so on. This grouping process can be done using an automatic clustering mechanism, considering several reformulation rules, such as *generalisation* and *specialisation* [114]. Thomas et al. [213] also propose a useful rule that query Q_2 is considered as the reformulation for query Q_1 when Q_2 has at least $1/3$ of its terms in common. Another option would be to hire crowd-workers to manually group them, or to collect query sequences from crowd-workers.

A Few Final Words. We have constructed a methodology for computing empirical continuation probability (query-level behaviour) and reformulation probability (session-level behaviour) using logged behaviours (impressions, eye-fixations, or clickthroughs). This suggests that it is possible to compare effectiveness metrics based on evidence derived from logged behaviours, and provides a foundation for future directions in regard to the investigation of other possible factors that affect user behaviour. Further, we found evidence that query- and session-level behaviours are goal directed, and are affected by the progress towards goal. These findings suggest that search effectiveness metrics should be goal sensitive, adaptive, and session-oriented. The development of new metrics should also be supported by the development of new session test collections that accommodate user variations.

Conversational and interactive search systems are the future in the field of IR [9]. In order to improve interactive IR systems, appropriate evaluation methods should be developed. In contrast to the classical IR effectiveness model that is based on a single-query response, actual interactions between users and systems involve search sessions, each of which consists of multiple queries. In this thesis, we have addressed this challenge, and using a wide range of data and modelling techniques have demonstrated that user behaviour and user satisfaction are critical ingredients of good session evaluation strategies.

Bibliography

[1] M. Ageev, Q. Guo, D. Lagun, and E. Agichtein. Find it if you can: A game for modeling different types of web search success using interaction data. In *Proc. SIGIR*, pages 345–354, 2011.

[2] E. Agichtein, E. Brill, and S. T. Dumais. Improving web search ranking by incorporating user behavior information. In *Proc. SIGIR*, pages 19–26, 2006.

[3] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proc. WSDM*, pages 5–14, 2009.

[4] A. Al-Maskari and M. Sanderson. A review of factors influencing user satisfaction in information retrieval. *J. Amer. Soc. Inf. Sc. Tech.*, 61(5):859–868, 2010.

[5] A. Al-Maskari, M. Sanderson, and P. Clough. The relationship between IR effectiveness measures and user satisfaction. In *Proc. SIGIR*, pages 773–774, 2007.

[6] A. Al-Maskari, M. Sanderson, P. Clough, and E. Airio. The good and the bad system: Does the test collection predict users' effectiveness? In *Proc. SIGIR*, pages 59–66, 2008.

[7] M. J. Albers and L. Kim. User web browsing characteristics using palm handhelds for information retrieval. In *Proc. IEEE Professional Comm. Soc. Internat. Professional Comm. Conf*, pages 125–135, 2000.

[8] J. Allan, B. Carterette, and J. Lewis. When will information retrieval be "good enough"? In *Proc. SIGIR*, pages 433–440, 2005.

[9] J. Allan, J. Arguello, L. Azzopardi, P. Bailey, T. Baldwin, K. Balog, H. Bast, N. Belkin, K. Berberich, B. von Billerbeck, J. Callan, R. Capra, M. Carman, B. Carterette, C. L. A. Clarke, K. Collins-Thompson, N. Craswell, W. B. Croft,

J. S. Culpepper, J. Dalton, G. Demartini, F. Diaz, L. Dietz, S. T. Dumais, C. Eickhoff, N. Ferro, N. Fuhr, S. Geva, C. Hauff, D. Hawking, H. Joho, G. Jones, J. Kamps, N. Kando, D. Kelly, J. Kim, J. Kiseleva, Y. Liu, X. Lu, S. Mizzaro, A. Moffat, J. Nie, A. Olteanu, I. Ounis, F. Radlinski, M. de Rijke, M. Sanderson, F. Scholer, L. Sitbon, M. Smucker, I. Soboroff, D. Spina, T. Suel, J. Thom, P. Thomas, A. Trotman, E. Voorhees, A. P. de Vries, E. Yilmaz, and G. Zuccon. Research frontiers in information retrieval: Report from the third strategic workshop on information retrieval in Lorne (SWIRL 2018). *SIGIR Forum*, 52(1):34–90, Aug. 2018.

[10] E. Amigó, J. Gonzalo, and F. Verdejo. A general evaluation measure for document organization tasks. In *Proc. SIGIR*, pages 643–652, 2013.

[11] E. Amigó, H. Fang, S. Mizzaro, and C. Zhai. Axiomatic thinking for information retrieval: And related tasks. In *Proc. SIGIR*, pages 1419–1420, 2017.

[12] E. Amigó, H. Fang, S. Mizzaro, and C. Zhai. Are we on the right track? An examination of information retrieval methodologies. In *Proc. SIGIR*, pages 997–1000, 2018.

[13] E. Amigó, D. Spina, and J. C. de Albornoz. An axiomatic analysis of diversity evaluation metrics: Introducing the rank-biased utility metric. In *Proc. SIGIR*, pages 625–634, 2018.

[14] L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshank, R. Mayer, P. R. Pintrich, J. Raths, and M. C. Wittrock. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives.* New York: Longman, Jan. 2001.

[15] J. Arguello, F. Diaz, and J. Callan. Learning to aggregate vertical results into web search results. In *Proc. CIKM*, pages 201–210, 2011.

[16] J. A. Aslam, E. Yilmaz, and V. Pavlu. A geometric interpretation of R-precision and its correlation with average precision. In *Proc. SIGIR*, pages 573–574, 2005.

[17] J. A. Aslam, V. Pavlu, and E. Yilmaz. A statistical method for system evaluation using incomplete judgments. In *Proc. SIGIR*, pages 541–548, 2006.

[18] A. Aula, P. Majaranta, and K. Räihä. Eye-tracking reveals the personal styles for search result evaluation. In *Human-Computer Interaction - INTERACT*, 2005.

[19] L. Azzopardi. The economics in interactive information retrieval. In *Proc. SIGIR*, pages 15–24, 2011.

[20] L. Azzopardi, P. Thomas, and N. Craswell. Measuring the utility of search engine result pages. In *Proc. SIGIR*, pages 605–614, 2018.

[21] L. Azzopardi, P. Thomas, and A. Moffat. CWL_eval: An evaluation tool for information retrieval. In *Proc. SIGIR*, pages 1321–1324, 2019.

[22] L. Azzopardi, R. W. White, P. Thomas, and N. Craswell. Data-driven evaluation metrics for heterogeneous search engine result pages. In *Proc. CHIIR*, pages 213–222, 2020.

[23] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley Longman Publishing Co., Inc., 1999.

[24] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. UQV100: A test collection with query variability. In *Proc. SIGIR*, pages 725–728, 2016.

[25] K. Balog, L. Kelly, and A. Schuth. Head first: Living labs for ad-hoc search evaluation. In *Proc. CIKM*, pages 1815–1818, 2014.

[26] F. Baskaya, H. Keskustalo, and K. Järvelin. Modeling behavioral factors ininteractive information retrieval. In *Proc. CIKM*, pages 2297–2302, 2013.

[27] S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, and D. Grossman. Temporal analysis of a very large topically categorized web query log. *J. Amer. Soc. Inf. Sc. Tech.*, 58(2):166–178, 2007.

[28] Y. Bernstein and J. Zobel. Redundant documents and search effectiveness. In *Proc. CIKM*, pages 736–743, 2005.

[29] D. Bilal. Children's use of the yahooligans! web search engine: I. Cognitive, physical, and affective behaviors on fact-based search tasks. *J. Amer. Soc. Inf. Sci.*, 51(7): 646–665, 2000.

[30] P. Borlund and N. Pharo. A need for information on information needs. In *Proc. CoLIS*. University of Borås, Information Research, 2019.

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

[32] C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *Proc. SIGIR*, pages 33–40, 2000.

[33] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *Proc. SIGIR*, pages 25–32, 2004.

[34] C. Buckley and E. M. Voorhees. Retrieval system evaluation. *In TREC: Experiment and Evaluation in Information Retrieval*, pages 53–75, 2005.

[35] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proc. ICML*, pages 89–96, 2005.

[36] S. Büttcher, C. L. A. Clarke, P. C. K. Yeung, and I. Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. In *Proc. SIGIR*, 2007.

[37] F. Cacheda and A. Viña. Understanding how people use search engines: A statistical analysis for e-Business. In *Proc. Conf. and Exhb. e-Business and e-Work*, pages 319–325, 2001.

[38] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li. Context-aware query suggestion by mining click-through and session data. In *Proc. KDD*, pages 875–883, 2008.

[39] B. Carterette. System effectiveness, user models, and user utility: A conceptual framework for investigation. In *Proc. SIGIR*, pages 903–912, 2011.

[40] B. Carterette and P. Chandar. Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proc. CIKM*, pages 1287–1296, 2009.

[41] B. Carterette and R. Jones. Evaluating search engines by modeling the relationship between relevance and clicks. In *Proc. Neural Inf. Processing Systems*, pages 217–224, 2007.

[42] M. Chae and J. Kim. Do size and structure matter to mobile users? An empirical study of the effects of screen size, information structure, and task complexity on user activities with standard web phones. *Behav. Inform. Tech.*, 23(3):165–181, 2004.

[43] P. Chandar, J. Garcia-Gathright, C. Hosey, B. S. Thomas, and J. Thom. Developing evaluation metrics for instant search using mixed methods methods. In *Proc. SIGIR*, pages 925–928, 2019.

[44] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proc. CIKM*, pages 621–630, 2009.

[45] O. Chapelle, T. Joachims, F. Radlinski, and Y. Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM Trans. Inf. Sys.*, 30(1):6:1–6:41, Mar. 2012.

[46] Y. Chen, K. Zhou, Y. Liu, M. Zhang, and S. Ma. Meta-evaluation of online and offline web search evaluation metrics. In *Proc. SIGIR*, pages 15–24, 2017.

[47] A. Chuklin, P. Serdyukov, and M. de Rijke. Click model-based information retrieval metrics. In *Proc. SIGIR*, pages 493–502, 2013.

[48] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proc. SIGIR*, pages 659–666, 2008.

[49] C. L. A. Clarke, N. Craswell, I. Soboroff, and A. Ashkan. A comparative analysis of cascade measures for novelty and diversity. In *Proc. WSDM*, pages 75–84, 2011.

[50] C. Cleverdon. The Cranfield tests on index language devices. *Aslib Proceedings*, 19 (6):173–194, 1967.

[51] C. Cleverdon and M. Keen. *Factors affecting the performance of indexing systems*, volume 2. ASLIB, Cranfield Research Project. Bedford, UK: C. Cleverdon, 1966.

[52] P. Clough and M. Sanderson. Evaluating the performance of information retrieval systems using test collections. *Information Research*, 18(2), June 2013.

[53] W. S. Cooper. Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American Documentation*, 19:30–41, 1968.

[54] W. S. Cooper. The inadequacy of probability of usefulness as a ranking criterion for retrieval system output. *School of Library and Information Studies, University of California, Berkeley*, 1971.

[55] W. S. Cooper. On selecting a measure of retrieval effectiveness. *J. Amer. Soc. Inf. Sci.*, 24(2):87–100, 1973.

[56] W. S. Cooper. The paradoxical role of unexamined documents in the evaluation of retrieval effectiveness. *Inf. Proc. & Man.*, 12(6):367–375, 1976.

[57] N. Craswell and D. Hawking. Overview of the TREC 2004 web track. In *Proc. TREC*, 2004.

[58] N. Craswell and S. E. Robertson. *Average Precision at n*, pages 193–194. Encyclopedia of Database Systems. Springer US, 2009.

[59] E. Cutrell and Z. Guan. What are you looking for? An eye-tracking study of information usage in web search. In *Proc. CHI*, pages 407–416, 2007.

[60] E. Cutrell, D. Robbins, S. T. Dumais, and R. Sarin. Fast, flexible filtering with Phlat. In *Proc. CHI*, pages 261–270, 2006.

[61] A. P. De Vries, G. Kazai, and M. Lalmas. Tolerance to irrelevance: A user-effort oriented evaluation of retrieval systems without predefined retrieval unit. In *Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval*, pages 463–473, 2004.

[62] B. Diedenhofen and J. Musch. cocor: A comprehensive solution for the statistical comparison of correlations. *PLOS ONE*, 10(4):1–12, 04 2015.

[63] H. A. Feild, J. Allan, and R. Jones. Predicting searcher frustration. In *Proc. SIGIR*, pages 34–41, 2010.

[64] M. Ferrante, N. Ferro, and M. Maistro. Towards a formal framework for utility-oriented measurements of retrieval effectiveness. In *Proc. ICTIR*, pages 21–30, 2015.

[65] S. Fox, K. Karnawat, M. Mydland, S. T. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Sys.*, 23(2):147–168, 2005.

[66] H. P. Frei and P. Schäuble. Determining the effectiveness of retrieval algorithms. *Inf. Proc. & Man.*, 27(2-3):153–164, 1991.

[67] N. Fuhr. A probability ranking principle for interactive information retrieval. *Inf. Retr.*, 11(3):251–265, 2008.

[68] W. Goffman. A searching procedure for information retrieval. *Inf. Stor. & Retr.*, 2 (2):73–78, 1964.

[69] C. D. Gull. Seven years of work on the organization of materials in the special library. *American Documentation*, 7(4):320–329, 1956.

[70] Q. Guo and E. Agichtein. Towards predicting web searcher gaze position from mouse movements. In *Proc. CHI*, pages 3601–3606, 2010.

[71] Q. Guo, R. W. White, Y. Zhang, B. Anderson, and S. T. Dumais. Why searchers switch: Understanding and predicting engine switching rationales. In *Proc. SIGIR*, pages 335–344, 2011.

[72] Q. Guo, D. Lagun, and E. Agichtein. Predicting web search success with fine-grained interaction data. In *Proc. CIKM*, pages 2050–2054, 2012.

[73] D. Harman. Overview of the second text retrieval conference (TREC-2). *Inf. Proc. & Man.*, 31(3):271–289, 1995.

[74] D. Harman. Overview of the TREC 2002 novelty track. In *Proc. of the Eleventh TREC*, pages 46–55, 2002.

[75] S. P. Harter and C. A. Hert. Evaluation of information retrieval systems: Approaches, issues, and methods. *Annual Review of Information Science and Technology (ARIST)*, 32:3–94, 1997.

[76] A. Hassan. A semi-supervised approach to modeling web search satisfaction. In *Proc. SIGIR*, pages 275–284, 2012.

[77] A. Hassan, R. Jones, and K. L. Klinkner. Beyond DCG: User behavior as a predictor of a successful search. In *Proc. WSDM*, pages 221–230, 2010.

[78] A. Hassan, X. Shi, N. Craswell, and B. Ramsey. Beyond clicks: Query reformulation as a predictor of search satisfaction. In *Proc. CIKM*, pages 2019–2028, 2013.

[79] D. Hawking and N. Craswell. Overview of the TREC-2001 web track. In *Proc. TREC*, Jan. 2001.

[80] D. Hawking, N. Craswell, and P. Bailey. Measuring search engine quality. *Inf. Retr.*, 4:33–59, 2001.

[81] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proc. SIGIR*, pages 76–84, 1996.

[82] W. Hersh, C. Buckley, T. J. Leone, and D. Hickam. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *Proc. SIGIR*, pages 192–201, 1994.

[83] W. Hersh, A. Turpin, S. Price, B. Chan, D. Kramer, L. Sacherek, and D. Olson. Do batch and user evaluations give the same results? In *Proc. SIGIR*, pages 17–24, 2000.

[84] C. R. Hildreth. Accounting for users' inflated assessments of on-line catalogue search performance and usefulness: An experimental study. *Information Research*, 6(2), 2001. URL `http://www.informationr.net/ir/6-2/paper101.html`.

[85] K. Hofmann. *Online Experimentation for Information Retrieval*, pages 21–41. Springer International Publishing, 2015.

[86] K. Hofmann, S. Whiteson, and M. de Rijke. Estimating interleaved comparison outcomes from historical click data. In *Proc. CIKM*, pages 1779–1783, 2012.

[87] K. Hofmann, L. Li, and F. Radlinski. Online evaluation for information retrieval. *Foundation and Trends in IR*, 10(1):1–117, June 2016.

[88] C. Hölscher and G. Strube. Web search behavior of internet experts and newbies. *Computer Networks*, 33(1):337–346, 2000.

[89] K. Hornbaek. Current practice in measuring usability: Challenges to usability studies and research. *International Journal of Human-Computer Studies*, 64(2):79–102, 2006.

[90] H. Hotelling. The selection of variates for use in prediction with some comments on the general problem of nuisance parameters. *The Annals of Mathematical Statistics*, 11(3):271–283, 1940.

[91] S. B. Huffman and M. Hochster. How well does result relevance predict session satisfaction? In *Proc. SIGIR*, pages 567–574, 2007.

[92] E. M. Hufnagel. User satisfaction-are we really measuring system effectiveness. In *Proc. HICSS*, volume 4, pages 437–446, 1990.

[93] D. Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proc. SIGIR*, pages 329–338, 1993.

[94] P. Ingwersen. Cognitive perspectives of information retrieval interaction: Elements of a cognitive IR theory. *J. Documentation*, 52:3–50, Jan. 1996.

[95] B. J. Jansen and A. Spink. How are we searching the world wide web? A comparison of nine search engine transaction logs. *Inf. Proc. & Man.*, 42(1):248–263, 2006.

[96] B. J. Jansen, A. Spink, J. Bateman, and T. Saracevic. Real life information retrieval: A study of user queries on the web. *SIGIR Forum*, 32(1):5–17, 1998.

[97] B. J. Jansen, K. J. Jansen, and A. Spink. Using the web to look for work: Implications for online job seeking and recruiting. *Internet Research*, 15:49–66, 2005.

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

[99] N. Jardine and C. J. Van Rijsbergen. The use of hierarchic clustering in information retrieval. *Inf. Stor. & Retr.*, 7(5):217–240, 1971.

[100] K. Järvelin. Explaining user performance in information retrieval: Challenges to IR evaluation. In *Proc. ICTIR*, pages 289–296, 2009.

[101] K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *Proc. SIGIR*, pages 41–48, 2000.

[102] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Sys.*, 20(4):422–446, 2002.

[103] K. Järvelin, S. L. Price, L. M. Delcambre, and M. L. Nielsen. Discounted cumulated gain based evaluation of multiple-query IR sessions. In *Proc. ECIR*, pages 4–15, 2008.

[104] J. Jiang and J. Allan. Correlation between system and user metrics in a session. In *Proc. CHIIR*, pages 285–288, 2016.

[105] J. Jiang and J. Allan. Adaptive effort for search evaluation metrics. In *Proc. ECIR*, pages 187–199, 2016.

[106] J. Jiang and J. Allan. Adaptive persistence for search effectiveness measures. In *Proc. CIKM*, pages 747–756, 2017.

[107] J. Jiang, D. He, and J. Allan. Searching, browsing, and clicking in a search session: Changes in user behavior by task and over time. In *Proc. SIGIR*, pages 607–616, 2014.

[108] J. Jiang, A. H. Awadallah, X. Shi, and R. W. White. Understanding and predicting graded search satisfaction. In *Proc. WSDM*, pages 57–66, 2015.

[109] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. KDD*, pages 133–142, 2002.

[110] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proc. SIGIR*, pages 154–161, 2005.

[111] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Sys.*, 25(2), Apr. 2007.

[112] M. Jones, G. Marsden, N. Mohd-Nasir, K. Boone, and G. Buchanan. Improving web interaction on small displays. *Computer Networks*, 31(11):1129–1137, 1999.

[113] D. Kahneman, B. L. Fredrickson, C. A. Schreiber, and D. A. Redelmeier. When more pain is preferred to less: Adding a better end. *Psychological Science*, 4(6): 401–405, 1993. doi: 10.1111/j.1467-9280.1993.tb00589.x.

[114] E. Kanoulas, B. Carterette, P. D. Clough, and M. Sanderson. Overview of the TREC 2010 session track. In *Proc. TREC*, 2010.

[115] E. Kanoulas, B. Carterette, P. D. Clough, and M. Sanderson. Evaluating multi-query sessions. In *Proc. SIGIR*, pages 1053–1062, 2011.

[116] P. B. Kantor and E. M. Voorhees. The TREC-5 confusion track: Comparing retrieval methods for scanned text. *Inf. Retr.*, 2(2-3):165–176, 2000.

[117] D. Kelly. Methods for evaluating interactive information retrieval systems with users. *Foundation and Trends in IR*, 3(1&2):1–224, 2009.

[118] D. Kelly and C. R. Sugimoto. A systematic review of interactive information retrieval evaluation studies, 1967–2006. *J. Amer. Soc. Inf. Sc. Tech.*, 64(4):745–770, 2013.

[119] D. Kelly, J. Arguello, A. Edwards, and W. Wu. Development and evaluation of search tasks for IIR experiments using a cognitive complexity framework. In *Proc. ICTIR*, pages 101–110, 2015.

[120] A. Kent, M. M. Berry, F. U. Luehrs Jr., and J. W. Perry. Machine literature searching VIII. Operational criteria for designing information retrieval systems. *American Documentation*, 6(2):93–101, 1955.

[121] H. Keskustalo, K. Järvelin, A. Pirkola, T. Sharma, and M. Lykke. Test collection-based IR evaluation needs extension toward sessions: A case of extremely short queries. In *Proc. Asia Info. Retri. Soc. Conf.*, pages 63–74, 2009.

[122] J. Kim, P. Thomas, R. Sankaranarayana, and T. Gedeon. Comparing scanning behaviour in web search on small and large screens. In *Proc. Aust. Doc. Comp. Symp.*, pages 25–30, 2012.

[123] K. Klöckner, N. Wirschum, and A. Jameson. Depth- and breadth-first processing of search result lists. In *Proc. CHI*, pages 1539–1539, 2004.

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

[125] D. Kraft. A software package for sequential quadratic programming. *Tech. Rep. DFVLR-FB 88-28, DLR German Aerospace Cent. Inst. Flight Mech., Koln*, 1988.

[126] D. H. Kraft and D. A. Buell. Advances in a Bayesian decision model of user stopping behavior for scanning the output of an information retrieval system. In *Proc. SIGIR*, pages 421–433, 1984.

[127] U. Krishnan, A. Moffat, and J. Zobel. A taxonomy of query auto completion modes. In *Proc. Aust. Doc. Comp. Symp.*, pages 1–8, 2017.

[128] M. Kudlyak and J. Faberman. The intensity of job search and search duration, 2014. Working Paper 14-12, Federal Reserve Bank of Richmond, `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2442910`.

[129] D. Lagun, C. Hsieh, D. Webster, and V. Navalpakkam. Towards better measurement of attention and satisfaction in mobile search. In *Proc. SIGIR*, pages 113–122, 2014.

[130] R. Lempel and S. Moran. Predictive caching and prefetching of query results in search engines. In *Proc. WWW*, pages 19–28, 2003.

[131] M. E. Lesk and G. Salton. Relevance assessments and retrieval system evaluation. *Inf. Stor. & Retr.*, 4(4):343–359, 1968.

[132] A. Leuski and J. Allan. Improving interactive retrieval by combining ranked lists and clustering. In *Content-Based Multimedia Information Access - Volume 1*, pages 665–681, 2000.

[133] J. Li, S. Huffman, and A. Tokuda. Good abandonment in mobile and PC internet search. In *Proc. SIGIR*, pages 43–50, 2009.

[134] J. Li, D. Arya, V. Ha-Thuc, and S. Sinha. How to get them a dream job? Entity-aware features for personalized job search ranking. In *Proc. KDD*, pages 501–510, 2016.

[135] A. Lipani, B. Carterette, and E. Yilmaz. From a user model for query sessions to session rank biased precision (sRBP). In *Proc. ICTIR*, pages 109–116, 2019.

[136] J. Liu, C. Liu, M. Cole, N. J. Belkin, and X. Zhang. Exploring and predicting search task difficulty. In *Proc. CIKM*, pages 1313–1322, 2012.

[137] M. Liu, Y. Liu, J. Mao, C. Luo, and S. Ma. Towards designing better session search evaluation metrics. In *Proc. SIGIR*, pages 1121–1124, 2018.

[138] M. Liu, Y. Liu, J. Mao, C. Luo, M. Zhang, and S. Ma. Satisfaction with failure or unsatisfied success: Investigating the relationship between search success and user satisfaction. In *Proc. WWW*, pages 1533–1542, 2018.

[139] M. Liu, J. Mao, Y. Liu, M. Zhang, and S. Ma. Investigating cognitive effects in session-level search user satisfaction. In *Proc. KDD*, pages 923–931, 2019.

[140] Y. Liu, Y. Chen, J. Tang, J. Sun, M. Zhang, S. Ma, and X. Zhu. Different users, different opinions: Predicting search satisfaction with mouse movement information. In *Proc. SIGIR*, pages 493–502, 2015.

[141] X. Lu, A. Moffat, and J. S. Culpepper. The effect of pooling and evaluation depth on IR metrics. *Inf. Retr.*, 19(4):416–445, 2016.

[142] J. Luo, C. Wing, H. Yang, and M. Hearst. The water filling model and the cube test: Multi-dimensional evaluation for professional search. In *Proc. CIKM*, pages 709–714, 2013.

[143] J. Mackenzie, R. Benham, M. Petri, J. R. Trippas, J. S. Culpepper, and A. Moffat. CC-News-En: A large English news corpus. In *Proc. CIKM*, pages 3077–3084, 2020.

[144] B. Mansouri, M. S. Zahedi, R. Campos, and M. Farhoodi. Online job search: Study of users' search behavior using search engine query logs. In *Proc. SIGIR*, pages 1185–1188, 2018.

[145] J. Mao, Y. Liu, K. Zhou, J. Nie, J. Song, M. Zhang, S. Ma, J. Sun, and H. Luo. When does relevance mean usefulness and user satisfaction in web search? In *Proc. SIGIR*, pages 463–472, 2016.

[146] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *J. ACM*, 7(3):216–244, July 1960.

[147] D. Maxwell, L. Azzopardi, K. Järvelin, and H. Keskustalo. An initial investigation into fixed and adaptive stopping strategies. In *Proc. SIGIR*, pages 903–906, 2015.

[148] D. Maxwell, L. Azzopardi, K. Järvelin, and H. Keskustalo. Searching and stopping: An analysis of stopping rules and strategies. In *Proc. CIKM*, pages 313–322, 2015.

[149] A. Moffat. Seven numeric properties of effectiveness metrics. In *Proc. Asia Info. Retri. Soc. Conf.*, pages 1–12. Springer Berlin Heidelberg, 2013.

[150] A. Moffat and A. F. Wicaksono. Users, adaptivity, and bad abandonment. In *Proc. SIGIR*, pages 897–900, 2018.

[151] A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Trans. Inf. Sys.*, 27(1):2.1–2.27, 2008.

[152] A. Moffat, F. Scholer, and P. Thomas. Models and metrics: IR evaluation as a user process. In *Proc. Aust. Doc. Comp. Symp.*, pages 47–54, 2012.

[153] A. Moffat, P. Thomas, and F. Scholer. Users versus models: What observation tells us about effectiveness metrics. In *Proc. CIKM*, pages 659–668, 2013.

[154] A. Moffat, P. Bailey, F. Scholer, and P. Thomas. INST: An adaptive metric for information retrieval evaluation. In *Proc. Aust. Doc. Comp. Symp.*, pages 5:1–5:4, 2015.

[155] A. Moffat, P. Bailey, F. Scholer, and P. Thomas. Incorporating user expectations and behavior into the measurement of search effectiveness. *ACM Trans. Inf. Sys.*, 35(3):24:1–24:38, 2017.

[156] C. N. Mooers. Zatocoding applied to mechanical organization of knowledge. *American Documentation*, 2(1):20–32, 1951.

[157] R. Navarro-Prieto, M. Scaife, and Y. Rogers. Cognitive strategies in web searching. In *Proc. Conf. on Human Factors and the Web*, pages 1–12, 1999.

[158] J. F. Nunamaker, L. M. Applegate, and B. R. Konsynski. Facilitating group creativity: Experience with a group decision support system. *J. Manage. Inf. Syst.*, 3 (4):5–19, Apr. 1987.

[159] D. Odijk, R. W. White, A. H. Awadallah, and S. T. Dumais. Struggling and success in web search. In *Proc. CIKM*, pages 1551–1560, 2015.

[160] K. Ong, K. Järvelin, M. Sanderson, and F. Scholer. Using information scent to understand mobile and desktop web search behavior. In *Proc. SIGIR*, pages 295–304, 2017.

[161] P. Over. TREC-7 interactive track report. In *Proc. of the Seventh TREC*, pages 57–64, 1999.

[162] U. Ozertem, R. Jones, and B. Dumoulin. Evaluating new search engine configurations with pre-existing judgments and clicks. In *Proc. WWW*, pages 397–406, 2011.

[163] L. A. F. Park and Y. Zhang. On the distribution of user persistence for rank-biased precision. In *Proc. Aust. Doc. Comp. Symp.*, pages 1:1–1:8, 2007.

[164] P. Pirolli and S. Card. Information foraging. *Psychological Review*, 4(106):643–675, 1999.

[165] F. Radlinski and N. Craswell. Comparing the sensitivity of information retrieval metrics. In *Proc. SIGIR*, pages 667–674, 2010.

[166] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *Proc. ICML*, pages 784–791, 2008.

[167] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *Proc. CIKM*, pages 43–52, 2008.

[168] S. E. Robertson. The probability ranking principle in IR. *J. Documentation*, 33(4): 294–304, 1977.

[169] S. E. Robertson. A new interpretation of average precision. In *Proc. SIGIR*, pages 689–690, 2008.

[170] S. E. Robertson. A brief history of search results ranking. *IEEE Annals of the History of Computing*, 41(02):22–28, apr 2019.

[171] S. E. Robertson, E. Kanoulas, and E. Yilmaz. Extending average precision to graded relevance judgments. In *Proc. SIGIR*, pages 603–610, 2010.

[172] K. Roitero, E. Maddalena, G. Demartini, and S. Mizzaro. On fine-grained relevance scales. In *Proc. SIGIR*, pages 675–684, 2018.

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

[174] A. Saha and D. Arya. Generalized mixed effect models for personalizing job search. In *Proc. SIGIR*, pages 1129–1132, 2017.

[175] T. Sakai. Ranking the NTCIR systems based on multigrade relevance. In *Proc. Asia Info. Retri. Soc. Conf.*, pages 251–262, 2004.

[176] T. Sakai. New performance metrics based on multigrade relevance: Their application to question answering. In *Proc. NTCIR*, 2004.

[177] T. Sakai. Evaluating evaluation metrics based on the bootstrap. In *Proc. SIGIR*, pages 525–532, 2006.

[178] T. Sakai. Alternatives to Bpref. In *Proc. SIGIR*, pages 71–78, 2007.

[179] T. Sakai. On penalising late arrival of relevant documents in information retrieval evaluation with graded relevance. In *Proc. Workshop on Eval. Inf. Acc.*, pages 32–43, 2007.

[180] T. Sakai. Modelling a user population for designing information retrieval metrics. In *Proc. Workshop on Eval. Inf. Acc.*, pages 30–41, 2008.

[181] T. Sakai and Z. Dou. Summaries, ranked retrieval and sessions: A unified framework for information access evaluation. In *Proc. SIGIR*, pages 473–482, 2013.

[182] T. Sakai and Z. Zeng. Which diversity evaluation measures are “good”? In *Proc. SIGIR*, pages 595–604, 2019.

[183] B. Salehi, D. Spina, A. Moffat, S. Sadeghi, F. Scholer, T. Baldwin, L. Cavedon, M. Sanderson, W. Wong, and J. Zobel. A living lab study of query amendment in job search. In *Proc. SIGIR*, pages 905–908, 2018.

[184] M. Sanderson. Test collection based evaluation of information retrieval systems. *Foundation and Trends in IR*, 4(4):247–375, 2010.

[185] M. Sanderson and J. Zobel. Information retrieval system evaluation: Effort, sensitivity, and reliability. In *Proc. SIGIR*, pages 162–169, 2005.

[186] M. Sanderson, M. L. Paramita, P. Clough, and E. Kanoulas. Do user preferences and evaluation measures line up? In *Proc. SIGIR*, pages 555–562, 2010.

[187] T. Saracevic. Relevance: A review of and a framework for the thinking on the notion in information science. *J. Amer. Soc. Inf. Sci.*, 26(6):321–343, 1975.

[188] T. Saracevic. Evaluation of evaluation in information retrieval. In *Proc. SIGIR*, pages 138–146, 1995.

[189] T. Saracevic and P. B. Kantor. A study of information seeking and retrieving. ii. users, questions, and effectiveness. *J. Amer. Soc. Inf. Sci.*, 39:177–196, 1988.

[190] R. Savolainen. Information need as trigger and driver of information seeking: a conceptual analysis. *Aslib J. Inf. Manag.*, 69:2–21, 2017.

[191] L. Schamber, M. Eisenberg, and M. S. Nilan. A re-examination of relevance: Toward a dynamic, situational definition. *Inf. Process. Manage.*, 26(6):755–776, 1990.

[192] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

[193] M. Sloan and J. Wang. Dynamic information retrieval: Theoretical framework and application. In *Proc. ICTIR*, pages 61–70, 2015.

[194] C. L. Smith and P. B. Kantor. User adaptation: Good results from poor systems. In *Proc. SIGIR*, pages 147–154, 2008.

[195] M. D. Smucker and C. L. A. Clarke. Time-based calibration of effectiveness measures. In *Proc. SIGIR*, pages 95–104, 2012.

[196] M. D. Smucker and C. P. Jethani. Human performance and retrieval precision revisited. In *Proc. SIGIR*, pages 595–602, 2010.

[197] D. Soergel. Is user satisfaction a hobgoblin? *J. Amer. Soc. Inf. Sci.*, 27(4):256–259, 1976.

[198] E. Sormunen. Liberal relevance criteria of TREC: Counting on negligible documents? In *Proc. SIGIR*, pages 324–330, 2002.

[199] K. Spärck Jones. *Information Retrieval Experiment.* Butterworths, 1981.

[200] D. Spina, M. Maistro, Y. Ren, S. Sadeghi, W. Wong, T. Baldwin, L. Cavedon, A. Moffat, M. Sanderson, F. Scholer, and J. Zobel. Understanding user behavior in job and talent search: An initial investigation. In *SIGIR Wrkshp. eCommerce*, 2017.

[201] A. Spink. A user-centered approach to evaluating human interaction with web search engines: An exploratory study. *Inf. Proc. & Man.*, 38(3):401–426, 2002.

[202] A. Spink, D. Wolfram, B. J. Jansen, and T. Saracevic. Searching the web: The public and their queries. *J. Amer. Soc. Inf. Sc. Tech.*, 52(3):226–234, 2001.

[203] A. Spink, S. Ozmutlu, H. C. Ozmutlu, and B. J. Jansen. U.S. versus European web searching trends. *SIGIR Forum*, 36(2):32–38, 2002.

[204] L. T. Su. Evaluation measures for interactive information retrieval. *Inf. Proc. & Man.*, 28(4):503–516, 1992.

[205] L. T. Su. The relevance of recall and precision in user evaluation. *J. Amer. Soc. Inf. Sci.*, 45(3):207–217, 1994.

[206] L. T. Su. A comprehensive and systematic model of user evaluation of web search engines: I. theory and background. *J. Amer. Soc. Inf. Sc. Tech.*, 54(13):1175–1192, 2003.

[207] J. A. Swets. Information retrieval systems. *Science*, 141:245–250, 1963.

[208] J. M. Tague-Sutcliffe. Some perspectives on the evaluation of information retrieval systems. *J. Amer. Soc. Inf. Sci.*, 47(1):1–3, Jan. 1996.

[209] J. Teevan, C. Alvarado, M. S. Ackerman, and D. R. Karger. The perfect search engine is not enough: A study of orienteering behavior in directed search. In *Proc. CHI*, pages 415–422, 2004.

[210] P. Thomas and D. Hawking. Evaluation by comparing result sets in context. In *Proc. CIKM*, pages 94–101, 2006.

[211] P. Thomas, F. Scholer, and A. Moffat. What users do: The eyes have it. In *Proc. Asia Info. Retri. Soc. Conf.*, pages 416–427, 2013.

[212] P. Thomas, A. Moffat, P. Bailey, and F. Scholer. Modeling decision points in user search behavior. In *Proc. IIiX*, pages 239–242, 2014.

[213] P. Thomas, A. Moffat, P. Bailey, F. Scholer, and N. Craswell. Better effectiveness metrics for SERPs, cards, and rankings. In *Proc. Aust. Doc. Comp. Symp.*, pages 1:1–1:8, 2018.

[214] E. G. Toms and L. Freund. Predicting stopping behaviour: A preliminary analysis. In *Proc. SIGIR*, pages 750–751, 2009.

[215] V. T. Tran and N. Fuhr. Using eye-tracking with dynamic areas of interest for analyzing interactive information retrieval. In *Proc. SIGIR*, pages 1165–1166, 2012.

[216] V. T. Tran and N. Fuhr. Markov modeling for user interaction in retrieval. In *SIGIR 2013 Wrkshp. Model. User Beh. Inf. Retr. Eval.*, 2013.

[217] V. T. Tran, D. Maxwell, N. Fuhr, and L. Azzopardi. Personalised search time prediction using Markov chains. In *Proc. ICTIR*, pages 237–240, 2017.

[218] A. Turpin and W. Hersh. Why batch and user evaluations do not give the same results. In *Proc. SIGIR*, pages 225–231, 2001.

[219] A. Turpin and F. Scholer. User performance versus precision measures for simple search tasks. In *Proc. SIGIR*, pages 11–18, 2006.

[220] A. Turpin, F. Scholer, S. Mizzaro, and E. Maddalena. The benefits of magnitude estimation relevance assessments for information retrieval evaluation. In *Proc. SIGIR*, pages 565–574, 2015.

[221] C. J. Van Rijsbergen. Foundation of evaluation. *J. Documentation*, 30(4):365–373, 1974.

[222] C. J. Van Rijsbergen. *Information Retrieval.* Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. ISBN 0408709294.

[223] M. Viswanathan. *Measurement Error and Research Design.* Thousand Oaks, CA: SAGE Publications, 2005.

[224] E. M. Voorhees. Evaluation by highly relevant documents. In *Proc. SIGIR*, pages 74–82, 2001.

[225] E. M. Voorhees. The philosophy of information retrieval evaluation. In *Proc. CLEF*, pages 355–370, 2002.

[226] E. M. Voorhees. Overview of the TREC 2004 robust retrieval track. *Thirteenth Text Retrieval Conference (TREC 2004)*, 2005.

[227] E. M. Voorhees and D. K. Harman. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing).* The MIT Press, 2005.

[228] Q. H. Vuong. Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica*, 57(2):307–333, 1989.

[229] H. Wang, Y. Song, M. Chang, X. He, A. Hassan, and R. W. White. Modeling action-level satisfaction for search task satisfaction prediction. In *Proc. SIGIR*, pages 123–132, 2014.

[230] J. Wang and J. Zhu. Portfolio theory of information retrieval. In *Proc. SIGIR*, pages 115–122, 2009.

[231] Y. Wang, D. Yin, L. Jie, P. Wang, M. Yamada, Y. Chang, and Q. Mei. Beyond ranking: Optimizing whole-page presentation. In *Proc. WSDM*, pages 103–112, 2016.

[232] W. Webber, A. Moffat, J. Zobel, and T. Sakai. Precision-at-ten considered redundant. In *Proc. SIGIR*, pages 695–696, 2008.

[233] R. W. White and S. M. Drucker. Investigating behavioral variability in web search. In *Proc. WWW*, pages 21–30, 2007.

[234] W. J. Wilbur. An information measure of retrieval performance. *Information Systems*, 17(4):283–298, 1992.

[235] B. Wildemuth, L. Freund, and E. Toms. Untangling search task complexity and difficulty in the context of interactive information retrieval studies. *J. Documentation*, 70:1118–1140, 2014.

[236] W. C. Wu, D. Kelly, and A. Sud. Using information scent and need for cognition to understand online search behavior. In *Proc. SIGIR*, pages 557–566, 2014.

[237] Y. Yang and A. Lad. Modeling expected utility of multi-session information distillation. In *Proc. ICTIR*, pages 164–175, 2009.

[238] Z. Yang, A. Moffat, and A. Turpin. Pairwise crowd judgments: Preference, absolute, and ratio. In *Proc. Aust. Doc. Comp. Symp.*, pages 1–8, 2018.

[239] E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Proc. CIKM*, pages 102–111, 2006.

[240] E. Yilmaz, M. Shokouhi, N. Craswell, and S. E. Robertson. Expected browsing utility for web search evaluation. In *Proc. CIKM*, pages 1561–1564, 2010.

[241] F. Zhang, Y. Liu, X. Li, M. Zhang, Y. Xu, and S. Ma. Evaluating web search with a bejeweled player model. In *Proc. SIGIR*, pages 425–434, 2017.

[242] F. Zhang, J. Mao, Y. Liu, W. Ma, M. Zhang, and S. Ma. Cascade or recency: Constructing better evaluation metrics for session search. In *Proc. SIGIR*, pages 389–398, 2020.

[243] Y. Zhang and A. Moffat. Some observations on user search behavior. In *Proc. Aust. Doc. Comp. Symp.*, pages 1–8, 2006.

[244] Y. Zhang, L. A. F. Park, and A. Moffat. Click-based evidence for decaying weight distributions in search effectiveness metrics. *Inf. Retr.*, 13(1):46–69, 2010.

[245] Y. Zheng, J. Mao, Y. Liu, M. Sanderson, M. Zhang, and S. Ma. Investigating examination behavior in mobile search. In *Proc. WSDM*, pages 771–779. Association for Computing Machinery, 2020.

[246] X. Zhu, J. Guo, X. Cheng, Y. Lan, and W. Nejdl. Recommending high utility query via session-flow graph. In *Proc. ECIR*, pages 642–655, 2013.

[247] J. Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proc. SIGIR*, pages 307–314, 1998.

[248] J. Zobel, A. Moffat, and L. A. F. Park. Against recall: Is it persistence, cardinality, density, coverage, or totality? *SIGIR Forum*, 43(1):3–8, June 2009.

University Library

MINERVA ACCESS

A gateway to Melbourne's research publications

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Wicaksono, Alfan Farizki

Title:
Modelling search and session effectiveness

Date:
2020

Persistent Link:
http://hdl.handle.net/11343/258806

File Description:
Final thesis file - Compressed file

Terms and Conditions:
Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.